How to track an object using ultrasound

Hur man följer ett objekt med hjälp av ultraljud

RAPHAEL HASENSON
CHRISTOFFER LARSSON OLSSON
How to track an object using ultrasound

CHRISTOFFER LARSSON OLSSON, RAPHAEL HASENSON

Bachelor Thesis at ITM
Supervisor: Masoumeh Parseh
Examiner: Nihad Subasic

TRITA MMK 2017:07 MDAB 625
Abstract

Ultrasound has previously been used mainly for imaging applications such as Sonography or range finding.

The objective of this thesis is to find out the suitability of using ultrasound as means of tracking an object. This was implemented through the tracking object fitted with an ultrasonic emitter whereas the robot was fitted with an ultrasonic receiver. The result of the robot’s performance shows some of the restrictions, such as the user having to be within a certain range, and possibilities, such as allowing foreign obstacles in the path between user and robot, when implementing ultrasound for tracking purposes. This thesis shows that applying ultrasound technology in a tracking environment is a worthwhile endeavour and describes one possible setup that works.
Referat

Hur man följer ett objekt med hjälp av ultraljud

Ultraljud har tidigare använts främst för avbildning, bland annat sonografi, eller för distansmätning.

Acknowledgements

We would like to thank our supervisor Nihad Subasic his continued support throughout the project. We would also like to thank Hans Johansson and the assistants for their guidance and criticism, our fellow students for engaging in constructive discussions and Staffan Qvarnström for helping us order the right components.

Raphael Hasenson, Christoffer Larsson Olsson
Stockholm, May, 2017
# Contents

## Acknowledgements

 iii

### Abbreviations

 viii

## 1 Introduction

 1

 1.1 Background ................................. 1

 1.2 Purpose ..................................... 1

 1.3 Scope ...................................... 2

 1.4 Method .................................... 2

 1.4.1 Tracking ................................. 2

## 2 Theory

 3

 2.1 Pulse Width Modulation (PWM) .................. 3

 2.2 Ultrasound (US) ............................... 3

 2.2.1 Diffraction ................................ 3

 2.3 DC motors ................................... 4

 2.4 H-Bridge ..................................... 4

 2.5 Microcontroller ............................... 4

## 3 Demonstrator

 7

 3.1 Problem Formulation .......................... 8

 3.2 Software ..................................... 8

 3.2.1 Robot ..................................... 8

 3.2.2 Transmitter ................................. 9

 3.3 Electronics .................................. 10

 3.3.1 Transmitter circuit ......................... 10

 3.3.2 Receiver circuit ............................ 10

 3.3.3 H-bridge .................................. 11

 3.4 Hardware .................................... 11

 3.4.1 Arduino UNO ................................ 11

 3.4.2 DC motor ................................... 11

 3.4.3 Receiver Circuit ............................ 12

 3.4.4 Transmitter Circuit ......................... 12

 3.4.5 US transducer .............................. 12
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Visual representation of the entire system</td>
</tr>
<tr>
<td>3.2</td>
<td>Software Flowchart, created in the online software draw.io</td>
</tr>
<tr>
<td>3.3</td>
<td>Visual representation of the assembled transmitter</td>
</tr>
<tr>
<td>4.1</td>
<td>Allowed Area, created in the software Matlab</td>
</tr>
<tr>
<td>4.2</td>
<td>Obstacle interference on signal, where the marked area for cylinder and cuboid represent signal contact, created in the software Matlab</td>
</tr>
<tr>
<td>B.1</td>
<td>Transmitter wiring, made with the fritzing software</td>
</tr>
<tr>
<td>B.2</td>
<td>Transmitter wiring, made with the fritzing software</td>
</tr>
<tr>
<td>C.1</td>
<td>Raw data from the Obstacle allowance test</td>
</tr>
<tr>
<td>C.2</td>
<td>Specifications of the dimensions of objects used when testing obstruction allowance</td>
</tr>
<tr>
<td>C.3</td>
<td>Raw data from the test to determine the allowed area</td>
</tr>
<tr>
<td>D.1</td>
<td>Transmitter circuit PCB, made in the Eagle software</td>
</tr>
<tr>
<td>D.2</td>
<td>Receiver circuit PCB, made in the Eagle software</td>
</tr>
<tr>
<td>E.1</td>
<td>Holder used for attaching H-bridge to robot as well as transmitter circuit to Sleeve Base, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.2</td>
<td>US Holder, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.3</td>
<td>Sender Screen, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.4</td>
<td>Receiver Screen, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.5</td>
<td>Handle, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.6</td>
<td>Bottom Hinge, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.7</td>
<td>Top Hinge, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.8</td>
<td>Receiver Base Holder, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.9</td>
<td>Receiver Holder Offset, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.10</td>
<td>Sender Lid, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.11</td>
<td>Sleeve Attachment, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.12</td>
<td>Sleeves Base, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.13</td>
<td>Sleeve Base Offset, created in the SolidEdge, Academicversion software</td>
</tr>
<tr>
<td>E.14</td>
<td>Sleeve Bottom, created in the SolidEdge, Academicversion software</td>
</tr>
</tbody>
</table>
E.15 Sleeve Ledge, created in the *SolidEdge, Academicversion* software . . . 64
E.16 Sleeve Lid, created in the *SolidEdge, Academicversion* software . . . . 65
E.17 Sleeve Side, created in the *SolidEdge, Academicversion* software . . . . 66
E.18 Sleeve Top, created in the *SolidEdge, Academicversion* software . . . . 67
E.19 Receiver Holder, created in the *SolidEdge, Academicversion* software . 68
List of Tables

