THE HAGUE UNIVERSITY OF APPLIED SCIENCES

MECHATRONICS

GRADUATION

# Improving autonomous docking of assistive robots

## DESIGN OF AN EFFICIENT WIRELESS CHARGING MECHANISM

*Author:*
Bart Geert Entink

*1st Coordinator:*
Ing. T.J. Koreneef
*2nd Coordinator:*
Drs. D.G. van Teylingen

*1st Supervisor:*
Ing. Guus Liqui Lung
*2nd Supervisor:*
Ir. Aswin Chandarr

June 3, 2015


# Preface

This report of my graduation assignment was written in order to achieve a bachelor of Mechatronics from the Hague University of applied sciences. With the report, I would like to show the skills I learned in my courses and show the process and result of the assignment.

This document is meant as an informative description of an internship at the Delft Biorobotics lab of the TU Delft department of biomechanical engineering at the faculty 3ME, and consists of the realisation of a charging device for assistive robots. The reader will be shown all phases of the project, from research to testing.

It was written for people working at the biomechanical department and the coordinators chosen for my graduation. Having said this, all other people, interested in this subject, are welcome to read it.

Last, I would like to thank the people who helped me with getting the assignment done; be it through technical expertise, critic reviewing or even moral support. Special thanks to Guus Liqui Lung and Aswin Chandarr for giving me the opportunity to do this assignment.

Delft, June 2015

# Contents

## Summary

Research and development of robots which can support elderly or disabled people is pursued to increase the quality of home care. With the ever increasing quality and functions of robots, full-time care by these robots becomes feasible. To make the robot function without any human interferences, the robot should function autonomously and be able to recharge by itself.

Increasing the quality of the charging process of autonomous docking improves the overall quality of autonomous robots. The currently used docking station is susceptible to damage caused by small errors in the docking process, resulting in electrical sparks. The idea of the robot is that it can operate autonomously, while damage to the charging mechanism makes human interference necessary.

By researching complete systems, charging principles and docking methods, a new docking station was designed. Through use of a different charging principle that uses magnetic fields instead of electricity to transfer energy, electrical sparks cannot occur, thus making it less susceptible to damage.
To decrease the occurring errors when docking, a simple yet effective docking method is proposed. The docking method uses available vision systems on a robot to identify a part of the docking station and move towards it.

Tests concluded that the efficiency of the energy transfer can go as high as $95\%$. The tested docking method shows that it is possible to reach the required efficiency even with small alignment errors.

The resulting docking station provides a better way to charge robots without human interference. Optimisation of the system is still required, but the first test results are promising and show the feasibility of the system.

# List of Figures



# List of Tables

# 1 Preamble

To create robots, which can assist in healthcare, that can operate for weeks, months or even years without any human interference, an autonomous way to recharge is needed. This recharging is currently done through the use of docking stations, which the robot can locate and move towards. The docking process however, needs accuracy to prevent electrical sparks between contacts from occurring. This required accuracy is often not met and the electrical sparks which occur can cause permanent damage to the mechanical contacts. A possible result can be that the robot needs human interference thus stopping its current and future tasks till repaired.

This report tries to solve the problems that occur while docking autonomously, with the docking station being the main subject. It will give an in depth view of the several project phases that led to the final docking station.

In chapter 2 the given problem is thoroughly explained while different views are given on the subject. The chapter also describes the requirements which the system should meet. Chapter 3 describes the research which was done to find an optimal solution. The chapter describes different possible options for two subsystems. Chapter 4 gives a description of which options are the best choice based on the requirements, after which these options are made into a design for the application. In chapter 5 an explanation is given on how the subsystems were realised and tested. Finally the conclusions and recommendations are given for further improvements of the system.

# 2 Introduction

## 2.1 Background

The personal robot group of Delft Biorobotics Lab (DBL) focusses on developing mobile robots for assisting elderly, disabled and children with special needs. Currently three generations of these robots have passed, named Robby, Lea and Chico, all developed at DBL. The focus lies on having a simple, yet smart mechanical design coupled with sophisticated control and perception systems to tackle the challenges of operating in dynamic human environments. Various components currently implemented include autonomous navigation, interactive object manipulation, person following, face recognition and voice recognition. There is an active development of behaviour engines to integrate the various modules and perform task planning based on user intentions. Along with an effective industrial design the aim is to develop an affordable and socially acceptable personal robot.

Even though the assignment is done at the DBL, robot care systems (RCS) and robot security systems (RSS) also profit from this research. RCS is a spin-off of the DBL to make several researched applications commercially available. RSS is a company in the same office area as RCS and concentrates on security and surveillance robots.

Long term autonomous robots are necessary to make these systems self operating in known environments. This requires an automatic docking and charging system for the robots power supply.

## 2.2 Problem description

The current docking stations for the personal robots, use mechanical contacts to connect to the docking station. Only RSS has an operating docking station for their robots. The alignment is done with the robots software and a small part of hardware on the docking station to support this. The hardware on the docking station is added to increase the allowable error. The reason for increasing the allowable error is because of electrical sparks occurring if the alignment is not good enough. These sparks can potentially harm the contacts permanently, requiring replacements.

To make the problem more comprehensible, the system is split into two subsystems. These subsystems will be discussed and options will be given, after which a choice will be made using a morphological analysis. This is done to ensure the best components for each subsystem and to cut the amount of concepts to be able to invest more time in the better ones. The two subsystems and their description are given below.

- Power transfer; the most important part of a charging device. Research is done on how the power transfer can be be optimised for the requirements given in paragraph 2.4
- Docking behaviour; consists of the software and hardware needed for docking the robot. It is assumed that with navigation the robot is in proximity of the dock.

For this internship, the main goal will be designing and developing a docking station which can charge personal robots without the current problems.

## 2.3 Orientation

To get an idea of the current standing of the needed system and its subsystems, robots with similar docking station were analysed.

### 2.3.1 RSS

RSS has a working docking station, of which a photo can be seen in figure 1. The hardware consists of spring loaded contacts and a mechanism which can rotate the contacts slightly. The spring loaded contacts are applied in sets of three to increase the efficiency of the power transfer. Four of these sets are used, two for the actual power transfer and two for testing whether the robot is connected.

Even though additions have been made to improve this docking station, it still does not suffice for autonomous robots. If these robots are going to be operating without human interference, a docking station has to be made which allows for a larger error margin.

**Figure 1:** Docking station used at RSS.

### 2.3.2 Existing systems

Different companies have brought robots to the market which can already autonomously dock. As most of these robots are designed with a different goal, but do use methods which could potentially be used as example, an overview of some robots is given. As the power transfer is identical for these robots, the focus lies with the docking behaviour.

- Roomba, developed by iRobot [7].

  Probably one of the best known robots. It vacuums your house automatically and starts charging when needed or when the vacuuming is done. The Roomba finds its docking station by detecting the non-visible light field the station emits [9], as shown in figure 2a. If the Roomba passes through this light field, it can find its way back to the docking station, if it does not find its way back in time, it automatically shuts down, waiting for human interference.

- Double, developed by Double Robotics [17].

  A remote telepresence robot which can be manually controlled from anywhere to be in the office without physically being there. The Double can be ordered with a docking station which can be seen in figure 2b. The docking station is a rather simple design, it uses hardware to mate the contacts. The guiding hardware is movable and moves to the left or right when the robot is not being docked perfectly.

- Turtlebot, developed by clearpath robotics [5].

  The Turtlebot, shown in figure 2c, is a robot often used for development because of the open source software to use with it. The Turtlebot is built on top of a Kobuki that was developed by Yujin robot [16]. The Turtlebot package consists of the Kobuki, a Kinect and a frame to which the Kinect and, optionally, other hardware can be attached. A docking station is optional and software exists which makes the Kobuki autodock.

  The docking of the robot is done through three infrared emitters on the dock, which cover three specific regions: left, right and centre. The Kobuki is equipped with three sensors, one on the front and one on each side. By checking which sensor receives a signal, the dock can be located. By identifying the emitter of the signal, action can be undertaken to go to the centre region and dock from there.

**(a)** Roomba vacuumcleaner going towards its docking station.

**(b)** Double personal assistant docked in its docking station.

**(c)** Turtlebot with the docking station in front of it.

**Figure 2:** Docking stations of the Roomba, the Double and the Turtlebot.

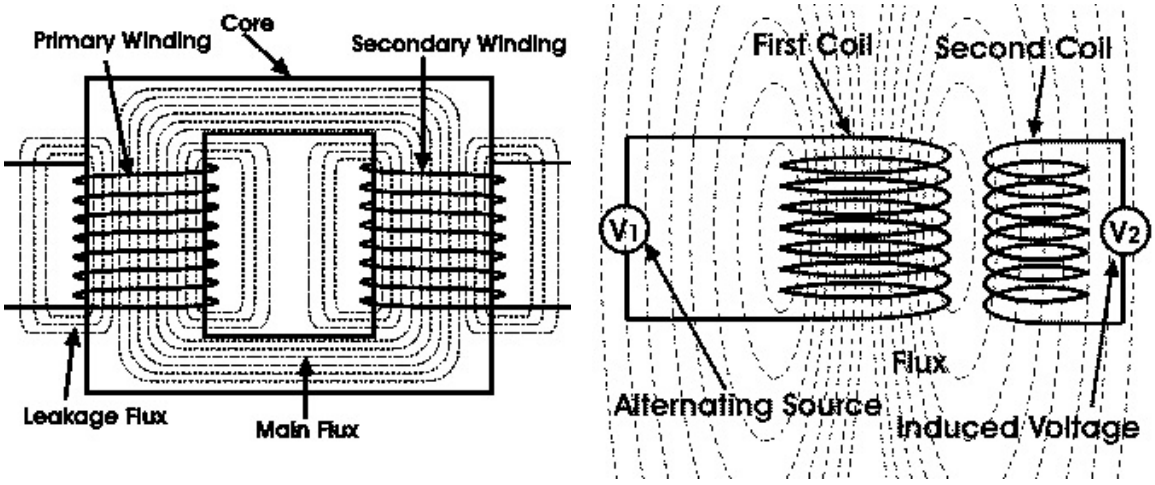
### 2.3.3 Power transfer

There are a lot of methods to transfer energy, most of these methods are however experimental or not applicable for this situation. To get a an understanding of the possible methods, a small explanation on the energy transfers is given.

All the robots described in the previous paragraph, dock and charge with conductive power transfer while there is physical contact. Conductive power transfer is basically connecting two electricity conducting parts so a current can pass through. This is the most common way to charge batteries and can be seen in all electrical equipment.

A different method is to transfer energy through induction, inductive power transfer (IPT) [2]. IPT makes use of an alternating current (AC) and coils to create an electromagnetic field. By using two coils, of which the transmitting coil is connected to the AC while the receiving coil is connected to a load, energy can be transferred through inducting a current into the receiver. This effect is mainly used in transformers to transform the voltage and current. The transformer uses a core to guide the flux from the magnetic field reducing leakage flux, thus losses, as can be seen in figure 3a.

Power can also be transferred wireless with IPT by removing the core, as can be seen in figure 3b. More losses occur however, as there is nothing to guide the flux towards the receiving coil thus increasing the leakage flux.




**(a)** Coils with a core, which guides the flux in the correct direction. The leakage flux becomes significantly less with use of a core

**(b)** Coils without a core which shows a lot of leakage flux, for wireless charging the coils are aligned vertically to give better efficiency.

**Figure 3:** Comparing coils with and without core

## 2.4 Requirements

For the system, requirements were made to make clear what is expected. The full description of the requirements can be found in appendix A. The requirements are split into four categories: Electrical (E), mechanical (M), software (S) and global (G) and each have an unique reference number.

**Electrical**

E01: *Mutation date 19-02-2015* The docking station has to be able to charge wireless.
E19: *Mutation date 16-03-2015* The docking station has to be charged electrically.
E20: The docking station has to be able to charge wireless.

E21: The docking station should use wall power at, possibly higher frequencies.
E02: Charging has to be done with a distance of atleast 5 mm.
E03: The charging mechanism charges the robot with atleast 30V and 10A.
E04: The charging time must not exceed 2 hours.
E05: The power transfer efficiency must be atleast 80%.
E06: Insulation is needed to prevent disturbances.
E07: The dock should be activated when the robot is in charging range of the dock.

**Mechanical**

M08: The system must be designed modular, to be adapted to different robots, with different energy requirements.
M15: To the robot, a maximum of 0.5 kg can be added.
M16: The docking station may weigh up to 3 kg.
M17: The maximum size of the docking station is 500mmx500mmx100m (w x h x d).
M18: The docking station should not contain any actuators.

**Software**

S09: The docking behaviour of the robot should use its navigation system and equipped sensors.
S13: The docking station should be able to indicate whether or not the robot is in proximity.
S14: The docking station should detect if the input and output voltages are correct.

**Global**

G10: The dock will be connected to the main power supply in the house.
G11: The docking station will be in a fixed position.
G12: The docking station has to be safe.

# 3 Research

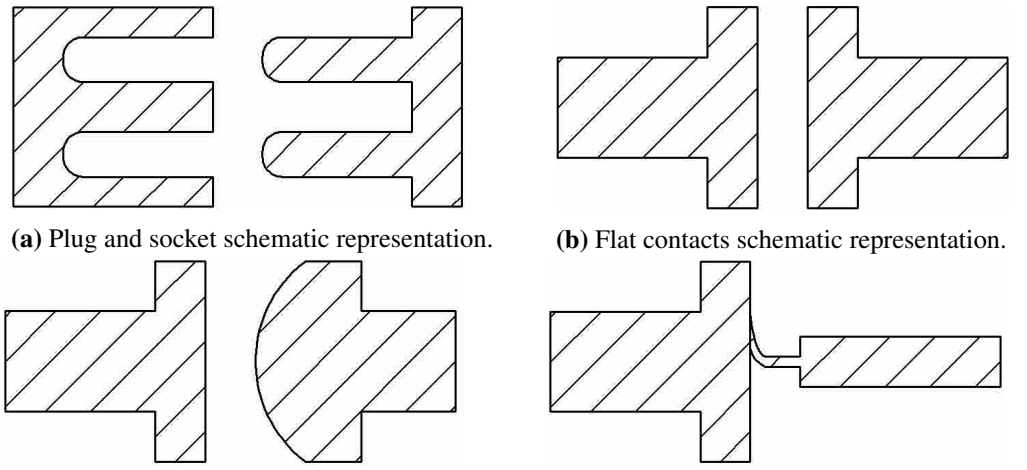
This chapter describes the research on the different aspects of the docking station. It will explain several possibilities for each subsystem.

## 3.1 Power transfer

Even though there are different ways to transfer energy, most of these are not applicable for a docking station. The two earlier mentioned methods to transfer energy will be researched, conductive and inductive.