4.1 Signal strength dependency on configuration ........................................ 16
C.1 Raw data from the US testing as described above ................................ 40

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>Ultrasound</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse-Width Modulation</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of Vision</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>A/D</td>
<td>Analog to Digital</td>
</tr>
<tr>
<td>LED</td>
<td>Light-Emitting Diode</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Ultrasound (US) is commonly used for its ranging capabilities. When applying US technology in such a setting, one is mainly using the well-researched area of sound propagation in various media in order to gain information regarding to various distances. This data may then be interpreted differently according to the desired output, such as imaging when performing a sonogram or tracking changes in sea-levels over time.

This is done by using US transducers (Transmitter and receiver pairs) in various setups and measure the time it takes for the receiver to recognize the signal sent out by the transmitter. Depending on the geometry of the transducers relative placement in conjunction with information about which medium the sound transverses, the distance is calculated between the transducer and the object.

Applying US technology for other purposes than imaging may open new propositions for the suitability of this technology in varying context, possibly making use of more of US properties.

1.2 Purpose

The purpose of this project is to investigate the potential of using US transducers applied in an object tracking context.

In order to do so, the following research questions must be answered:

- How can US transducers be set up geometrically to allow object tracking?
- How can the information relayed by the US transducers be used to control the movement of the receivers?
- How does the algorithm impact the following potential of the setup?
To what degree does US technology allow for obstructions between transmitter and receiver?

1.3 Scope

The main objective of this thesis is to construct a land vehicle capable of following its user using ultrasound. This thesis will also supply a guideline and reference point on how to construct a robot that is capable of following a user using ultrasonic sensor and receiver pairs. It will be evaluated based on how effective such a system is in relation to the various constraints such a system imposes on its utility. The project is limited in scope to a single robot following a user and stopping at a given distance from the user, and does not attempt to find a solution for several robots operating in the same environment or an environment with other sources of ultrasound emitters.

1.4 Method

Initially, the included components shall be tested. This is done to empirically determine the constraints respective to their individual application, which will lead to a suitable construction method. Once the vehicle is finalized, another test will be run to figure out the constraints of the system in its entirety. These constraints will determine how suitable the robot is in the application of following an object. This result will then express the suitability of applying US technology for tracking purposes.

1.4.1 Tracking

The tracking is to be implemented by three ultrasound receiver sensors on the robot. The sensors detection area gives three slightly overlapping zones, as they are slightly angled from each other. This will be further explained in chapter 3. The user will have an ultrasound transmitter. When the ultrasound from the transmitter reach the ultrasound receivers, the robot will be able to determine if it’s in an allowed zone relative to the user. The robot can then make appropriate corrections.
Chapter 2

Theory

This chapter will briefly explain some of the theories applied when creating the demonstrator, or otherwise relevant to the outcome of this thesis.

2.1 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) is a commonly used technique in order to vary the effective output voltage from a Direct Current (DC) source. The idea behind PWM is that instead of sending a continuous signal, the signal is only allowed to be on during a fraction of the time. Thus, the output signal will have an average that is lower than the continuous signal. This effectively lowers the amplitude of the perceived signal received by the end device. [1]

2.2 Ultrasound (US)

Sound moves in its medium in waves and the most common medium on land is air. These sound waves have many frequencies depending on how the sound was created. Most of the frequencies are inaudible for humans, who can only hear in the slim band of 20 Hz to 20kHz. Ultrasound consists of sound from 20kHz and up, meaning it’s inaudible for humans. This makes it an invisible and silent way of transferring information[2].

2.2.1 Diffraction

When sound encounters a barrier, the soundwave have the ability to bend around the barrier. This phenomenon is called diffraction and is the reason that sound can be heard even when the source of the sound and recipient isn’t in line of sight.
The sound’s ability to diffract is dependent on the wavelength of the wave and which in turn is dependent on the frequency and speed of sound. As described by equation 2.1

$$\lambda = \frac{v}{f}$$  \hspace{1cm} (2.1)

where \(v\) is the speed of sound, \(f\) the frequency and \(\lambda\) the wavelength. The sound speed is highly dependent on many factors such as medium, air pressure, humidity and temperature. Under the conditions that it’s room temperature, at sea level, low humidity and the medium is air, the speed can be approximated to 344 m/s [3].

The amount of diffracted sound is as previously stated dependent on wavelength, where longer wavelengths are able to diffract more than shorter [2], [4]. Since frequency and wavelength are inversely proportional a low frequency gives a high wavelength and high frequency a low wavelength.

### 2.3 DC motors

A DC motor converts electrical power into mechanical through the use of a commutator, coil and magnets. The wire coil is wrapped around a piece of iron which is located at the center of the motor. Around it are several magnets located. When electrical power is connected to the commutator, current flows through the coil which produces a magnetic field. The strength and direction of the magnetic field can be controlled by the direction and amplitude of the current that passes through the coil. This magnetic field influences the magnets and thus creating torque which forces the motor to spin. Depending on the direction the current flows, the motor will spin either clockwise or counterclockwise [5].

### 2.4 H-Bridge

The H-bridge is an electrical component for changing a current’s direction, this is achieved by opening and closing gates. The current’s direction decides the rotational direction of the DC motors.

A H-bridge which receives a signal from a microcontroller and depending on the signal aligns a current. This current is used to power the motors. The H-bridge allows the controller to run the motors at different speeds and directions. By varying speed and direction, the controller can turn the robot.

### 2.5 Microcontroller

Microcontrollers can be viewed as a self-contained processing unit including a Central Processing Unit (CPU), Read Only Memory (ROM) and Random Access Memory (RAM) as well as other peripherals for connecting other devices. These vary in
both size and cost but also in the amount of built-in peripherals such as including an A/D converter or LED lights. The way that the different parts of the microcontroller communicate is via a connection called bus. Defining how the connections between the components and the bus is done as well as the rules regulating it’s use is done according to the standard called \textit{I}^2\textit{C}. It is described how multiple devices may use the same physical wire to communicate and how the selection is done according to which device may send data at any given time [6].
Chapter 3

Demonstrator

This chapter describes the components that make up the system as a whole, broken down into various sections that each show their respective part. The main parts are software and hardware which combine to create the robot as it is. Further explanation about various circuits built can also be found in this chapter.

Figure 3.1. Visual representation of the entire system
3.1 Problem Formulation

To be able to answer the research question, a ground vehicle was built. In order to account for its suitability in tracing objects, a variety of questions need to be answered:
- How will the vehicle determine the user’s location?
- How does the vehicle move in order to approach its target?
- How can the vehicle stop at a distance from the user?

3.2 Software

This section describes the overall programming implemented in order to make the system work. Details about the code can be found in appendix A.

3.2.1 Robot

The robot is equipped with three ultrasound receivers which point in different directions. These receivers cover the front area of the robot. The receivers each transmit a signal to the micro-controller. The code compares the signals and decides based on the the signal strength where the sender is. If the middle receiver receives the strongest signal the robot moves toward the signal thus following the user. When the signal is strongest on either of the outer receivers the robot is not aligned with the sender. The robot will then turn until the signal is once again strongest on the middle receiver. To keep the robot from driving into the user there is an upper limit on the signal. When the limit is reached, the robot stops. Since the signal strength is dependent on distance, the signal only reaches the limit when transmitter and receiver are very close together. Figure 3.2 depicts the logical flow for the program controlling the robot.
3.2. SOFTWARE

In order to allow the robot to run smoothly, a delay was introduced after the measurement. When implementing the delay, the set parameters for the motors will continue to run until the new readings have been forwarded to the Arduino, which will then set the updated parameters for the motors. If the delay is too long, the reactivity of the system will lower. If the delay is too short, the robot will display a stoccato.

3.2.2 Transmitter

In order to generate the desired pulses for the transmitter, the Arduino needs to open and close the transistor. Since the desired frequency is 40kHz, it needs to emit a PWM at that frequency. The internal timers of the Arduino do not support such a high frequency by default, and thus modifications to its internal programming needed to be made. The reprogramming of the timers allowed for a 50% duty cycle PWM to be transmitted from pin 9.
3.3 Electronics

3.3.1 Transmitter circuit

For optimal performance, the transmitter requires a square-wave pulse with the frequency of 40 kHz. To create the pulse, the transmitter is connected to a transistor which acts as a fast paced on/off switch. The transistor is controlled by the Arduino which emits a 40 kHz PWM signal from pin 9, as described in section 3.2.2.

In order to allow the transmitter to properly reset between pulses, a parallel circuit consisting of a diode and two parallel resistors is attached. To allow the transmitter to reset fast enough, a low resistance is desired, although since the circuit is high voltage they get warm over time. This is mitigated by having two parallel resistances according to equation 3.1

$$\frac{1}{R_{\text{tot}}} = \frac{1}{R_1} + \frac{1}{R_2}$$

(3.1)

A schematic of how the transmitter circuit is constructed can be found in appendix D.