### 3.1.1 Conductive

Conductive power transfer consists of creating contact to transfer electrical energy. The shape of the contact creates the biggest influence on the transfer which why why the focus of this paragraph lies on the contact shape. Four shapes are shown in figure 4 and each one will be discussed shortly.



**(a)** Plug and socket schematic representation.

**(b)** Flat contacts schematic representation.

**(c)** Flat and shaped schematic representation.

**(d)** Sliding contact schematic representation.

**Figure 4:** Overview of all wired charging connectors.

The shape shown in figure 4a represents a socket and a plug. This is the safest option as long as the current comes from the socket. The efficiency can be high as the plug can fully connect with the socket, giving a large contact area. The problem lies with the aligning, as it is rather hard to fit the plug in the correct way.

The second shape shown in figure 4b basically consists of two flat parts, which can be pushed against each other. The safety of having flat plates conduct electricity is low as these contacts can be easily touched, which is why extra safety measures have to be taken in account. An advantage of this shape is the larger contact area without the harder alignment as with the plug. However the contacts need to be parallel to each other to use the whole contact area and prevent electrical arcs.

In figure 4c a contact is shown with one flat contact and one round contact. This round contact makes the aligning easier, as there is no requirement to the contacts being parallel. The safety is low for the same reason that two flat contacts have a low safety. The contact area of this shape is low and thus the chance for electrical sparks occurring is higher.

The last shape shown in figure 4d represents a sliding contact, the figure shows a brushlike contact that scrapes a flat contact. The aligning can be made easier by increasing the size of the conducting plate. It has to be noticed that increasing the conducting plate size, increases the touchable area resulting in a bigger safety hazard. The contact area can be increased by making the brushlike contact thicker.

Depending on the analysis in the next section, these contacts shapes can be combined depending on the strengths and weaknesses of each shape. These four given options only represent basic concepts, while combinations of these options could work better for the given problem.

### 3.1.2 Inductive

The method which is used in several different applications is the inductive power transfer (IPT), it consists of two coils which are magnetically coupled, the transmitting coil and the receiving coil [2]. Below is an overview of different applications where IPT can be found.

- Phone chargers and low power devices.

  These chargers are commercially available in a lot of different shapes and power classes. Often these chargers come in the shape of pads having several transmitting coils and requiring the device to be put on top of it [10] [22]. The main focus of these researches lie at range and received power. The ultimate goal is to charge these devices independent of the location. As IPT requires a receiving coil to be present in the device, several standards are in development to make these coils universal [14] [11] [23].

- Pacemakers and hard to reach low power devices.

  In the medical world, IPT is used to recharge pacemakers which would otherwise require surgery. The design criteria of these systems are similar to the phone chargers but the range has to be higher while the efficiency can be lower [21].

- Automated guided vehicles.

  These kind of robots are charged when moving along a specified path [18]. Research was done on making these robots charge without contact, reducing wear while increasing safety [19]. The goal of these researches is to find a way to constantly provide power to a robot without physical contact. The aim of the research lies at increasing the efficiency of the system and creating longer tracks.

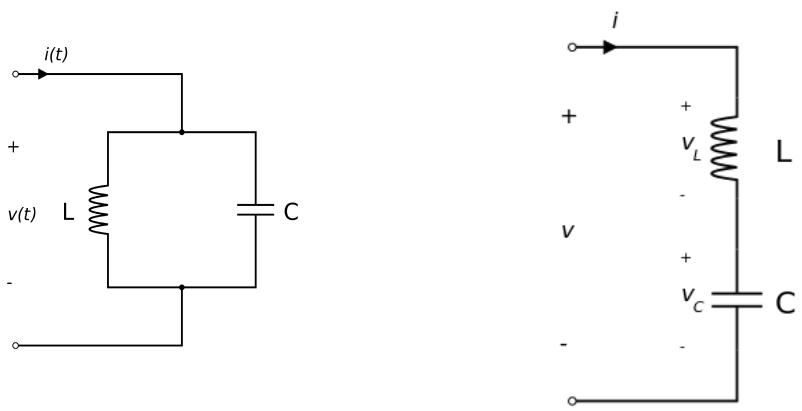
- Electric vehicles.

  Currently there are a lot of car companies releasing their electrical models which require high power to charge (in the range of 2-5 kW). Most of these cars require a plug to be inserted for charging, however, some cars use IPT as their charging method.

Two ways exist to use IPT as power transfer, the first is by inserting the transmitting coil in the receiving coil, going towards the original plug and socket, but increasing safety even more [20] while also removing the chance of electrical sparks. The second uses charging pads which are specifically designed to aim the magnetic field the right way [1] [13]. These pads are designed to cope with relatively big displacements while retaining efficiency.

In these applications, IPT has been implemented in different ways. To increase efficiency of these applications, resonance is used. This means a capacitor is connected in either parallel or series with the coil, an inductor, as seen in figure 5. This creates the following possibilities:

- SS; a capacitor in series with the transmitter and a capacitor in series with the receiver.
- SP; a capacitor in series with the transmitter and a capacitor in parallel with the receiver.
- PP; a capacitor in parallel with the transmitter and a capacitor in parallel with the receiver.
- PS; a capacitor in parallel with the transmitter and a capacitor in series with the receiver.




**(a)** Parrallel LC-circuit. **(b)** Serie LC-circuit.

**Figure 5:** Different LC-circuit for use in wireless power transfer.

Equation 1 and 2 show the equations to calculate the reactance, $X$ [$\Omega$], of a capacitor and an inductor. The $j$ shows it is an imaginary value, $f$ is the frequency [$Hz$], $C$ the capacitance [$F$] and $L$ the inductance [$H$]. As the frequency affects the reactance linearly for the inductor and inverse for the capacitor (thus not linear), a single frequency at which the reactance is equal can be found, using equation 3. The frequency at which this happens is called the resonant frequency, $f_0$ [$Hz$].

$$X_C = \frac{1}{j2\pi fC} \tag{1}$$

$$X_L = j2\pi fL \tag{2}$$

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \tag{3}$$

The impedance can be calculated the same way resistances are added together, but with an imaginary part. With the reactance of an inductor being positive and the reactance of a capacitor being negative. The real part consists of only the resistance in the circuit. In an ideal situation at resonant frequency, this would cause the series topology to have zero reactance, the parallel topology would have infinite reactance. practical applications will never have zero resistance however, and the frequency will never be the precise value. Research [15] shows the effects of the topology on power transfers, a table can be found in appendix B and shows the differences per topology.

## 3.2 Docking behaviour

Getting the robot at the docking station and coupling the chosen power transfer method requires a system which guides it. The docking behaviour includes all systems which make the coupling of the power transfer system possible. For this reason, some options might be combined at the end supporting each other, this will be further discussed in the next chapter. For this subsystem it is assumed that the robot can navigate close to the docking station using its available navigation.

As the option wireless power transfer option, IPT, does not have contacts, yet needs to be aligned, for convenience the transmitter and receiver are called contacts from here on. Options which are strongly related to the same principle, yet differ enough to be seen as different options, are numbered through brackets with a number behind the option.

### 3.2.1 Vision

The option to use the vision hardware available on most, if not all, robots makes for a software option for the robot to find the docking station. Vision however, is a broad term which is why it is split in several options.

- Object recognition (1)

  An option would be to recognize the docking station and know its orientation and position relative to the robot. To simplify this process, a feature, like a qr-code, could be added to the docking station which makes it easier for recognition. As recognition of features is already available, the only required addition to the software would estimating the relative position.

- Light field (2)

  Vision could also be used similar to docking process of the earlier mentioned robot 'Roomba' [7]. This would mean the docking station needs a non-visible light emitter to create a field only visible to the robot. This field would then guide the robot towards the docking station. However, when a field is used, the docking direction is not specified thus requiring an additional part to align correctly.

- Line follower (3)

  A line follower could also be used to create a path towards the docking station. This does require a small hardware modifications for most robots. As the line should not be visible, non-visible light should be used to project a line on the ground. The small hardware modifications consists of a line follower, which basically consists of two light detection sensors.

### 3.2.2 Navigation

Depending on the quality of the indoor navigation of the robot, it could be used to bring the robot close to the docking station. If the robot is close enough, hardware could be used to guide it towards the optimal position. This would be similar to the mentioned robot 'Double'. For the hardware several options can be thought of as well.

- Wheel guiding (1)

  The wheels can be guided so the robot enters the docking in a specific way. This would require the wheels, on the side of where the contacts for docking are, to be steerable and be able to cope with slight changes.

- Body guiding (2)

  Instead of the wheels, the whole body of the robot can be guided. This limits the flexibility of the docking station to a single body however.

- Contact guiding (3)

  The contacts could be guided to the correct position requiring at least one side of the contacts to be moveable. Depending on the size of the contacts, this hardware part can be made quite small compared to the previous mentioned methods.

Even though the mentioned options can all be made using actuators, using the force exerted by the robot is preferred.

# 4 Design

This chapter will start with analysing the importance of several aspects for the two subsystem described in the previous section. From this analysis the options will be judged based on a morphological analysis, after which combinations are made to create full concepts. These concepts will be further developed and described at the end of this chapter.

## 4.1 Option analaysis

For the morphological analysis, criteria have to be chosen on which to judge the subsystems. All criteria will be chosen based on the requirements, which will be indicated by their ID number in brackets, or on problems existing with current docking stations, as described in the assignment chapter. As not all criteria should have the same influence, weight factors are created using tables [3]. The first subsystem for which criteria and weight factors will be decided upon, is power transfer.

Efficiency (E05) is the first criterion, as the power consumption of a system is an important part of every electrical system. As the power needed by the robot (E03) is set as fact, reducing losses in the charging system is the only way to increase the efficiency. Increasing the efficiency above the set requirement would improve the performance and quality of the total system. With increased efficiency, the charging time (E04) will also decrease significantly.

As the docking station should not exceed the given weight of $3kg$ (M16) and only $0.5kg$ can be added to the robot (M15), weight becomes a criterion.

Even though neither the requirements nor the assignment specify anything about manufacturing, the use of easily available components is preferred above creating custom components. By using available components, costs and invested time will be reduced.

As the current docking station has problems with sparks between contacts, caused by slight alignment errors, alignment of the charging system is an important criterion. By increasing the margin at which the robot can dock, different options, and later, different combinations open up.

The last criterion is the robustness of the system. In this case, the robustness describes the possibility of damage on the system and the effect of this damage for future charging. Current docking station, as previously mentioned, have arcs between contacts that can be destructive for the contacts, which in turn results in lower efficiencies of the total system.

The corresponding weights for the criteria are determined using table 1. In this table a '1' indicates the column is equal or more important than the row, while the '0' indicates it is less important. The final weights of the aspect of the column are calculated by adding all numbers in the column after which one is added to prevent a weight of zero. Whether or not some criteria are more important than others was decided by the use of the requirements and further influenced by the supervisors.

For the final approach, the criterion repeatability is chosen to represent the possibility of making errors. As some charging systems, like the contacts used in the current docking station, do not cope well with errors and can leave permanent damage, the importance of good repeatability is high.

The next criterion used is aligning, which basically represents the acceptable errors of the alignment. Having a system that is capable of handling errors without significant reductions in other aspects of the system, means the quality of the aligning system can be reduced.

| | Efficiency | Weight | Manufac-turing | Aligning | Robustness |
|---|---|---|---|---|---|
| Efficiency | - | 0 | 0 | 1 | 0 |
| Weight | 1 | - | 0 | 1 | 0 |
| Manufacturing | 1 | 1 | - | 1 | 1 |
| Aligning | 0 | 0 | 0 | - | 0 |
| Robustness | 1 | 1 | 1 | 1 | - |
| Weight factor | 4 | 3 | 2 | 5 | 2 |

**Table 1:** Weight factors used for the morphological analysis of power transfer.

Flexibility is based on the requirement to make the docking station adaptable (M08). As the docking station has to be able to be adapted to several robots, it should be easy to modify the way wanted. This especially focusses on the amount of actions and knowledge required to change between different robots.

Manufacturing is a criterion seen before, and is not any different for this subsystem; using available components reduces costs and invested time. The weight table shown in table 2 works the same as previously described and gives the weight factors of the alignment.

| | Repeatability | Aligning | Flexibility | Manufacturing |
|---|---|---|---|---|
| Repeatability | - | 1 | 1 | 1 |
| Aligning | 1 | - | 0 | 0 |
| Flexibility | 1 | 1 | - | 1 |
| Manufacturing | 1 | 1 | 0 | - |
| Weight factor | 4 | 4 | 2 | 3 |

**Table 2:** Weight factors used for the morphological analysis of alignment.

For the analysis of the options, two separate morphological tables have been made. The morphological tables give a value from 1-6 for all aspects and every option. Multiplying the given values with the weight of the criterion, determined in the previous paragraph, and adding it all together, results in an amount which indicates the level of performance. As this overview is susceptible to personal preference, the upper 10% is accepted as good enough and can be used for final concepts.

The first overview, shown in table 3 is made for the power transfer. The table shows that there are three options which are in the upper 10%; inductive power transfer (IPT), flat contacts and shaped contacts.

| Scale 1-6 | Weight | IPT | Plug and socket | Flat contacts | Shaped contacts | Sliding contacts |
|---|---|---|---|---|---|---|
| Efficiency | 4 | 3 | 6 | 6 | 2 | 3 |
| Weight | 3 | 4 | 6 | 6 | 6 | 5 |
| Manufacturing | 2 | 4 | 4 | 6 | 5 | 3 |
| Aligning | 5 | 5 | 1 | 3 | 5 | 6 |
| Robustness | 2 | 6 | 4 | 4 | 3 | 1 |
| Total | 96 | 69 | 63 | 77 | 67 | 65 |
| Percentage | 100% | 72% | 66% | 80% | 70% | 68% |

**Table 3:** Morphological analysis for power transfer.

The second one, shown in table 4 gives an overview for the final approach options. The options are the following:

For vision:

1. Object recognition
2. Light field
3. Line follower

For navigation:

1. Wheel guiding
2. Body guiding
3. Contact guiding

| Scale 1-6 | Weight | Vision | | | Navigation | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 |
| Repeatability | 4 | 5 | 3 | 4 | 6 | 4 | 6 |
| Aligning | 3 | 3 | 2 | 4 | 3 | 5 | 5 |
| Flexibility | 2 | 5 | 3 | 5 | 2 | 1 | 3 |
| Manufacturing | 1 | 6 | 4 | 2 | 2 | 1 | 3 |
| Total | 60 | 45 | 28 | 40 | 39 | 34 | 48 |
| Percentage | 100% | 75% | 47% | 67% | 65% | 57% | 80% |

**Table 4:** Morphological analysis for the final approach.

From the morphological analysis, two options are in the upper 10%, the vision option object recognition and the navigation option contact guiding.

## 4.2 Design: Power transfer

From the previous paragraph, three power transfer methods were picked as best options and now developed further. The three options are IPT, the flat contacts and the shaped contacts.