3.3.2 Receiver circuit

The receivers mounted on the robot is connected to an amplifier circuit as shown in appendix D.

The circuit enhance the receiver’s signal strength before it reaches the microcontroller. Since the receiver create an AC in the form of a sine-wave and the micro-
3.4. HARDWARE

controller only reads positive current a rectifier was used. The rectifier takes the
negative parts of the sine-wave and turn them into a equivalent positive current.

3.3.3 H-bridge

The main component for the H-bridge is the L298N module which directs the current
depending on the input signals it receives [7]. Schematics for the H-bridge can be
found in appendix D.

3.4 Hardware

In this section, it is shown what components are being used together with a description
of each component’s properties.

The robot consists of a plastic chassis onto which the other components are
attached. A simplified schematic of how the components are attached can be found
in appendix B.

Components belonging to the robot include the microcontroller Arduino UNO, dual
H-bridge to control the DC motors, receiver circuits, receiver holders, Li-Po battery
pack to power the DC motors as well as a battery pack to power the microcontroller.

The sender unit consists of an acrylic casing with a retractable plate onto which
the other components are attached. A schematic depicting an overview of how this
is done can be found in appendix E.

Components attached to the sender unit comprise of the microcontroller Arduino
UNO, sender circuit, sender holder, battery pack to power the sender circuit as well
as a battery pack to power the microcontroller.

3.4.1 Arduino UNO

The microcontroller Arduino UNO was used in this project. This microcontroller
was chosen due to it being open source and supplying the amount pins required for
the project.

To power the Arduino, battery packs consisting of 6 AA batteries each, were used.
Arduino code was used to create the algorithm controlling the sender circuit and
receiver circuit.

3.4.2 DC motor

The DC motors used in this project are DAGU DG02SA130GEARMOTOR [8].
These were controlled using an H-bridge as described in section 3.3.3. Since the
H-bridge can only control two motors simultaneously and the robot has four, it was
imperative to connect two motors on each side to act as a pair. Thus, the robot
will have two pairs of motors; on the left and on the right side respectively.
3.4.3 Receiver Circuit

The receivers are placed facing the front of the robot yet slightly angled upwards through the use of the receiver holder showcased in figure E.8. The holders are placed on the platform which allows the outer receivers to be angled slightly away from the middle, thus covering a broader area and creating three overlapping areas. Drawings for the platform can be found in figure E.19.

In order to properly read the signals picked up by the US receivers, a receiver circuit was built. The circuit is built around the amplifier LM358P [9]. The amplifier is needed in order to create a stronger signal for the Arduino to read which in turn allows for a greater range of the receiver. Schematics for the receiver circuit can be found in appendix D.

3.4.4 Transmitter Circuit

The transmitter signal strength is dependant on voltage level. In order to achieve a higher voltage than the microcontroller can supply, a transistor of type TIP 141 was used [10]. The transistor is controlled by the microcontroller and connected to the transmitter and a separate power supply of higher voltage. The microcontroller’s pulse opens and closes the transistor thus creating the square wave for the transmitter.

3.4.5 US transducer

The US transducers used in this project are Murata MA40S4 [11]. These were chosen due to the transducers being directional and covering a distance of up to four meters.

The transmitter requires a pulse ranging from -20V to +20V in order to oscillate. The oscillation is supposed to happen at a frequency of 40kHz which will induce the US wave to be generated and propelled through the medium. The strength of the signal will vary depending on the input voltage.

The receiver is influenced by the US wave generated by the transmitter and starts oscillating accordingly. The receiver will generate an electric sine wave with the frequency of 40kHz and varying amplitude. The amplitude depends on how strong the received signal is. The strength of the received signal is dependant on the distance from the transmitter. Thus, a weaker signal is received if the transmitter is further away.
3.4. HARDWARE

3.4.6 Battery packs
Due to the different components requiring varying voltage levels to work optimally, it is necessary to have four different power supplies. Two of the battery packs consist of Lithium PP3 9V batteries used to power the microcontrollers. One battery pack consists of two 9V batteries connected in series resulting in a 18V source to power the sender circuit. One Li-PO battery pack is used to power the DC motors [12].

3.4.7 Mechanical parts
All the various holders, screens for the transducers as well as the acrylic casing were custom made for this project. The acrylic parts were laser cut using the Epilog Laser Fusion M2 and the plastic parts were made using the 3D printer Ultimaker 2. Drawings for the designs can be found in appendix E.
Chapter 4

Results

4.1 Testing US transmitter/receiver

To test the impact of the amplifying circuits and screens on the signal, the signal was measured with an oscilloscope for three configurations of transmitter and receivers respectively. This is to create the best possible signal which in turn contributes to a better performance of the system. The signal strength is the main variable regarding the limits to the allowed area where the user must be inside.

The configurations was the following:

- oscilloscope generated wave for transmitter; unmodified receiver
- oscilloscope generated wave for transmitter; receiver with screen
- oscilloscope generated wave for transmitter; receiver with screen and amplifier circuit
- transmitter with amplifier circuit and screen; unmodified receiver
- transmitter with amplifier circuit and screen; receiver with screen
- transmitter with amplifier circuit and screen; receiver with screen and amplifier circuit

The results are presented in table 4.1.
CHAPTER 4. RESULTS

<table>
<thead>
<tr>
<th>Transmitter configuration</th>
<th>Receiver configuration</th>
<th>Signal strenght</th>
</tr>
</thead>
<tbody>
<tr>
<td>oscilloscope generated wave</td>
<td>unmodified receiver</td>
<td>450 mV</td>
</tr>
<tr>
<td>oscilloscope generated wave</td>
<td>screen</td>
<td>600 mV</td>
</tr>
<tr>
<td>oscilloscope generated wave</td>
<td>screen and amplifier circuit</td>
<td>2250 mV</td>
</tr>
<tr>
<td>amplifier circuit and screen</td>
<td>unmodified</td>
<td>1500 mV</td>
</tr>
<tr>
<td>amplifier circuit and screen</td>
<td>screen</td>
<td>2500 mV</td>
</tr>
<tr>
<td>amplifier circuit and screen</td>
<td>screen and amplifier circuit</td>
<td>2500 mV</td>
</tr>
</tbody>
</table>

Table 4.1. Signal strength dependency on configuration

The result clearly shows that both the screens and amplification circuits magnifies the signal strength. Further data can be found in appendix C.

4.2 Testing the System

The test design for the system consists of two tests. For the first test, the robot is placed stationary at a fix point. The transmitter was placed on a movable table at the height of 1m. The table was then moved radially in order to find the furthest point where the robot still picks up the signal. This point was defined as when the motors were turning for at least half the time, in the sense that it picks up the signal over 50% of the time. Once the furthest point along one line was found, the table was moved laterally and then adjusted so that the robot picks up the signal. This was repeated a number of times and results can be found in figure 4.1.
4.2. TESTING THE SYSTEM

Figure 4.1. Allowed Area, created in the software Matlab
CHAPTER 4. RESULTS

This test describes the area within which the user must be located in order for the system to work as intended. The shape of the area shows the outer boundaries, which are the restrictions imposed on the user when operating the system.

The second test comprises of a stationary robot where the transmitter is located at or near the edge of the allowed area described previously. Between the transmitter and receiver an object will be placed. This object will move linearly between the transmitter and receiver in order to see when it blocks the signal. This will create an area of where obstacles are allowed in order to not interfere with the robots functionality.

As can be seen in figure 4.2, the tests show that both the form and size of the object matter. The smaller object disturbed the signal less then the larger and the cylindrical performed better then the cubic.

![Figure 4.2. Obstacle interference on signal, where the marked area for cylinder and cuboid represent signal contact, created in the software Matlab](image)

18
4.2. TESTING THE SYSTEM

This result is confirmed by diffraction theory of sound, which states that when sound encounters a barrier or object it will attempt to bend around it. The sound’s ability to bend around the object is affected by the frequency, where high frequency sounds do not diffract as easily as low frequency sounds. Thus when high frequency sound encounters a barrier it often creates a sound shadow behind the object where the sound can’t reach. This sound shadow is the reason the robot couldn’t find the signal when behind an object [4].

To determine how the range between transmitter and receiver is affecting the result, the above described test was performed a second time with a new shorter distance between transmitter and robot.

In that case the small cylinder was the only object where the robot could pick up the signal.

The result of this test shows the possibilities of how the system allows for foreign objects to be placed in its path, yet still work.
Chapter 5

Discussion and conclusions

5.1 Discussion

The main topic for discussion is the angular dependency of the US transducers. This phenomena directly impacts the performance of the system as it affects the three zones described in section 1.4.1.

In order to limit the field of vision that the receivers have, special screens were created and attached to the receivers. This led to a slight increase in the received signal strength. A similar but shorter sleeve was designed for the transmitter. Since the robot’s height is 10 cm and the sender circuit is a hand-held device, it is important to direct the receivers slightly upwards in order to face the projected location of the transmitter. For this reason, the Receiver Holder was constructed.

Although these measures did increase the efficiency of the robot, in the sense that it made it easier for the receivers to pick up the signal, the system still requires the user to align the transmitter with the receivers mounted on the robot. Therefore, there are constellations in which the robot can not pick up the signal even though the transmitter is within the boundaries described in figure 4.1, such as if the transmitter is aligned horizontally when standing close to the robot resulting in the transmitter aiming above the receivers.