### 4.2.1 IPT

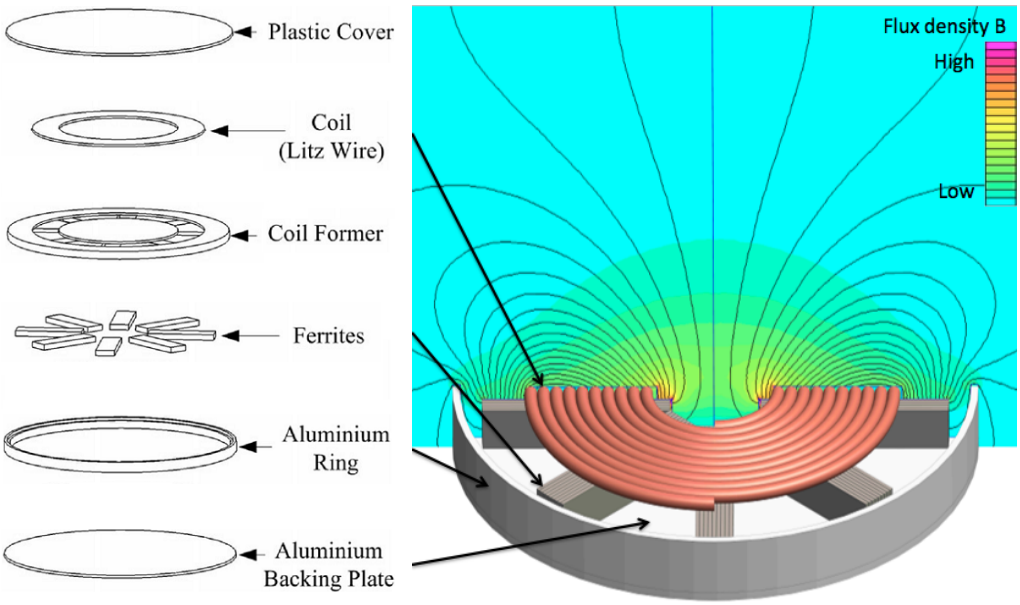
As stated in paragraph 2.3.3 there are several applications for which IPT is researched and developed. As the requirements state the efficiency should be atleast $80\%$, the power transferred should be $300W$

($30V$ x $10A$) and a wireless charging range of $5mm$ is sufficient, the design specifications of the electric vehicles come closest.

The first option for optimal efficiency would be the toroidal coil with the primary coil and secondary coil split. By inserting the primary in the secondary, an electrical current will be induced [20]. The reason for this option to not be further designed is that in the morphological analysis plug and socket, the electrical equivalent, was not in the upper $10\%$ because of the bad aligning capabilities.

The second mentioned option, was the charging pad. The charging pads are designed for larger distances while trying to retain the efficiency. This is done by using a ferrite core underneath the coil, while an aluminium outer ring blocks the flux on the sides. Figure 6a shows a schematic example of a charging pad, the aluminium bottom is to cool the cores and to stop some of the flux. Figure 6b shows a solid works model of the coil with its corresponding magnetic field, which was analysed with the program FEMM [12].

The blue colour shows a low flux density, $B$ $(T)$, while the orange inside of the coil shows a high flux density, the lines represent the flow of the flux. As the figure shows, the flux density is highest just above the coil. Having a similar receiving coil on top of the transmitting coil, would cause the most efficient power transfer. However, small alignment errors might occur that would cause the efficiency to drop. In a later stage, this will be thoroughly tested to specify the maximum alignment error while still reaching the $80\%$ efficiency requirement.




**(a)** Schematic overview of a charging pad.

**(b)** Charging pad modeled in solid works, analysed in FEMM.

**Figure 6:** Charging pad used in wirelessly charging of electric vehicles.

Most researches which use these charging pads do not define their design choices. However, some assumptions can be made through these papers.

- Core

  Even though the core in normal transformers is designed specifically for the application, [13] concludes the disk core used in earlier charging pads can be a lot smaller without huge differences. Even though [1] gives an optimal layout and optimal bar shapes for the charging pad, the material choice is not specified. For this design, a core material which is well documented and available was chosen [6] to be able to test situations for this specific application, at a later stage this can be optimized. The bar specifications can be found in appendix C.

- Coil

  The coil is often designed for a particular inductance $[L]$. This inductance can be flexible thus the design of the coil can be made without this goal. As the bars of the core material have a length of $26.4mm$, the windings should not exceed this according to the optimal design shown in [1]. As a current of $10A$ will go through the windings, it was chosen to use AWG 13 wire with a diameter of $1.8mm$. An estimate is to cover a maximum of $20mm$ of the core, thus resulting in 11 to 12 windings.

- Frequency

  With the already available PCB's at DBL which were created in a previous internship [8] the frequency is set at $10kHz$ as start and can be increased up to $100kHz$ if needed.

Following from these assumptions, the equation 4 can be filled in. With $E$ being the effective voltage ($30V$), $f$ the frequency ($10kHz$), $B$ the flux density ($T$), $A$ the effective core area ($m^2$) and N the coil turns (12). This gives two unknown values, $B$ and $A$.

$$E = 4.44fBAN \tag{4}$$

According to the data sheet of the core [6], the effective area per core is $40.3mm^2$. Equation 5 gives the relation ship between $B$ and the permeability, with $\mu_0$ as the permeability of a vacuum ($4\pi * 10^{-7}H/m$), $\mu_r$ the permeability of the material compared to the vacuum and $H$ the magnetic field strength ($A/m$).

$$B = \mu_0\mu_r H \tag{5}$$

When $\mu_r$ increases, less $H$ is needed to get the same $B$. For this reason the ideal situation is the part where the permeability increases the fastest over a flux density of $0$ to $x$. The optimal flux density can be found through the graph shown in figure 7. The graph shows a linear increasing permeability till about $150mT$, for higher values, the permeability increase declines. This decline means that the ratio of permeability increase per flux density goes down. Thus the flux density is set at $150mT$. Now the only unknown variable is $A$ which can be calculated.

$$A = \frac{30}{4.44 * 10 * 10^3 * 150 * 10^{-3} * 12} * 10^6 = 375mm^2 \tag{6}$$

$A$ is equal to more than 9 times the effective area for a single core, thus 10 cores per coils should be started with. When the frequency is increased, the needed area will be reduced thus reducing the amount of cores needed. As the coils are made with a testing purpose in mind, it was chosen to not add the receiving coil cores to the core area to prevent heating when the receiving coil is not present.

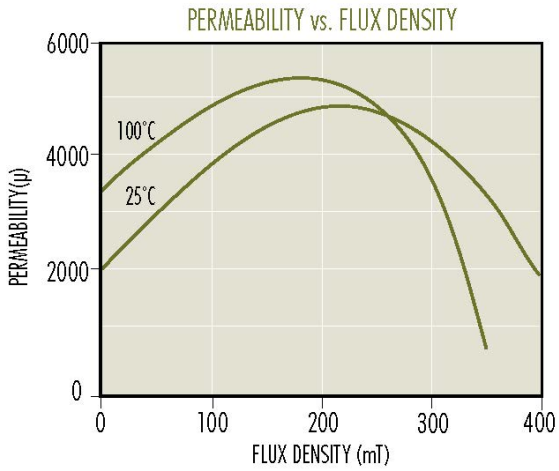



**Figure 7:** Permeability compared to the flux density for two temperatures.

In a later stage this could mean less cores can be used per coil, as a requirement is to only charge when the robot is in proximity.

As mentioned earlier, the topology of the electrical circuit has four possibilities, two for each coil (SS,SP,PP,PS). In [15] the characteristics of the four possibilities are compared, the table can be found in appendix B. The table makes it possible to decide on the most convenient topology for a specific application. If the transmitting coil uses a parallel topology, the AC source has to have a high voltage or be a current source for optimal transfer. Thus the transmitter topology has been chosen to be in series.

Especially for testing purposes, it is good to be able to vary the frequency. As the table in appendix B shows, frequency variations of the SS topology decrease the efficiency more than those of the SP topology. Figure 8 [15] shows the differences in more detail and shows the efficiency versus a normalised frequency, which means all frequencies are divided by a constant frequency value (in this figure the resonant frequency of $100kHz$). Based on this graph, the choice was made to use an SS topology because the efficiency is significantly higher, even if the frequency becomes higher than the resonant frequency. As the efficiency drops significantly when the used frequency is below the resonant frequency, the resonant frequency should always be chosen below the operating frequency.

### 4.2.2 Flat and shaped contacts

As stated earlier, the chosen contact shapes were basic shapes which, if needed, could be combined to cancel out disadvantages. Even though both contact shapes would suffice without combining, according to the morphological analysis shown in table 4, both have a lack in a specific area. The flat contacts lack aligning capabilities while the shaped contacts lack the efficiency due to the small contact area.

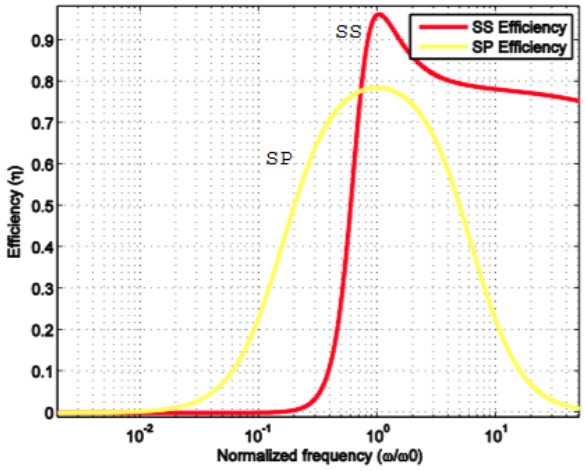
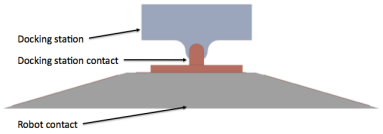
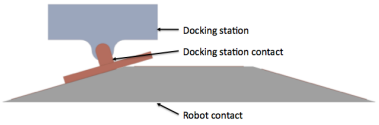



**Figure 8:** Graph that shows the normalised frequency versus the efficiency for a SS and a SP topology.

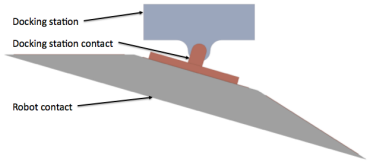
To combine the two contacts, the shaped contact can be approximated using flat sides. This would cause the contact area to increase compared to the original shaped contact area, and the aligning to increase slightly because of several options. Figure 9 shows three situations where three sides with flat contacts are used to mate the contacts.




**(a)** Aligning with the robot moving straight towards the docking station without any alignment error, this would be the ideal situation.




**(b)** Aligning with the robot moving straight towards the docking station with a deviation causing an alignment error.



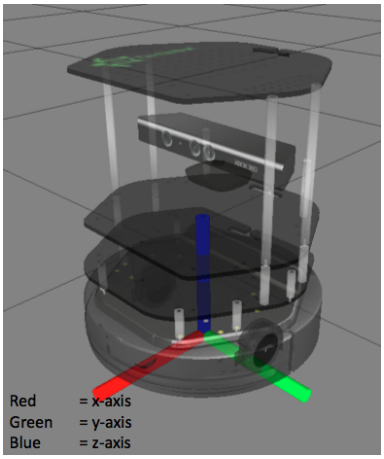

**(c)** Aligning with the robot moving towards the docking station from the side, while aiming for the contacts.

**Figure 9:** Top view of three situations where the contacts of the robot mate the movable contacts of the docking station.

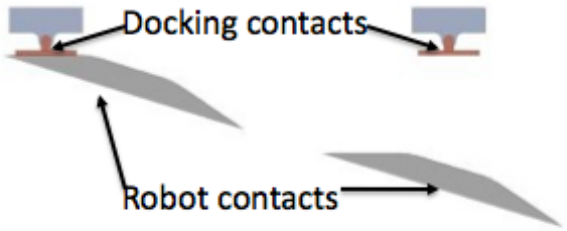
In this example the construction used in the RSS design, which causes the contacts to be able to slightly move, is used to prevent the contacts from getting stuck on a corner of the contact. As shown, the robot can dock from several sides without any problems. Figure 9b shows the possibility exists that, with a maximum of half the contact, part of the contact does not mate. When further designing these contacts, this should be taken in account.

Further it should be noted that in the RSS design, the contacts were aligned horizontally. This causes there to be an extra alignment problem which can easily be eliminated by aligning the contacts vertically.

For describing this problem figure 10a shows a model of the Turtlebot with thee axises, red the x-axis, green the y-axis and blue the z-axis. Because a robot is designed to not fall over, rotations around the x-and y-axis are minimised. The rotation around the z-axis however is controlled by the robots steering mechanism. Figure 10b shows an example of the robot docking from the side with the previously described concept. The contacts are aligned horizontally, which causes the robot to connect with only one of its contacts. If the contacts are aligned vertically, this problem will not occur, as the positions shown in figure 9 show.




**(a)** Turtlebot visualisation with co-ordinate system.




**(b)** Aligning with the robot moving towards the docking station from the side, while aiming for the contacts.

**Figure 10:** Top view of horizontally aligned contacts docking from the side.

## 4.3 Design: Docking behaviour

From the morphological analysis in paragraph 4.1, two methods were in the upper 10% and selected to be designed further. These options are object recognition and contact guiding.

### 4.3.1 Vision: Object recognition

The object recognition option requires the vision system on the robot to recognise the docking station. While it is possible to recognise the whole docking station, it is also possible to recognise only a single feature on the docking station.

In this case, it is preferred to have a feature which has a simple shape yet is not present on other objects. The QR-code would be a solution for this, as it is square shaped and part of this square is used to make the QR-code unique. Finding the orientation and distance relative to the docking station is done by using transformation matrices. Using these matrices, the QR-code can be transformed till it has its original shape and size again. There are three parts of the transformation matrix which have to be calculated. The shear of the image is caused by viewing the feature under an angle. Using the shear, the orientation of the robots viewing direction relative to the docking station can be found. The

translation of the QR-code is used to calculate the amount of rotation needed to perfectly aim at the QR-code, the translation is done relative to the whole image to find the deviation to the sides. The scale of the images changes with the distance towards the docking station. When a reference point is used at a known distance and the scaling is done to make the QR-code the same size, basic geometry can be used to calculate the distance, as will be shown in section 4.5.

When using feature recognition the feature has to be always visible. As the robot moves towards the docking station, the view of most of the docking station will be blocked. To be able to keep aligning till the robot is docked, the QR-code should be placed at the same height as robot vision system. By having the QR-code on the same height, the complexity of calculating the transformation matrix is reduced as the shear can only be caused by one axis.