Another point of discussion is the designed movement of the robot. The speed of the motors has been set and will remain constant regardless of the distance to the transmitter. This means the robot will always turn the same amount of degrees when the turning sequence is initiated. If the transmitter is further away, only small corrections might be necessary in order to align the robot towards the transmitter. Therefore, turning too far may happen for some constellations where the transmitter is relatively far away and only slightly to one side. To mitigate these effects, the turning sequence intends to only turn the robot slightly when initiated and readings to happen relatively frequently. This in turn affects the reactiveness of the robot when the transmitter is relatively close.

A related possible issue is that the speed of the robot when moving forward has been set. Since the objective of the robot is to be at a given distance from the
transmitter at all times, it may be beneficial to increase the speed when the trans-
mitter is relatively far from the robot and decrease the speed when the transmitter
is closer.

The fact that the US signal is only differentiated by its frequency may also
cause problems in environments where there are multiple identical robots operating
simultaneously or where there are other US emitters. This introduces limitations
to the implementation of multiple robots.

It may be possible to encode the signals so that a specific pair of transmitter and
receiver can be created which may circumvent this issue, although this was not ex-
plored in this project.

5.2 Conclusions

The purpose of this thesis is to answer the research questions mentioned in section
1.2. Throughout this paper it has been described how a number of hardware items
may be physically placed in combination with a description of how software can
be implemented which will produce a robot capable of following it’s user. A US
transmitter is being carried by the user, aimed towards the robot. The robot is fitted
with US receivers which pick up the signal and forward it to the microcontroller.
It is shown that signals from US receivers can be interpreted by a microcontroller
which in turn controls a robot so that it tracks the US emitter. This is done using
the Arduino’s built-in AD converter, and using the integer value produced as input
data for the algorithm controlling the robot’s movement. The algorithm contains a
delay function which helps in keeping the robot’s movement smooth, as a too short
delay would result in the robot displaying a start-and-stop behaviour, impacting
the user experience. The PWM signal used for controlling the motors needs to be
adjusted according to the delay function, since a PWM signal that turns the motors
too quickly may cause the robot to move too far in either direction before a new
measurement is done. These two factors create limits for the system in terms of
maximum velocity of the user, meaning the user may not move too fast away from
the robot or it will lose track.

The measurements are made in numbers of five, and then averaged in order to even
out possible reading spikes incurred through the geometry or other factors hard to
predict. This also helps when objects are placed in it’s path as described in section
4.2.
Chapter 6

Recommendations and future work

Although the robot is able to follow the transmitter given that it is aligned properly and within the specified area, there are modifications and additions that can be made in order to improve the system as a whole.

In order to make the robot move more smoothly, a more dynamic editing of the PWM signals to control the motors can be done, possibly by implementation of a PID controller. To increase the range, different US transducers can be implemented. One suitable replacement may be the MA40B8R/S [11], since it already has a smaller FOV and longer range. Alternatively, a greater amplification of the receiver signal may be achieved if filtered appropriately or a stronger transmitting signal by using the entire allowable spectrum from -20V to +20V rather than 0V to +18V as was used in this project.

To fully make use of the US property of allowing obstacles in it’s path, it may be beneficial to introduce another set of sensors which detect the objects and will then allow the programming of obstacle avoidance. It would be recommended to use IR technology for this, for example the SHARP GP2Y0A02YK0F [13], as it would not interfere with the US technology used for the tracking.
Bibliography


Appendix A

Arduino Code

A.1 Transmitter code

This code modifies the internal clock-frequency of the Arduino [14]. It is used for the transmitter circuit to allow it to send a 40 kHz PWM for controlling the transmitter as described in section 3.2.2.


void analogWriteSAH_Init ( void )
{

    TCCR1B = (0 << ICNC1) | (0 << ICES1) | (0 << WGM13) | (0 << WGM12) |
        (0 << CS12) | (0 << CS11) | (0 << CS10);

    TCCR1A =
        (1 << COM1A1) | (0 << COM1A0) | // COM1A1, COM1A0 = 1, 0
        (0 << COM1B1) | (0 << COM1B0) |
        (1 << WGM11) | (0 << WGM10); // WGM11, WGM10 = 1, 0

    // Set TOP to...
    //
    // fclk_I/O = 16000000
    // N = 1
    // TOP = 799
    //
APPENDIX A. ARDUINO CODE

// fOCnxPWM = fclk_I/O / (N * (1 + TOP))
// fOCnxPWM = 16000000 / (1 * (1 + 799))
// fOCnxPWM = 16000000 / 800
// fOCnxPWM = 20000

ICR1 = 399; // modified from retrieved code to make it 40kHz

// Ensure the first slope is complete
TCNT1 = 0;

// Ensure Channel B is irrelevant
OCR1B = 0;

// Ensure Channel A starts at zero / off
OCR1A = 0;
// We don’t need no stinkin interrupts
TIMSK1 = (0 << ICF1E) | (0 << OCIE1B) | (0 << OCIE1A) | (0 << TOIE1);

// Ensure the Channel A pin is configured for output
DDRB |=(1 << DDB1);

// Start the timer...

// CS12 CS11 CS10 Description
// −−−− −−−− −−−−−−−−−−−−−−−−−−−−−−−−−−−−
// 0 0 1 clkI/O/1 (No prescaling)

TCCR1B =
(0 << ICNC1) | (0 << ICES1) |
(1 << WGM13) | (1 << WGM12) |
(0 << CS12) | (0 << CS11) | (1 << CS10); // WGM13, WGM12 = 1, 1

void setup()
{
analogWriteSAH_Init();
}

void loop()
{
analogWriteSAH(250);
}

void analogWriteSAH(uint16_t value)
{
A.1. TRANSMITTER CODE

if ( (value >= 0) && (value < 400) )
    OCR1A = value;
}
A.2 Robot code

This code is used for controlling the robot, interpreting the incoming US signals as well as controlling the motor output signals [15].

// The basis for this code, with the purpose of controlling the H-bridge, was supplied by Fredrika Kringberg on February 2017, via the internal canvas forum.
// https://kth.instructure.com/courses/837/discussion_topics/3530

// Connect the motor controller pins to Arduino digital pins
/* Not working?
 * --> Doublecheck connections
 * --> Is Arduino grounded to the rest of the circuit?
 * --> Are the diodes mounted in the right direction?
*/
// initiate all needed pins and variable names
int distance;

// all relative positioning (left/right) is seen from robot POV
// motor one − LEFT (H-bridge side)
  int enA = 10;
  int in1 = 9;
  int in2 = 8;
// motor two − RIGHT (arduino side)
  int enB = 5;
  int in3 = 7;
  int in4 = 6;
int pwm_A;
int pwm_B;
bool turn = true;

//US
  int US1 = 0;
  int US2 = 0;
  int US3 = 0;
  int US1_pin = A0;
  int US2_pin = A1;
  int US3_pin = A2;
void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
A.2. ROBOT CODE

```c
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
Serial.begin(9600);
}
// loops the program
void loop()
{
  US1 = USRead1(); // calls of function that reads US reciver 1
  // String stringone = "US1";
  // Serial.println(stringone);
  // Serial.println(US1);

  US2 = USRead2();
  // String stringtwo = "US2";
  // Serial.println(stringtwo);
  // Serial.println(US2);

  US3 = USRead3();
  // String stringthree = "US3";
  // Serial.println(stringthree);
  // Serial.println(US3);

  // compares signal strengh to decide appropriate action
  // if signal strength is stronger then 60 the robot stands still
  if (US2 > 60){
    pwm_A = 50; // pwm 50 represent robot standing still
    pwm_B = 50;
  }
  else if (US1 < 5 && US2 < 5 && US3 < 5){
    pwm_A = 50;
    pwm_B = 50;
  }
  else if (US2 > US1 && US2 > US3){
    pwm_A = 65; // set pwm for motor A: 0–100
    pwm_B = 65; // set pwm for motor B: 0–100
  }
  else if (US1 > US2 && US1 > US3){
    pwm_A = 35; // set pwm for motor A: 0–100
    pwm_B = 70; // set pwm for motor B: 0–100
  }
  else if (US3 > US2 && US3 > US1){
```
APPENDIX A. ARDUINO CODE

```
pwm_A = 65; // set pwm for motor A: 0–100
pwm_B = 30; // set pwm for motor B: 0–100
}
if (pwm_A == 50)
{
    enA_clockwise(map(pwm_A, 50, 100, 0, 0));
}
else if (pwm_A < 50)
{
    enA_counter_clockwise(map(pwm_A, 0, 50, 0, 255));
}
else if (pwm_A > 50)
{
    enA_clockwise(map(pwm_A, 50, 100, 0, 255));
    // Serial.println(map(pwm_A, 50, 100, 0, 255));
}
if (pwm_B == 50)
{
    enB_clockwise(map(pwm_B, 50, 100, 0, 0));
}
else if (pwm_B < 50)
{
    enB_counter_clockwise(map(pwm_B, 0, 50, 0, 255));
}
else if (pwm_B > 50)
{
    enB_clockwise(map(pwm_B, 50, 100, 0, 255));
    // Serial.println(map(pwm_B, 50, 100, 0, 255));
}
// test code: turn motor A from min to max
/* if (turn)
{
    pwm_A += 10;
} else {pwm_A -= 10;}
*/
delay(500);

// following four functions control the engines
// rotational direction and speed
void enA_counter_clockwise(int PWM)
{
    digitalWrite(in2, LOW);
    digitalWrite(in1, HIGH);
```
A.2. ROBOT CODE

```c
analogWrite(enA, (255 - PWM));
}
void enA_clockwise(int PWM)
{
    digitalWrite(in2, HIGH);
    digitalWrite(in1, LOW);
    analogWrite(enA, PWM);
}

void enB_counter_clockwise(int PWM)
{
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(enB, (255 - PWM));
}

void enB_clockwise(int PWM)
{
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enB, PWM);
}

// US receiver reads signal strength, one for each receiver
int USRead1()
{
    int averaging = 0; // Holds value to average readings

    // Get a sampling of 5 readings from sensor
    for (int i = 0; i < 5; i++) {
        distance = analogRead(US1_pin) - 140;
        //distance2 = analogRead(US2_pin);
        averaging = averaging + distance;
        //delay(5); // Wait 55 ms between each read
    }
    distance = averaging / 5; // Average out readings
    return (distance); // Return value
}

int USRead2() {
    int averaging = 0; // Holds value to average readings

    // Get a sampling of 5 readings from sensor
    for (int i = 0; i < 5; i++) {
        //distance1 = analogRead(US1_pin);
```
APPENDIX A. ARDUINO CODE

```c
#include <Servo.h>
#include <LiquidCrystal.h>

int servoPin = 9;
int US2 = A0;
int US3 = A1;

int averaging = 0;

void setup() {
  digitalWrite(servoPin, HIGH);
  servo.attach(servoPin);
  LiquidCrystal lcd(12, 11, 5, 6, 4, 3);
}

void loop() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Reading US2");
  lcd.setCursor(0, 1);
  lcd.print("Reading US3");
  float distance2 = USRead2();
  float distance3 = USRead3();
  lcd.print(distance2);
  lcd.setCursor(16, 0);
  lcd.print(distance3);
  lcd.setCursor(0, 1);
  lcd.print("\n\nAverage: ");
  float average = (distance2 + distance3) / 2;
  lcd.print(average);
  servo.write(average * 90 / 512);
}

float USRead2() {
  int distance = analogRead(US2) - 140;
  averaging = averaging + distance;
  delay(5); // Wait 55 ms between each read
  return (distance); // Return value
}

int USRead3() {
  int averaging = 0;
  // Holds value to average readings

  // Get a sampling of 5 readings from sensor
  for (int i = 0; i < 5; i++) {
    distance = analogRead(US3) - 140;
    averaging = averaging + distance;
    delay(5); // Wait 55 ms between each read
  }
  distance = averaging / 5; // Average out readings
  return (distance); // Return value
}
```