### 4.3.2 Navigation: Contact guiding

With contact guiding, one side of the contacts is made in such a way that it can adjust to the orientation of the contacts on the other side. The first design, designed around the coils from [8] and shown in figure 11a, uses a sunk in contact on the dock and a counter part on the robot.

The force the robot exerts to dock causes the contacts to be pressed against each other. Through using this force to improve the alignment, the efficiency can be increased. With the contact shown in 11a the whole robot is moved, which is not preferable as it might disturb equipped sensors like encoders on the wheels.

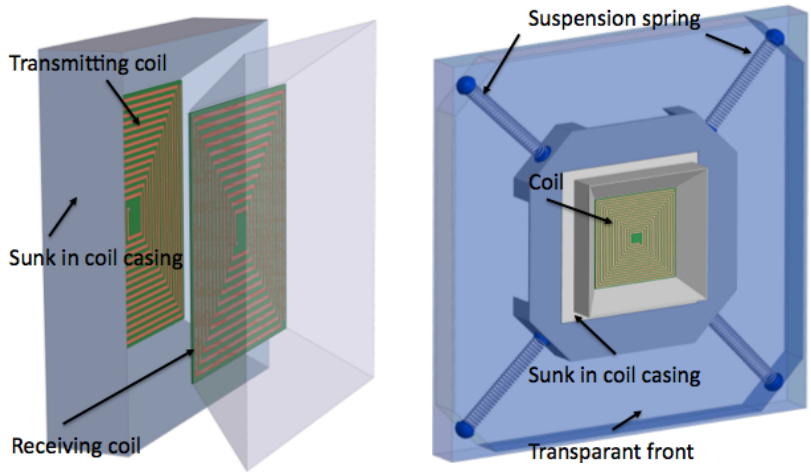
To make the contact move instead of the whole robot, the contact has to be made moveable on a single side. The contact on the docking station is made moveable as the robot should be changed as little as possible. Figure 11b shows the design of this moveable contact. The contact is hung in a suspension system attached to the docking station with four springs. The front of the docking station keeps the movements limited and controlled. When the robot moves away, the springs can go to their original state again.

The suspension system is made to counter errors in vertical and horizontal direction or a combination. Two springs could be removed to counter errors in only a single direction, the robot 'Double' [17] is an example of such contact guiding as it only counters horizontal alignment errors.

## 4.4 Concepts

The described methods can be used to combine into concepts. As there are four concepts possible if the shaped and flat contacts are combined, all concepts can be evaluated. Combinations of subsystem options can be made, which will be evaluated at a later stage. A brief overview of the concepts can be found below.

- Concept 1: IPT with object recognition.
- Concept 2: IPT with contact guiding.
- Concept 3: Contacts with object recognition.
- Concept 4: Contacts with contact guiding.




**(a)** Sunk in coil with its, transparant, counter part for aligning.

**(b)** Concept showing the holder for the transmitter coil in it with a suspension system holding it in the middle.

**Figure 11:** Concept of using contact guiding as aligning system.

As a requirement that was cancelled was added again due to safety reasons and plausibility of a working concept, the docking station has to charge wirelessly. This leaves only the IPT system which will thus be be realised, removing concept 3 and 4 from the possible concepts. Concept 1 and 2 are different because of the docking behaviour using object recognition and contact guiding, because object recognition is software based and can be applied to all robots with a vision system, object recognition is realised first. At a later stage, contact guiding can be added to improve efficiency, or if needed, to improve the alignment capabilities.

## 4.5 Design: Object recognition

The software for the robot should consist of a method to dock. Each robot has different hardware, which is why for realisation a specific robot, the Turtlebot, has been chosen. As not all robots have the same sensors and mechanical features, whenever the software uses a specific component, several other solutions are given to get to the same result. It is assumed the robot can find the area where the docking station is located, in the next section more precise values will be given.

The choice has been made to execute the docking in several steps, eliminating problems one at a time. This results in a modular software design where steps can be replaced if different robots are used. Figure 12 shows a flow chart of these steps which will be explained in the coming paragraphs.

Finding the orientation and distance is done simultaneously. As the charging system and the feature for recognition are located at the front panel of the docking station, orientation and position are determined relative to this panel. The orientation is affected by two different factors, the robots location and the direction it is faced towards.

Through making sure the robot is always aimed at the docking station, the robots face direction is known. The homing is done by getting the feature in the horizontal centre of the camera.

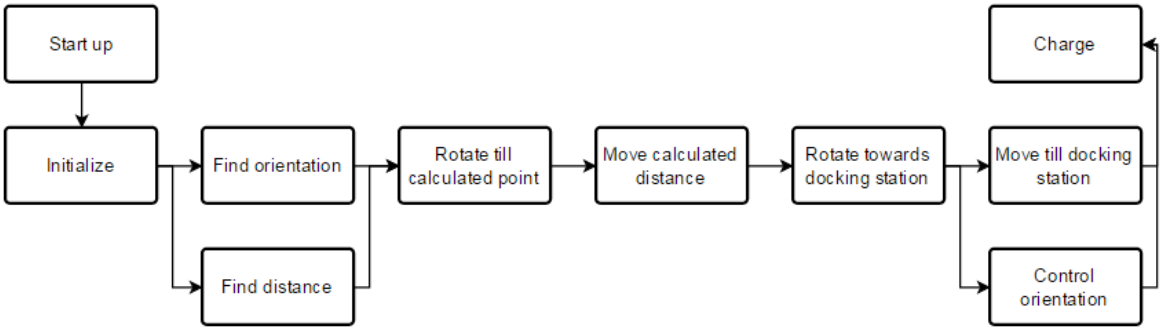



**Figure 12:** Flowchart of the docking software.

The feature is located on the frontal panel, by making the features x and y ratio correct, the angle relative to the flat surface can be calculated.

Now only the distance towards the docking station has to be measured. This is done through using the Kinect sensor equipped on the Turtlebot. When there is no Kinect available on the robot, any other camera system or distance measuring device can be used.

Figure 13 shows the two cases which can happen when the robot is docking. In figure 13a $y$ and $a$ are added together because the robot is further away from the docking station than point $P$. Figure 13b shows the situation where the robot is closer to the docking station than point $P$ and $a$ has to be subtracted from $y$. By measuring $\alpha$ and $d$ through the vision system and using distance $y$ as distance to the virtual reference point $P$, the robot can dock. The wanted values are the rotation to first point $P$, called $\gamma$ from here on, and the distance $c$.

Two assumptions were made for determining the situation.

- The cases which are mirrored versions of the shown situations are handled the same way. Turning direction is later determined through a different function.
- $y$ cannot be smaller than zero. In practical application this would mean the docking stations feature is not visible.

Because $b$ and the docking station are parallel it is known that for figure 13a:

$$\angle\alpha = \angle\beta_{12} \tag{7}$$

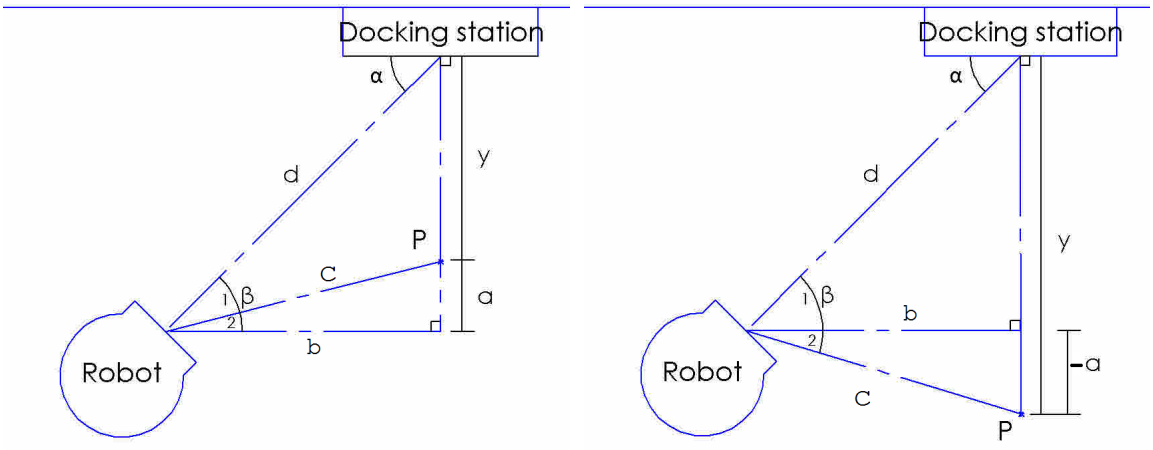
For the second case, shown in figure 13b:

$$\angle\alpha = \angle\beta_{1} \tag{8}$$

Through $\alpha$ and $d$ the length of $a$ and $b$ can be calculated:

$$a = d\cos\alpha - y \tag{9}$$

$$b = d\sin\alpha \tag{10}$$




**(a)** First situation with y+a greater than y.

**(b)** Second situation with y+a smaller than y.

**Figure 13:** Schematic topview of two possible situations.

With $a$ and $b$ the wanted values $\gamma$ and $c$ can be calculated according to:

$$c = \sqrt{a^2 + b^2} \tag{11}$$

$$\angle\gamma = \angle\alpha - \tan^{-1}(\frac{a}{b}) \tag{12}$$

The formulas written with the variables given at the start:

$$c = \sqrt{(d\cos\alpha - y)^2 + (d\sin\alpha)^2} \tag{13}$$

$$\gamma = \angle\alpha - \tan^{-1}(\frac{d\cos\alpha - y}{d\sin\alpha}) \tag{14}$$

with $a$ being positive in the case shown in figure 13a and negative in the case shown in figure 13b:

$$\angle\alpha = \begin{cases} \beta_{12}, & \text{if } y + a \geq y \\ \beta_1, & \text{if } y + a < y \end{cases}$$

Through first rotating the robot with $\gamma$, then moving it with distance $c$ the robot is placed on point $P$. One last rotation, $\phi$, has to rotate the robot in the opposite direction to face the docking station and is calculated by the following equation:

$$\phi = \gamma + (90 - \angle\alpha) \tag{15}$$

The Turtlebot is able to turn around its own axis without any linear movement. For robots which do not have this feature, an arc has to be calculated to get to the correct spot at the calculated angle.

The last step before the robot is docked requires the vision system. By keeping the feature of the docking in the horizontal center of the videostream, the aligning can be optimized. This is done through using a control loop of which a schematic can be seen in figure 14. The control loop takes the wanted orientation and the current orientation to calculate the error. The error is used in the PID controller, after which it is multiplied with a constant value to create a correct angular velocity output.

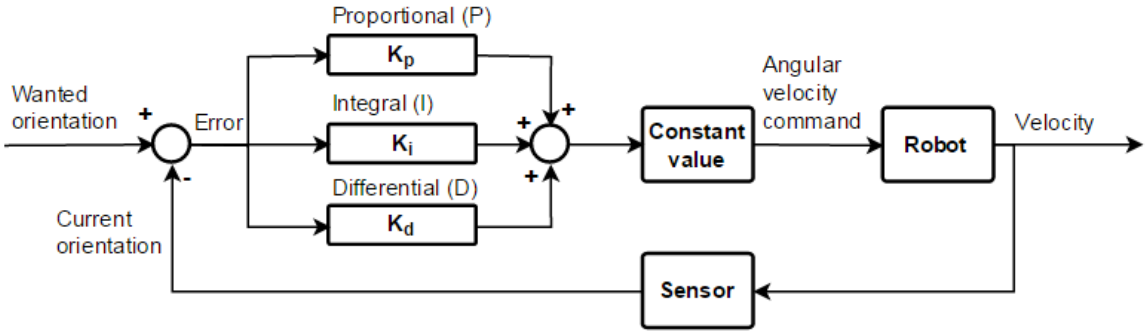



**Figure 14:** PID controller used to make the robot able to slightly adjust

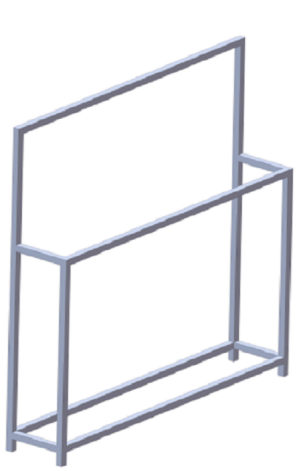
## 4.6 Docking station layout

For the docking station to be complete, all components need to be integrated in the area defined by the requirements on the size of the docking station $500mm$ x $500mm$ x $100mm$ (w x h x d). Even though the frame and housing will not be developed with the first prototype, the idea behind it is shown in this paragraph. There are several parts which have to be taken in account:

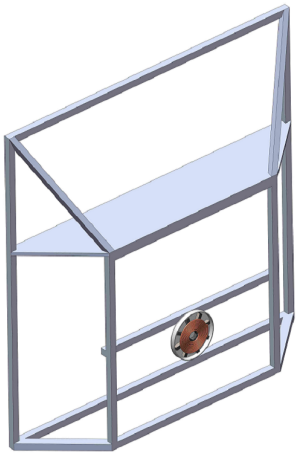
- Frame; Makes the docking station stable and strong enough.
- Hardware; The hardware that is required for the full concept to work.
- Housing; To finish the design, the housing of the docking station.

### 4.6.1 Frame

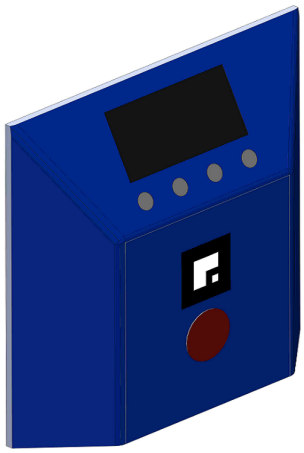
For the frame a strong design has to be made. It has to complement the needed hardware parts to make instalment of the parts as easy as possible. At the start, the ground frame will be the maximum size of $500mm$ x $100mm$ (w x d). The back frame will be the maximum size as well, $500mm$ x $500mm$ (w x h), however, the frontal frame can be smaller as the height of the coils on the robot will be lower than $500mm$. By creating a smaller front frame, a chamfer can be created on top of the docking station, that can be used for interaction hardware. For stability, small adjustable foots are needed as most floors are not perfectly flat. Figure 15a shows a simple frame which shows the maximum size of the frame without any size changes related to the coil size.



**(a)** Maximum frame size with frontal plane lower than backplane.



**(b)** Added coil, resized frontal plane and added plate above frontal plane.



**(c)** An example of what the final design could look like with interface and feature.

**Figure 15:** Design of the layout of the docking station.

### 4.6.2 Hardware

The coil for the IPT will be placed on the frontal plane. Some additional electronic hardware is needed to:

- transform the input signal ($230V$ at $50Hz$) to the needed signal, $30V$ at $10A$ AC at $10 - 100kHz$.
- check the output signal to correspond with the wanted value.
- control the interface on the docking station and all the other hardware.

This hardware will mainly consist of PCB's. Because of the flux generated from the coils, the backplane cannot be used because of possible disturbances, even with an aluminium plate. As the interface has to be on top of the docking station, it was decided to put all the hardware on the plane above and perpendicular to the frontal plane. Figure 15b shows how the docking station could look if a plate and the charging mechanism are added.

### 4.6.3 Housing

The last part of the docking station is the housing. Figure 15c shows a possible option for the housing. The interface is added on the chamfer on top of the docking station for easy access. The feature is added just above the charging mechanism, but should be changed according to the height of the robots vision system. The red circle in the figure indicates where the charging mechanism is located.

# 5 Realisation

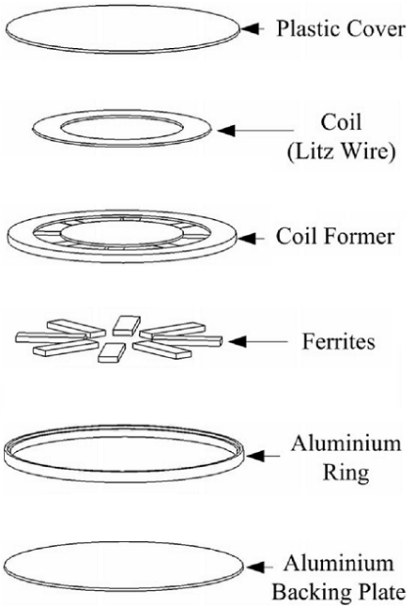
In this chapter the realisation of the system is explained in detail. Because testing is needed in between realisation, the chapter also describes testing of the system.

## 5.1 Realisation: Power transfer

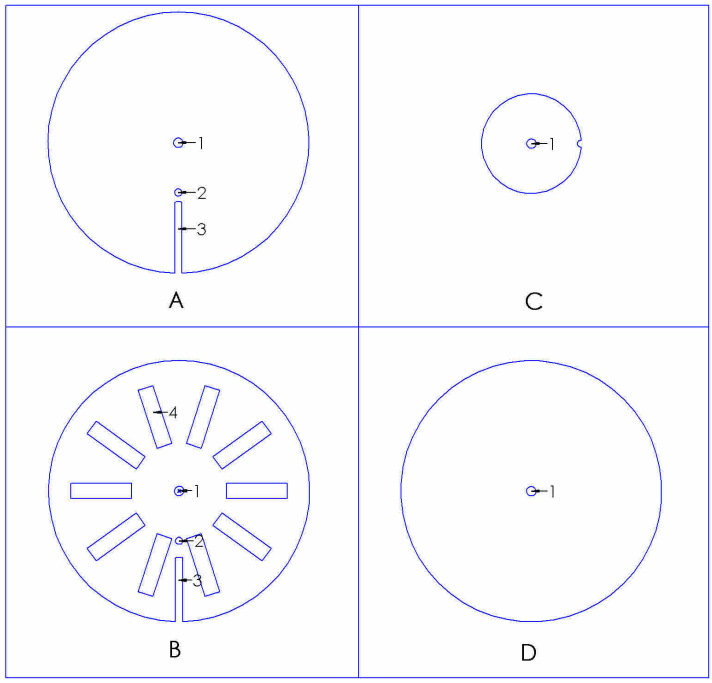
The power transfer consists of two parts, the coils and the electrical circuitry. Because the electrical circuitry is available as PCB at DBL created in [8], the focus will be on the coil and making it resonant.

### 5.1.1 The coil

The power transfer has been made according to charging pad schematic shown in figure 16a. Two similar pads are created for the transmitting and receiving coil and consist of several parts. The parts will be explained according to the schematic in the figure, from the bottom to the top.




**(a)** Schematic pad




**(b)** Four drawings: A the backing plate, B a core holder, C a coil former, D a plastic cover.

**Figure 16:** Coil schematic and drawings

The backing plate is created by laser cutting a plate, in this case plastic, with holes as shown in figure 16 drawing A. The gap (3) is created for the wire to go through after the coil is wound. The hole next to the gap (2) is used to put the wire of the coil through before winding. The center hole (1) is used for adding all plates together.

To position the cores correctly, a core holder, seen at B in the figure, has been laser cut. The core holder has ten cut out squares (4), which are the size of a single core, and are placed around an inner circle with a diameter of $40mm$. The gap and the two holes (1 to 3), as seen in the ground plate, are added as well.

When all cores are inserted in the core holder, a coil former is put on top. The coil former is a round plate, as seen at C in the figure, with a diameter of $42mm$ and a single hole in the center for montage. The plate has a thickness of $2mm$ as the wire diameter is $1.8mm$. Because the former is slightly bigger than the inner diameter of the core holder, the coil former lies on top of the cores.

The inner wire of the coil can be inserted in the hole (2) and bent in a right angle. The plastic cover of the coil can then be added and a sunk-in screw can be used to keep the whole coil design together. The coil can then be wound around the coil former and the outer wire can be pulled through the gap (3).

The end result can be seen in figure 17, showing the receiving coil. The transmitting coil has a single winding less. Compared to the schematic pad, the aluminium parts are missing and a core holder was added. The aluminium parts should be added when they are used in the final docking station to prevent disturbances. The total thickness of the coil can be calculated through adding the thickness of the backing plate, the core height, the thickness of the coil former and the thickness of the plastic cover. In the case of the created coils this results in approximately $15.5mm$

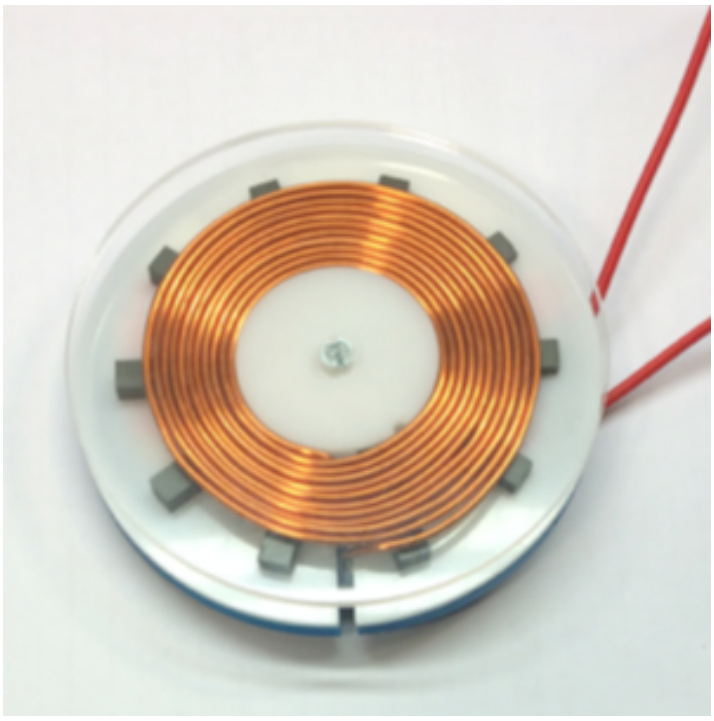


**Figure 17:** Photo of the final coil.

### 5.1.2 Resonance

As mentioned earlier, a series topology for both coils is wanted. To calculate the resonance, the inductance of the coils has to be known. As the inductance can change varying with the frequency, the measurement is an estimate of the inductance. The measurement was done using an L-measurement device while the coil had a high frequency AC ($100kHz$) on it. With the inductance, the capacitance can be calculated through equation 16 with $f_0$ being the resonance $[Hz]$, $L$ the inductance $[H]$ and $C$ the capacitance $[F]$.

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \tag{16}$$

The measured inductance of the transmitter coil at $100kHz$ is $11.2\mu H$; the measured inductance of the receiving coil at $100kHz$ is $13.6\mu H$. Table 5 shows the the capacitance values for frequencies in the range of 10 to $100kHz$ calculated with the measured inductance.

| Frequency $[kHz]$ | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| Transmitter Capacitance $[\mu F]$ | 22.61 | 5.65 | 2.51 | 1.41 | 0.91 | 0.62 | 0.46 | 0.35 | 0.28 | 0.23 |
| Receiver Capacitance $[\mu F]$ | 18.62 | 4.65 | 2.07 | 1.16 | 0.75 | 0.51 | 0.38 | 0.29 | 0.23 | 0.19 |

**Table 5:** Capacitance for the transmitter and the receiver for several frequencies.

To control the frequency, two MOSFETS are used in a half bridge. A schematic of the half bridge can be seen in figure 18. After the energy is transferred the AC current is turned into DC again through a AC/DC converter.

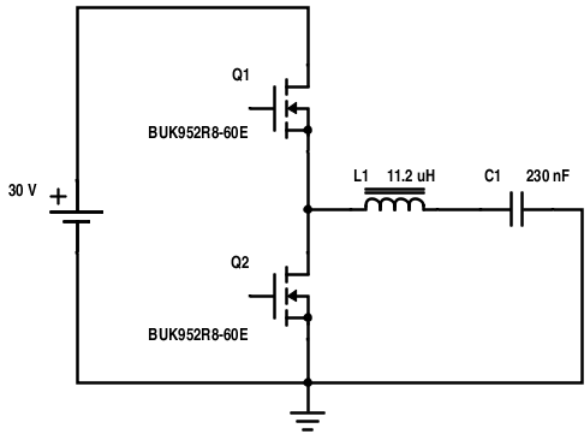



**Figure 18:** Simplified electrical schematic of the half bridge used for the transmitter coil.

### 5.1.3 Testing the power transfer

To validate the requirements and to find the limits of the coil for which the requirements can be met, measurements have been done with varying variables. The test is done using the coil shown in the previous paragraph.

Measurements of the coils will be done slightly above the resonant frequency to make sure the efficiency does not drop significantly. A resonant frequency of $40kHz$ was used for testing, which means a capacitance of $1.41\mu F$ and $1.16\mu F$ was added to the transmitter and receiver coil respectively. The operating frequency was set at $44kHz$, thus above the calculated resonance. To measure the output of the receiver correctly, a load was applied. The changing variables are the following:

- $d$ $[mm]$; Distance between two coils with measurements ranging from $0$ to $22mm$ with intervals of approximately $4mm$. An offset is created by the front panels of the coil thus increasing the range by two times the thickness of the front panel, $6mm$ in total.
- $x$ in $[mm]$; Misalignment between the center of the two coils in X direction ranging from $0$ to $40mm$ with intervals of $5mm$.

The values which are measured or set to calculate the efficiency can be found below:

- $U_i$ $[V]$; Input voltage will be set at $24V$.
- $I_i$ $[A]$; Input current.
- $U_o$ $[V]$; Output voltage after the AC/DC converter.
- $I_o$ $[A]$; Output current after the AC/DC converter.
- $duty$ $[\%]$; Duty cycle of the transistor switching, is set at $27.5\%$

These measured values will be used to calculate the efficiency, $\eta$ $[\%]$. The values are calculated according to equation 17 which only applies in ideal situations. The fraction is used to calculate the efficiency of a system, output power divided by input power. However, the input power is subject to the duty cycle because no current is applied when the Q1 transistor is off. When Q1 is switched off, Q2 is on and the inductor and capacitor use the attained power to resonate. As both states are equally important, the optimal duty cycle would be $50\%$. Because the scale for the duty cycle is $0$ to $100\%$, the duty cycle is multiplied by 2 in the equation to get the correct ratio.

$$\eta = \frac{U_o * I_o}{(U_i * I_i) * duty * 2} \tag{17}$$

The measured values, can be found in appendix E. The results have been plotted in a 3D graph seen in figure 19. The graph shows the efficiency against the distance, $d$, and misalignment, $x$. Because the pad is circular, the misalignment can be mirrored to visualise the graph for the range of $-40$ to $40mm$. The plane plotted with the data visualises requirement E05 stating $80\%$ efficiency is required..

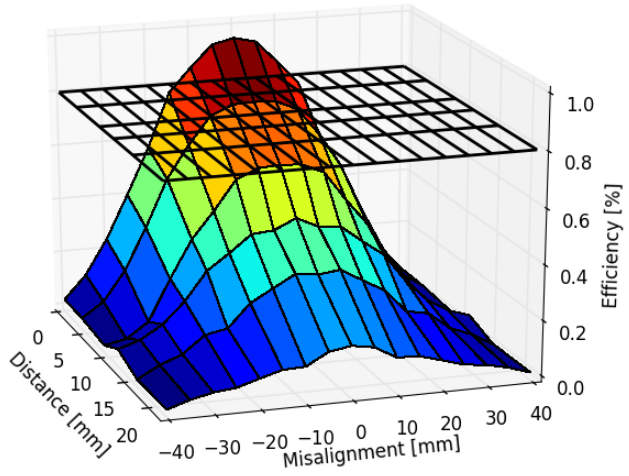



**Figure 19:** Result plotted in a 3D graph. The plane represents the requirement of reaching 80% efficiency.

### 5.1.4 Evaluation

As the graph in figure 19 shows, the power transfer is highly efficient, with a peak slightly above $95\%$. The test however, was a rather simple test which might not reflect the values perfectly. Below are some aspects which can be improved to find more accurate efficiency values.

- The used power supply which showed the input voltage and current, had analog meters on it, thus relatively large readout errors could have been made.
- The duty cycle was not set at the optimum value, which means the used equation had to be changed to account for this. For later tests, the duty cycle should be at $50\%$.
- The coils heated up during testing, for better tests, each measurement should be started with the same conditions, especially since the used cores have an optimum temperature of $90\,\mathrm{deg}$.
- The deadtime between the switching should be accounted for as this means a part of each cycle does not function as expected (a perfect square wave). This becomes more significant when operating frequencies become higher.

The plane in the graph shows there are several calculated values higher than the wanted $80\%$. The misalignment can go up to $10mm$ while still keeping a high enough efficiency. However, the efficiency goes below $80\%$ when the distance goes from $0mm$ to $4mm$. This means requirement E02 which states the power charging system should charge at a minimum distance of $5mm$ has not been met with this test. A probable way to meet this requirement is to decrease the thickness of the frontal plate of the cores ($3mm$ per coil) thus effectively increasing the distance towards $6mm$ (without frontal plates).

## 5.2 Realisation: Docking behaviour

The software was realised using Robot Operating System hydro (ROS), Python and C++. ROS is used as framework for several different software packages and fragments. C++ has the biggest community for programming with ROS and most packages are therefore C++ based. However, Python is used to create the software because development is faster.

### 5.2.1 ROS

ROS was developed to make it easy to connect several different kind of sensors and is mainly used in robot development. Software in ROS is built up of packages. These packages can contain all kinds of software, which can be launched to, for example, read out sensors. If there is need for using several packages at once, ROS is used to connect them through standardisation. A schematic of where ROS stands in a Linux operating system is shown in figure 20. The figure shows that inside ROS, several software programs of either C++ or Python can run, these programs can then connect with each other or through Linux with installed software or physical ports.

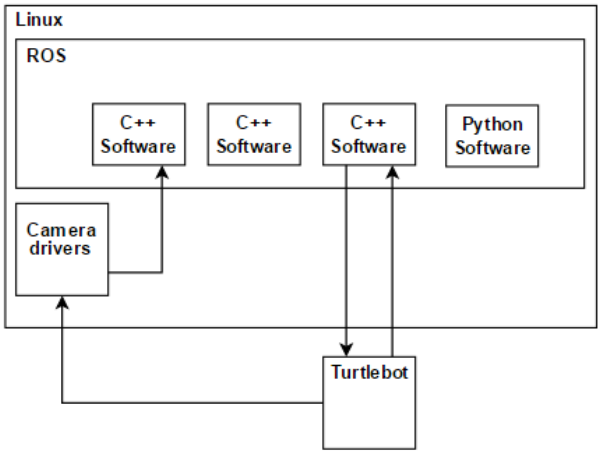



**Figure 20:** Schematic of how ROS operates inside a linux distribution.

Software is started by using a launch file which starts up several software parts sequentially. Most of these parts are nodes, nodes are parts which attain data, do something with the data and send out the data again.

For everything to work, nodes need to be connected. This is done through topics, topics are used to transfer the data from one node to the other. A topic is created by a node and is created to transfer a specified message. This message can be a datatype like an integer or a string, but can also be a self made message consisting of several datatypes. When the topic is created, nodes can publish messages on it. Different nodes can subscribe to this topic and read out the published messages. A schematic representation of ROS nodes and topics can be seen in figure 21.

So, in short, a network is created with nodes, connected by topics which can be used to send messages from node to node.

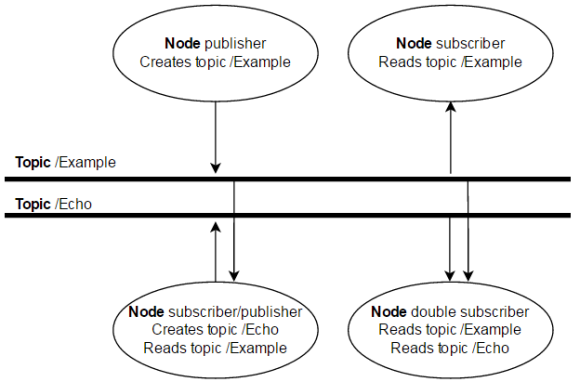



**Figure 21:** Schematic of how ROS connects nodes by creating topics.

By using available packages of the Turtlebot, all hardware of the Turtlebot can be accessed easily. The package which was used for the Turtlebot is turtlebot [5]. To use the Kinect for feature recognition, the package ar_tools was used [4], which is a version of ARToolkit for ROS. This package is used to find the relative distance and orientation of a specified marker using a 2D RGB image.

As both packages are used through ROS, they can be connected to each other. When installed, ar_tools uses a topic created by the package itself from a standard webcam interface. The Kinect however, publishes several different video streams using the package of the Turtlebot. By changing the topic to which the program subscribes to, to receive its videostream, the two packages can be connected and used together. For the videostream to be changed, the header file ar_single.h in the ar_tools package has to be adapted.

### 5.2.2 Coding

The realisation of the software was done according to the flow diagram shown in figure 22. The full code, including more specific code explanations through comments, can be found in appendix D.

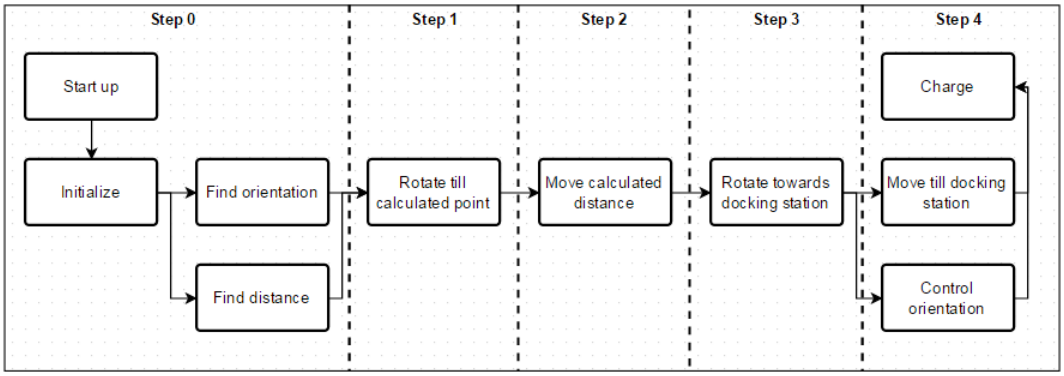



**Figure 22:** Revised flowchart showing what parts are included in each part of the code.

Figure 23a shows an Rviz visualisation of the Turtlebot with all packages launched and the feature in sight, as seen in figure 23. Before being able to move the robot, some data has to be attained as to where the robot is and where it should move.


**Figure 23:** Videostream image of the Kinect RGB channel on the Turtlebot.

First the node should be initialized, step 0, making it a part of ROS and thus being able to communicate through ROS. For being able to move the robot, a publisher has to be started. This publisher is connected to the topic "/mobile_base/commands/velocity" that takes a message setting velocities for rotations and linear movements. In case of the Turtlebot only two velocities are used, rotation around the z-axis and linear movement along the x-axis.

To find transformations between frames, a subscriber subscribed to the topic "/tf" is created. The frame $Odom$ is the point where the Turtlebot started and is thus the only reference to the area. All movements are tracked through a gyroscope for rotation and encoders on the wheels for position and are given through frame $Odom$. Before moving, the feature on the docking station should be detected. When the feature is detected, the program starts and transformations are taken which are correct until the robot moves.

The goal of the robot is to move to the front of the docking station at a specified range, from here on called $First\_point$. As the $ar\_marker$ frame also gives the orientation, this point can be created by a vector of a specified length attached to this frame. By using the transformations of the $ar\_marker$ to $base\_footprint$, the vector can be changed to point at the same position but from frame $base\_footprint$, which is the robots location. From this vector, the direction and the distance towards the frame $First\_point$ can be found. The distance can be calculated through using the Pythagoras equation. The rotation is found by a function which determines the angle of the vector and adds a set value depending on which quadrant it is in. Figure 23b shows a visualisation of the found data after step 0.

Now, the movement can be started according to the flow chart.

- Step 1: Home robot at $First\_point$

  The required rotation is compared to its current rotation causing the Turtlebot to rotate till the correct orientation has been found. As it should choose the fastest direction to rotate, a function has been written which determines whether it should go clockwise or counter clockwise. This function compares two given values which are in range of 0 - $2\pi$. Figure 23c shows how the robot has rotated and that this caused the feature to be out of the camera view.

- Step 2: Move robot to $First\_point$

  The robot is moved by the found distance to $First\_point$. Through comparing distance travelled to required distance, it is known when the distance has been reached and the robot stops. Figure 23d shows the data of the robot after moving.

- Step 3: Home robot at the Docking station

  The rotation and distance till the docking station is found in the same way as it was found for $First\_point$. By using the same technique as described in step 1, the robot can be aimed at the docking station. The feature gets in the camera view once again as can be seen in figure 23e

- Step 4: Dock the robot

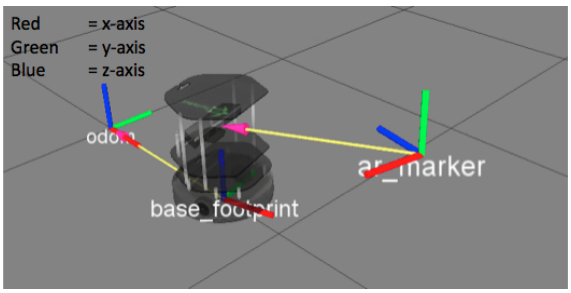
  The previous steps did not require a high accuracy because of the controller in step 4. By using a feedback controller in step 4, small misalignments are covered. Through the same methods as described in step 2, the robot is moved towards the docking station. The feedback controller compares the required direction to the current direction. By giving the output as angular velocity, the robot slightly adjusts while moving forward. When the marker is not detected, the angular velocity is set to zero to prevent the robot from changing direction drastically. Figure 23f shows the data of the robot when it is docked.

### 5.2.3 Testing docking behaviour

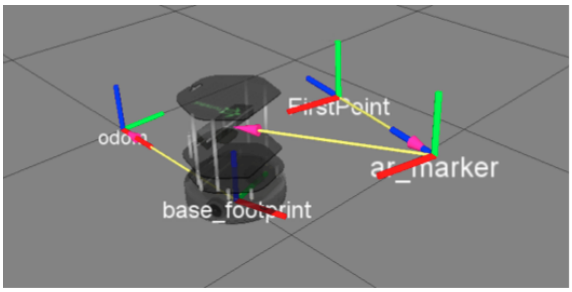
To test the software, it has been tested ten times starting at different locations. As there is no full docking station yet, the feature to detect was placed in an open area. To measure the accuracy, several measurements have been done at the end of the routine. Because the software controls the Turtlebot from its rotation point which is in the centre, an offset is needed. The Turtlebot has a diameter of $351.5mm$ so atleast half of it should be used to touch the feature.

The following variables in the program are set:

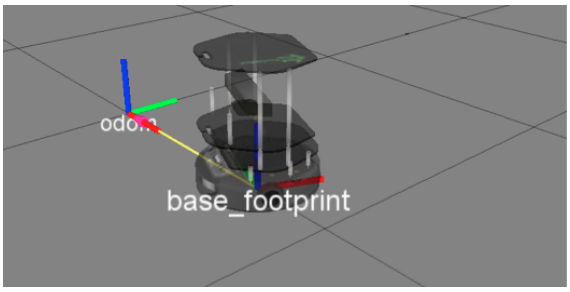
- The distance from the feature to the first point is set at $0.5m$.
- An offset of $250mm$ has been added to the feature to make the robot stop before it hits the feature. This is done to make it possible to also measure the distance errors beyond the feature. The expected distance from the feature is thus $74mm$.



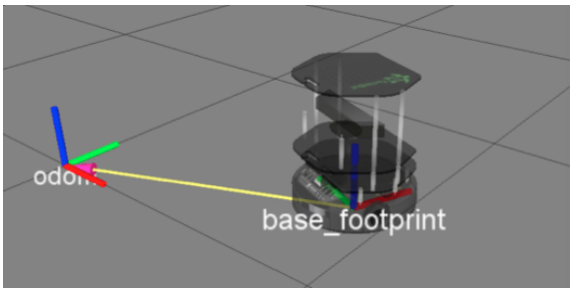

**(a)** Data after all software is launched.



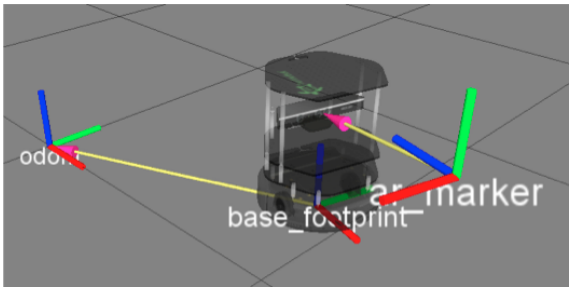

**(b)** Data after the initialising is done.



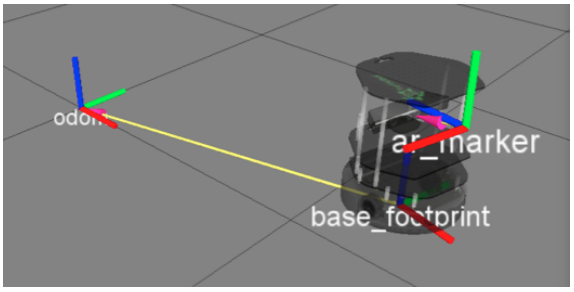

**(c)** Data after the first step is done.




**(d)** Data after the second step is done.




**(e)** Data after the third step is done.




**(f)** Data after the fourth step is done.

**Figure 23:** Visualisation of all the data attained used in the docking software.

The following measurements are done when the program has been executed

- $d$ in $mm$; Distance till the feature minus the offset of $25cm$.
- $\alpha$ in $\deg$; Angle at which the robot docked.

The measured values for ten tests can be found in table 6.

As there is a systematic error of significant size between the expected value ($74mm$)and the mean value ($59mm$), the standard deviation has been calculated using the mean. This results in $59 \pm 4.9mm$ with a systematic error of $15mm$ for the distance and $0 \pm 5.1\,\deg$ for the angle.

| Test number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $d$ $[mm]$ | 62 | 53 | 61 | 59 | 58 | 61 | 54 | 63 | 51 | 68 | 59 |
| $\alpha$ [deg] | -4 | -2 | 3 | 5 | 8 | 6 | -1 | 1 | 1 | -10 | 0.7 |

**Table 6:** Measured values of the tests done with the robot docking software.

### 5.2.4 Evaluation

The systematic error occurring in the program could be caused by several aspects:

- The camera position is not correctly configured, thus the distance relative to the camera is not correct.
- The camera calibration is not correct.
- The distance measurement in the program does not work correctly.
- Wheels slip causing the encoder to measure movement without any actual movement.
- The control loop causes the robot to make a curved path causing the distance moved correct, yet the movement straight ahead incorrect.

When the docking station detects the robot, this error does not matter. For further testing with this program, it is useful to be aware of the capabilities.

The error in the angle is caused by distance error as well. Because the robot does not move to the position in front of the docking station perfectly, some error in the angle should be expected.

# 6 Conclusions and recommendations

For autonomous robots, autonomous docking is required. Current systems do not give the desired results and a different method is therefore required.

Different complete systems consisting of robots using docking stations, showed different docking behaviours, designed specifically around each robot. These robots all use physical contact to charge and either use the docking station to optimise the alignment or accept a chance of failing.

By researching applications of charging methods, applications used in electric cars, more specifically the charging pad, showed the most similarities with the requirements. By scaling down previously designed charging pads, the pads can be implemented for robot applications. The first test of the charging pads showed promising results, with efficiencies going as high as $95\%$.

However, the tests should be done at $50\%$ duty cycle to make sure the efficiency was measured correctly. The efficiency, quality and overall performance can be further increased if the charging pads are made for a more specific applications. Optimising consists of changing the core layout, the core material and the surrounding construction material.

As all autonomous robots have some kind of vision system, vision was used for the docking behaviour. Through detecting a feature, moving the robot in front of the docking station and using a feedback control loop to guide the robot towards the docking station, software which complements the charging pad has been created.

In a later stage the software should be made more customisable to increase the flexibility and rewritten in C++ to increase performance and accessibility.

When combining the charging pads and the robot software and looking at the individual test results, most requirements can be met. The requirements which cannot be met, can be reached through further optimisation of the system.

The report shows that wireless charging is definitely feasible for higher power applications. Further optimisation of the system will show whether it can replace the conventional charging method.

# References

[1] M. Budhia, G.A. Covic, and J.T. Boys. Design and optimization of circular magnetic structures for lumped inductive power transfer systems. *Power Electronics, IEEE Transactions on*, 26(11):3096–3108, Nov 2011.

[2] G.A. Covic and J.T. Boys. Inductive power transfer. *Proceedings of the IEEE*, 101(6):1276–1289, June 2013.

[3] J.C.F de Beer. *Methodisch ontwerpen*. BIM Media BV, 2013.

[4] G. Dumonteil. Artoolkit for ros. *wiki.ros.org/ar_tools*, visited: 03-06-2015.

[5] T. Foote, M. Ferguson, and M. Wise. Turtlebot package. *wiki.ros.org/turtlebot*, visited: 03-06-2015.

[6] Magnetics Inc. Specifications for p material. *http://www.mag-inc.com/products/ferrite-cores/p-material*, visited: 03-06-2015.

[7] iRobot. irobot roomba stofzuigrobot. *www.irobot.nl/Huisrobots/Stofzuigen*, visited: 03-06-2015.

[8] C. Karijodinomo. Inductieve draadloze oplader. -, 2014.

[9] A. Kent. Hands-off: Roomba robotic vacuum review. *www.homeserverland.com/2010/08/hands-off-roomba-robotic-vacuum-review/*, visited: 03-06-2015.

[10] Xun Liu and S.Y.R. Hui. Equivalent circuit modeling of a multilayer planar winding array structure for use in a universal contactless battery charging platform. *Power Electronics, IEEE Transactions on*, 22(1):21–29, Jan 2007.

[11] Power matters alliance. Low power standard. *http://www.powermatters.org/*, visited: 03-06-2015.

[12] D. Meeker. Finite element method magnetics. *http://www.femm.info/wiki/HomePage*, visited: 03-06-2015.

[13] F. Nakao, Y. Matsuo, M. Kitaoka, and H. Sakamoto. Ferrite core couplers for inductive chargers. In *Power Conversion Conference, 2002. PCC-Osaka 2002. Proceedings of the*, volume 2, pages 850–854 vol.2, 2002.

[14] Wireless power consortium. Low power standard. *http://www.wirelesspowerconsortium.com/*, visited: 03-06-2015.

[15] V. Prasanth. Wireless power transfer for e-mobility. -, July 2012.

[16] Yujin Robot. Kobuki robot. *http://kobuki.yujinrobot.com/home-en/about/l*, visited: 03-06-2015.

[17] Double robotics. Double robotics. *www.doublerobotics.com*, visited: 03-06-2015.

[18] L. Schulze, S. Behling, and S. Buhrs. Automated guided vehicle systems: a driver for increased business performance. In *Proceedings of International Multi Conference of Engineers and Computer Scientists 2008 (IMECS 2008*, pages 1275–1280, 2008.

[19] P. Sergeant and A. Van den Bossche. Inductive coupler for contactless power transmission. *Electric Power Applications, IET*, 2(1):1–7, Jan 2008.

[20] R. Severns, E. Yeow, G. Woody, J. Hall, and J. Hayes. An ultra-compact transformer for a 100 w to 120 kw inductive coupler for electric vehicle battery charging. In *Applied Power Electronics Conference and Exposition, 1996. APEC '96. Conference Proceedings 1996., Eleventh Annual*, volume 1, pages 32–38 vol.1, Mar 1996.

[21] M. Sole, A. Sanni, A. Vilches, C. Toumazou, and T.G. Constandinou. A bio-implantable platform for inductive data and power transfer with integrated battery charging. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2605–2608, May 2011.

[22] E. Waffenschmidt. Wireless power for mobile devices. In *Telecommunications Energy Conference (INTELEC), 2011 IEEE 33rd International*, pages 1–9, Oct 2011.

[23] Alliance 4 wireless power. Low power standard. *http://www.rezence.com/*, visited: 03-06-2015.

# Appendix

## A Detailed requirement list

This document describes the requirements of the assignment ”Improving autonomous docking of assistive robots”. The requirements are split up in four categories:

- Electrical(E); All requirements based on the electrical system.
- Mechanical(M); All requirements which are based on a physical system but do not belong with the electrical system.
- Software(S); All requirements for the programming of the robot and docking station.
- Global(G); The requirements which are based on the environment and other non-changeable requirements.

The ID number of each requirement contains a reference to the category the requirement belongs, followed by an unique number. The ID numbers will not be changed or reused. Besides the ID, a value is given for priority and criticality, based on the amount of requirements in this document, it was chosen to use only the values 1, 5 and 10. The priority is used to determine the importance of meeting the requirement. The criticality represents the effect caused by not meeting the requirement. A low value corresponds to high priority or criticality.
Descriptions are not always added, only when extra information is needed to understand the requirement thoroughly.

# Electrical

| ID: | E01 | Mutation | date | 19-02-2015 |
|---|---|---|---|---|
| Requirement: | | | | |
| The docking station has to be able to charge wireless. | | | | |
| Description: | | | | |
| A wireless docking station is needed to remove the chance of damage to physical contacts. | | | | |

| ID: | E19 | Mutation | date | 16-03-2015 |
|---|---|---|---|---|
| Requirement: | | | | |
| The docking station has to be charged electrically. | | | | |
| Description: | | | | |
| How the energy is transferred can differ, but the input is electrical and the output must be electrical. | | | | |

| ID: | E20 | Priority: | 1 | Criticality: | 5 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The docking station has to be able to charge wireless. | | | | | |
| Description: | | | | | |
| A wireless docking station is needed to remove the chance of damage to physical contacts. | | | | | |

| ID: | E21 | Priority: | 1 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: The docking station should use wall power at, possibly higher frequencies. | | | | | |
| Description: | | | | | |

| ID: | E02 | Priority: | 5 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: Charging has to be done with a distance of atleast 5 mm. | | | | | |
| Description: Even though it is specified at 5 mm, higher ranges are preferred. | | | | | |

| ID: | E03 | Priority: | 1 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: The charging mechanism charges the robot with atleast 30V and 10A. | | | | | |
| Description: As there are some examples for robots, these are the specifications required to properly charge these robots. | | | | | |

| ID: | E04 | Priority: | 5 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: The charging time must not exceed 2 hours. | | | | | |
| Description: | | | | | |

| ID: | E05 | Priority: | 5 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: The power transfer efficiency must be atleast 80%. | | | | | |
| Description: | | | | | |

| ID: | E06 | Priority: | 10 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: Insulation is needed to prevent disturbances. | | | | | |
| Description: If the electrical system is not insulated properly, other devices, like the robot, can be disturbed. | | | | | |

| ID: | E07 | Priority: | 10 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: The dock should be activated when the robot is in charging range of the dock. | | | | | |
| Description: | | | | | |

## Mechanical

| ID: | M08 | Priority: | 5 | Criticality: | 5 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The system must be designed modular and be able to cope with different energy requirements. | | | | | |
| Description: | | | | | |
| As the docking station has to be used to charge several different robots, high flexibility is preferred. | | | | | |

| ID: | M15 | Priority: | 5 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| To the robot, a maximum of 0.5 kg can be added. | | | | | |
| Description: | | | | | |

| ID: | M16 | Priority: | 10 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The docking station may weigh up to 3 kg. | | | | | |
| Description: | | | | | |

| ID: | M17 | Priority: | 10 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The maximum size of the docking station is 500mmx500mmx100m (w x h x d). | | | | | |
| Description: | | | | | |

| ID: | M18 | Priority: | 1 | Criticality: | 5 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The docking station should not contain any actuators. | | | | | |
| Description: | | | | | |
| Extra actuators cause more components which could potentially break down. | | | | | |

## Software

| ID: | S09 | Priority: | 10 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: | | | | | |
| The docking behaviour of the robot should use its navigation system and equipped sensors. | | | | | |
| Description: | | | | | |

| ID: | S13 | Priority: | 10 | Criticality: | 10 |
|---|---|---|---|---|---|
| Requirement: The docking station should be able to indicate whether or not the robot is in proximity. | | | | | |
| Description: | | | | | |

| ID: | S14 | Priority: | 5 | Criticality: | 5 |
|---|---|---|---|---|---|
| Requirement: The docking station should detect if the input and output voltages are correct. | | | | | |
| Description: | | | | | |

## Global

| ID: | G10 | Priority: | 1 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: The dock will be connected to the main power supply available in the house. | | | | | |
| Description: The robots are designed for indoor use, thus the station should operate with the available supply. | | | | | |

| ID: | G11 | Priority: | 1 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: The docking station will be in a fixed position. | | | | | |
| Description: | | | | | |

| ID: | G12 | Priority: | 1 | Criticality: | 1 |
|---|---|---|---|---|---|
| Requirement: The docking station has to be safe. | | | | | |
| Description: This basically means, no loose wires, insulated, no open contacts etc. | | | | | |

## B Electrical topology comparison

Table to compare the different topologies used in inductive power transfer IPT.

| Characteristic of the topology | SS Topology | SP Topology | PS Topology | PP Topology |
|---|---|---|---|---|
| Dependence of the primary compensation capacitance on load | None | None | Dependent | Dependent |
| Circuit equivalent impedance at resonance | Minimum | Minimum | Maximum | Maximum |
| Type of ac source to be applied so as to transfer maximum power | Voltage source | Voltage source | Voltage source at high voltage/ Current Source | Voltage source at high voltage/ Current Source |
| Power transferred at constant source voltage (SS, SP)/ current (PS,PP) | Lower | Higher | Lower | Higher |
| Peak efficiency | Higher | Lower | Higher | Lower |
| Tolerance of efficiency to variable frequency | Lower | Higher | Lower | Higher |
| Tolerance of power factor to variable frequency | Higher | Lower | Higher | Lower |

# C Datasheet I-core P material

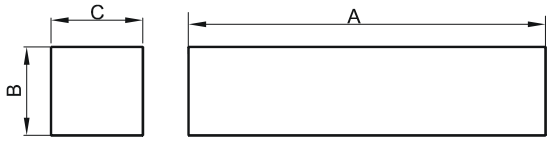
MAGNETICS ®
A Division of Spang & Company

Specification for:

**0P42516IC**

110 Delta Drive
Pittsburgh, PA 15238
Phone: 412/696-1333
Fax: 412/696-0333
Email:magnetics@spang.com

## DIMENSIONS




| (mm) | Nominal: | Tol. min.: | Tol. max.: |
|---|---|---|---|
| **A** | 25.4 | -0.51 | + 0.64 |
| **B** | 6.35 | -0.13 | + 0.13 |
| **C** | 6.35 | -0.13 | + 0.13 |
| **Eff. Parameters** (with 0_42515UC) | | | |
| **Ae mm $^2$** | **Amin mm $^2$** | **le mm** | **Ve mm $^3$** |
| 40.3 | 40.3 | 64.3 | 2590 |

## INDUCTANCE

| AL value (nH) | Test conditions (with 0P42516UC) |
|---|---|
| Nom: 2907<br>Min.: 2180 | 10 kHz, < 0.5 mT, 25 ℃ |

## MARKING

No marking

## CORE LOSSES

| $P_l$ max | Test conditions |
|---|---|
| 112 mW/cm³ (0.29 W/set) | 100 kHz, 100 mT, 100 ℃ |

## NOTE

| Spec. modifications | Previous | Revised |
|---|---|---|
| 2005-06-22 | A=25.91 Max.<br>Losses: General P material | A=26.04 Max.<br>Losses: Detail as indicated |

# D Python code

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 21 13:29:41 2015

@author: bart
"""

import numpy as np
import rospy
import time
import math
import tf
import geometry_msgs.msg
import nav_msgs.msg
import std_msgs.msg

def broadCastFrame(trans,rot,name,parent):
    """ Broadcasts a frame to the /tf topic.
    Arguments
    trans       -   x-y-z translation
    rot         -   quaternion
    name        -   string with name of the frame
    parent      -   string with name of the frame it should be connected to

    Returns:

    """
    br = tf.TransformBroadcaster()
    time.sleep(0.5)
    br.sendTransform(trans, rot, rospy.Time.now(), name, parent)
    print "Broadcasted "+name

def rotationGoal(vector):
    """ Calculates the angle of a vector.
    Arguments
    vector      -   x-y vector

    Returns:
    rot         -   rotation in radians

    """
    if vector[0] >= 0 and vector[1] >= 0:
        rot = math.atan(vector[1]/vector[0])
    elif vector[0] >= 0 and vector[1] < 0:
        rot = math.pi*2-abs(math.atan(vector[1]/vector[0]))
    elif vector[0] < 0 and vector[1] >= 0:
        rot = math.pi*0.5+abs(math.atan(vector[0]/vector[1]))
    elif vector[0] < 0 and vector[1] < 0:
        rot = math.pi*1.5-abs(math.atan(vector[0]/vector[1]))

    return rot

```

```
53
54 def callback(data):
55     """ Callback function for a publisher
56     Arguments
57     data        -   the data of the publisher
58
59     Returns:
60
61     """
62     # Global values
63     global First
64     global Second
65     global Third
66     global Fourth
67
68
69     # complete first step   : Rotation to first point
70     if First == True:
71         callback1(data)
72     # complete second step  : Linear movement to first point
73     elif Second == True:
74         callback2(data)
75     # complete third step   : Rotation to ar_marker
76     elif Third == True:
77         callback3(data)
78     # complete fourth step  : Linear movement to ar_marker with feedbackloop
79     elif Fourth == True:
80         callback4(data)
81
82 def callback1(data):
83     """ Finding the amount to turn to get to the goal
84     Arguments
85     data        -   the data of the publisher
86
87     Returns:
88
89     """
90     # Global values
91     global Velocity
92     global pub
93     global zGoal
94     global First
95     global Second
96     global InitF
97
98     # Initialisation of step 1.
99     if InitF == True:
100        raw_input('stap_1')
101        InitF = False
102
103        # Turn the quaternion into rotation around z-axis
104        zGoal = math.asin(zGoal[2])*2
105        zCurrent = math.asin(data.pose.pose.orientation.z)*2
106
107        # The new goal is the current position and the old position added
```

```
108         zGoal = zGoal+zCurrent
109 
110         # Remove full rotations from the goal
111         if zGoal < -math.pi:
112             zGoal = zGoal+2*math.pi
113         if zGoal > math.pi:
114             zGoal = zGoal-2*math.pi
115 
116         # Check if clockwise or counterclockwise
117         if cwOrCcw((zGoal+math.pi),(zCurrent+math.pi)) == True:
118             Velocity.angular.z = -0.5
119         else:
120             Velocity.angular.z = +0.5
121         # Make first false to prevent repeated execution
122 
123     # Main loop of step 1
124     # Compare the goal to the current values with a margin (in radians)
125     if (zGoal) < (math.asin(data.pose.pose.orientation.z)*2+0.02) and (zGoal) > (
    math.asin(data.pose.pose.orientation.z)*2-0.02) :
126         # For faster reaction, publish 0 velocity
127         Velocity.angular.z = 0
128         pub.publish(Velocity)
129         # Make second true to continue to the next part.
130         First = False
131         Second = True
132 
133     else:
134         pub.publish(Velocity)
135 
136 def callback2(data):
137     """ Finding the amount to move to get to the goal
138     Arguments
139     data        -   the data of the publisher
140 
141     Returns:
142 
143     """
144     # Global values
145     global Velocity
146     global pub
147     global distance
148     global Second
149     global Third
150     global InitS
151     global startx
152     global starty
153 
154     # Initialisation of step 2.
155     if InitS == True:
156         raw_input('stap_2')
157         InitS = False
158         Velocity.linear.x = 0.2
159         startx = data.pose.pose.position.x
160         starty = data.pose.pose.position.y
161 
```

```
162     # Main loop of step 2.
163     # Calculate the moved distance compared to the start
164     test = math.sqrt((data.pose.pose.position.x-startx)**2+(data.pose.pose.
        position.y-starty)**2+0.001)
165     # When the required distance is not the same as the moved distance, keep
        moving
166     if distance > test:
167         pub.publish(Velocity)
168     else:
169         Velocity.linear.x = 0
170         pub.publish(Velocity)
171         Second = False
172         Third = True
173
174 def callback3(data):
175     """ Finding the amount to turn to get to the final goal
176     Arguments
177     data        -   the data of the publisher
178
179     Returns:
180
181     """
182     # Global values
183     global Velocity
184     global pub
185     global zGoal
186     global listener
187     global nGoal
188     global Third
189     global Fourth
190     global InitT
191
192     # Initialisation of step 3.
193     if InitT == True:
194         InitT = False
195         raw_input("stap_3")
196         tTrans,tRot = listener.lookupTransform('/odom','/base_footprint', rospy.
        Time())
197         homOBn = listener.fromTranslationRotation(tTrans, tRot)
198         vBasen = np.dot(np.linalg.inv(homOBn),vectorA)
199         nGoal = rotationGoal(vBasen)
200
201         zCurrent = math.asin(data.pose.pose.orientation.z)*2
202
203         # The new goal is the current position and the old position added
204         nGoal = nGoal+zCurrent
205
206         # Remove full rotations from the goal
207         if nGoal < -math.pi:
208             nGoal = nGoal+2*math.pi
209         if nGoal > math.pi:
210             nGoal = nGoal-2*math.pi
211
212         # Check if clockwise or counterclockwise
213         if cwOrCcw((nGoal+math.pi),(zCurrent+math.pi)) == True:
```

```
214             Velocity.angular.z = -0.5
215         else:
216             Velocity.angular.z = +0.5
217 
218     # Main loop of step 3.
219     if (nGoal) < (math.asin(data.pose.pose.orientation.z)*2+0.02) and (nGoal) > (
    math.asin(data.pose.pose.orientation.z)*2-0.02) :
220         # For faster reaction, publish 0 velocity
221         Velocity.angular.z = 0
222         pub.publish(Velocity)
223 
224         # Make second true to continue to the next part.
225         Third = False
226         Fourth = True
227     else:
228         pub.publish(Velocity)
229 
230 def callback4(data):
231     """ Finding the amount to move and using a small control loop to go straigth.
    Arguments
232     data        -   the data of the publisher
233 
234     Returns:
235 
236     """
237     # Global values
238     global Velocity
239     global pub
240     global zGoal
241     global distance
242     global listener
243     global InitV
244     global startx
245     global starty
246     global tNow
247     global eNow
248     global Plant
249 
250     # Try to detect the marker, if marker is lost, move forward.
251     while(True):
252         try:
253             # Get the transformations from odom to ar_marker
254             (sTrans,sRot) = listener.lookupTransform('/odom','/ar_marker', rospy.
    Time())
255             break
256         except:
257             Velocity.angular.z = 0
258             pub.publish(Velocity)
259     # Get the new vector towards the marker from base_footprint.
260     vectorA = np.array([sTrans[0],sTrans[1],0,1])
261     tTrans,tRot = listener.lookupTransform('/odom','/base_footprint', rospy.Time
    ())
262     homOBn = listener.fromTranslationRotation(tTrans, tRot)
263     vBasen = np.dot(np.linalg.inv(homOBn),vectorA)
264     nGoal = rotationGoal(vBasen)
265 
```

```
266
267     # Initialisation of step 4
268     if InitV == True:
269         raw_input("stap_4")
270         tNow = time.time()
271         eNow = 0
272         distance = math.sqrt(vBasen[0]**2+vBasen[1]**2+0.01)
273         Velocity.linear.x = 0.05
274         startx = data.pose.pose.position.x
275         starty = data.pose.pose.position.y
276         Plant = 0.01 # Determines the plant value
277         InitV = False
278
279     zCurrent = math.asin(data.pose.pose.orientation.z)*2
280     # The new goal is the current position and the old position added
281     nGoal = nGoal+zCurrent
282     # Remove full rotations from the goal
283     if nGoal < -math.pi:
284         nGoal = nGoal+2*math.pi
285     if nGoal > math.pi:
286         nGoal = nGoal-2*math.pi
287
288     # setpoint is the goal + pi to always have a positive number
289     setpoint = nGoal + math.pi
290     # current is the current + pi to always have a positive number
291     current = zCurrent + math.pi
292     # Make the new time old time and get the current time
293     tOld = tNow
294     tNow = time.time()
295     # Make the new error old error and calculate the new error
296     eOld = eNow
297     eNow = setpoint - current
298
299     # Calculate propertional, intergral and differential values.
300     Prop =  eNow
301     Inte = (eNow - eOld)*(tNow-tOld)
302     Diff = (eNow - eOld)/(tNow-tOld)
303
304     # Set the influence these values have. Kd is set to zero because of the
305     # very small time changes.
306     Kp = 120
307     Ki = 50
308     Kd = 0
309
310     # Add all values together and multiply with the Plant value for the correct
        output
311     output = (Prop*Kp - Inte*Ki + Diff*Kd) * Plant
312     Velocity.angular.z = output
313
314     # Stop 10 cm before the ar_marker.
315     test = math.sqrt((data.pose.pose.position.x-startx)**2+(data.pose.pose.
        position.y-starty)**2+0.001)
316     if test+0.1 < distance:
317         pub.publish(Velocity)
318
```

```
319     else:
320         Velocity.linear.x = 0
321         Velocity.angular.z = 0
322         rospy.signal_shutdown(" Done ")
323 
324 def cwOrCcw(a,b):
325     # Determine whether clockwise or counterclockwise rotation is best.
326     if a < b:
327         if abs(a-b) < abs((a+2*math.pi-b)):
328             cw = True
329 #             print '1'
330         else:
331             cw = False
332 #             print '2'
333     if a > b:
334         if abs(a-b) < abs(a-(b+2*math.pi)):
335             cw = False
336 #             print '3'
337         else:
338             cw = True
339 #             print '4'
340     return cw
341 
342 
343 global Velocity
344 global zGoal
345 global pub
346 global distance
347 global listener
348 
349 global First
350 global Second
351 global Third
352 global Fourth
353 First = True
354 Second = Third = Fourth = False
355 
356 global InitF
357 global InitS
358 global InitT
359 global InitV
360 global vectorA
361 InitF = InitS = InitT = InitV = True
362 
363 Velocity = geometry_msgs.msg.Twist()
364 vector = np.array([0,0,0.5,1]).reshape(4,1)
365 
366 # Start the node
367 rospy.init_node("Testturn", anonymous=True)
368 
369 # Define the velocity publisher
370 pub = rospy.Publisher('mobile_base/commands/velocity', geometry_msgs.msg.Twist)
371 
372 # Transform listener
373 listener = tf.TransformListener()
```

```
374
375 # Check if the base transforms are available
376 try: listener.waitForTransform('/odom', '/base_footprint', rospy.Time(), rospy.
    Duration(5))
377 except: print "ROS exception was raised"
378
379 # Get the transformations from odom to base_footprint
380 (bTrans,bRot) = listener.lookupTransform('/odom','/base_footprint', rospy.Time())
381
382 # Try until an ar_marker is found
383 while(True):
384     try:
385         # Get the transformations from odom to ar_marker
386         (aTrans,aRot) = listener.lookupTransform('/odom','/ar_marker', rospy.Time
    ())
387         print "marker found"
388         break
389     except: continue
390
391 # Create homogeneous matrices for convenience
392 vectorA = np.array([aTrans[0],aTrans[1],0,1])
393 homOA = listener.fromTranslationRotation(aTrans, aRot)
394 homOB = listener.fromTranslationRotation(bTrans, bRot)
395
396 # Change vector from the ar_marker frame to odom frame
397 vOdom = np.dot(homOA, vector)
398
399 # Change vector from odom frame to base frame
400 vBase = np.dot(np.linalg.inv(homOB),vOdom)
401
402
403 # Project it onto ground plane by setting the z-axis to zero
404 vBase[2] = 0
405
406 broadCastFrame(vector,(0,0,0,1),'FirstPoint','ar_marker') # translates 2m from
    ar_marker
407
408 # Find the rotation needed seen from the base_footprint
409 zGoal = rotationGoal(vBase)
410
411 # Translate the rotation into a quaternion
412 zGoal = tf.transformations.quaternion_about_axis(zGoal,(0,0,1))
413
414 # Calculate length of the vector
415 distance = math.sqrt(vBase[0]**2+vBase[1]**2)
416
417 # Subscribe to /odom and jump to function callback
418
419 rospy.Subscriber("/odom", nav_msgs.msg.Odometry, callback)
420
421 rospy.spin()
422 while(rospy.is_shutdown()!=True): pass
423
424 print "Node is shut down"
```

## E Test data of coil efficiency with varying displacements

This appendix contains four tables that show measured data. The first table shows the input current [$I_i$] of the transmitting coil at $24V$ [$U_i$. The test data has been obtained with a duty cycle of $27.5\%$ and the final efficiency has been calculated using equation 18, which applies in ideal situations:

$$\eta = \frac{U_o * I_o}{(U_i * I_i) * duty * 2} \tag{18}$$

| Displacement [$mm$] Misalignment [$mm$] | 0 | 4 | 9 | 12 | 16 | 22 |
|---|---|---|---|---|---|---|
| 0 | 2.5 | 2.7 | 2.7 | 2.5 | 2.4 | 2.2 |
| 5 | 2.6 | 2.8 | 2.7 | 2.5 | 2.4 | 2.1 |
| 10 | 2.6 | 2.7 | 2.6 | 2.5 | 2.3 | 2.0 |
| 15 | 2.7 | 2.7 | 2.5 | 2.3 | 2.2 | 2.0 |
| 20 | 2.6 | 2.6 | 2.4 | 2.2 | 2.5 | 2.0 |
| 25 | 2.3 | 2.3 | 2.1 | 2.0 | 2.0 | 1.9 |
| 30 | 2.1 | 2.1 | 1.9 | 1.9 | 2.0 | 1.9 |
| 35 | 2.0 | 1.9 | 1.9 | 1.9 | 1.9 | 1.9 |
| 40 | 1.9 | 1.8 | 1.8 | 1.9 | 1.8 | 1.8 |

**Table 7:** Input current [A] measured at supply.

The following table shows the output current [$I_o$] measured after the AC/DC converter.

| Displacement [$mm$] Misalignment [$mm$] | 0 | 4 | 9 | 12 | 16 | 22 |
|---|---|---|---|---|---|---|
| 0 | 2.50 | 2.41 | 2.08 | 1.70 | 1.40 | 0.95 |
| 5 | 2.49 | 2.38 | 2.08 | 1.68 | 1.40 | 0.91 |
| 10 | 2.43 | 2.29 | 1.98 | 1.60 | 1.32 | 0.80 |
| 15 | 2.30 | 2.09 | 1.76 | 1.47 | 1.15 | 0.74 |
| 20 | 1.98 | 1.84 | 1.53 | 1.48 | 1.01 | 0.58 |
| 25 | 1.56 | 1.41 | 1.13 | 1.50 | 0.85 | 0.52 |
| 30 | 1.14 | 1.00 | 0.60 | 0.78 | 0.70 | 0.45 |
| 35 | 0.76 | 0.60 | 0.30 | 0.58 | 0.48 | 0.41 |
| 40 | 0.33 | 0.30 | 0.10 | 0.45 | 0.21 | 0.15 |

**Table 8:** Output current [A] after the AC/DC converter.

The following table shows the output voltage measured [$U_o$] after the AC/DC converter over a resistor.

| Displacement [$mm$] Misalignment [$mm$] | 0 | 4 | 9 | 12 | 16 | 22 |
|---|---|---|---|---|---|---|
| 0 | 12.6 | 11.0 | 10.5 | 9.0 | 7.3 | 4.9 |
| 5 | 12.4 | 11.5 | 10.4 | 8.7 | 7.2 | 4.7 |
| 10 | 12.1 | 11.4 | 9.9 | 8.3 | 6.7 | 3.8 |
| 15 | 11.5 | 10.4 | 8.7 | 7.7 | 6.1 | 3.8 |
| 20 | 9.9 | 9.2 | 7.6 | 7.0 | 5.2 | 3.8 |
| 25 | 7.8 | 7.3 | 5.6 | 7.0 | 4.4 | 3.0 |
| 30 | 5.8 | 5.1 | 3.2 | 4.0 | 3.0 | 2.8 |
| 35 | 3.8 | 3.1 | 1.7 | 3.0 | 2.5 | 2.0 |
| 40 | 1.7 | 1.6 | 1.2 | 2.4 | 1.4 | 1.5 |

**Table 9:** Output voltage [V] after the AC/DC converter.

The following table shows the efficiency [$\eta$] calculated from the three previously given tables according to equation 18.

| Displacement [$mm$] Misalignment [$mm$] | 0 | 4 | 9 | 12 | 16 | 22 |
|---|---|---|---|---|---|---|
| 0 | 95 | 77 | 61 | 46 | 32 | 17 |
| 5 | 93 | 76 | 60 | 43 | 33 | 15 |
| 10 | 85 | 73 | 57 | 42 | 29 | 11 |
| 15 | 77 | 63 | 46 | 36 | 24 | 11 |
| 20 | 57 | 51 | 37 | 30 | 16 | 8 |
| 25 | 40 | 34 | 23 | 20 | 14 | 6 |
| 30 | 24 | 18 | 7 | 12 | 8 | 5 |
| 35 | 11 | 7. | 2 | 7 | 5 | 3 |
| 40 | 2 | 1. | 0 | 4 | 0 | 1 |

**Table 10:** Efficiency [%] of the power transfer.