34
Appendix B

Hardware Coupling
Figure B.1. Transmitter wiring, made with the *fritzing* software.
Figure B.2. Transmitter wiring, made with the fritzing software
Appendix C

Testing

The setup for testing the US transducers is as follows:
The transmitter and receiver are facing each other and will be placed once on distance 10.5 cm apart, and once on distance 30 cm apart.
For the first test, a transmitter not fitted with the transmitter screen (naked transmitter) is powered by an oscilloscope exerting a 50% duty cycle PWM with an amplitude of 5V at 40kHz. It will face a receiver not fitted with the receiver screen (naked receiver) nor connected to the receiver circuit (before amplification).
For the second test, a naked transmitter is facing a receiver fitted with the receiver screen.
The third test involves a naked transmitter facing a receiver fitted with the receiver screen as well as connected to the receiver circuit (after amplification).
These tests were then repeated with transmitters powered by the transmitter circuit and fitted with the transmitter screen (sleeve).
The above procedure was then repeated where the transducers were at a distance of 30 cm apart. Raw data can be found in table C.1.
### Table C.1

Raw data from the US testing as described above

<table>
<thead>
<tr>
<th>Transmitter</th>
<th>Receiver</th>
<th>Distance [cm]</th>
<th>Signal Strength [mV]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naked</td>
<td>Naked</td>
<td>10.5</td>
<td>450</td>
</tr>
<tr>
<td>Naked</td>
<td>before Amp</td>
<td>10.5</td>
<td>600</td>
</tr>
<tr>
<td>Naked</td>
<td>after Amp</td>
<td>10.5</td>
<td>2250</td>
</tr>
<tr>
<td>Sleeve</td>
<td>Naked</td>
<td>10.5</td>
<td>1500</td>
</tr>
<tr>
<td>Sleeve</td>
<td>before Amp</td>
<td>10.5</td>
<td>2500</td>
</tr>
<tr>
<td>Sleeve</td>
<td>after Amp</td>
<td>10.5</td>
<td>2500</td>
</tr>
<tr>
<td>Naked</td>
<td>Naked</td>
<td>30</td>
<td>160</td>
</tr>
<tr>
<td>Naked</td>
<td>before Amp</td>
<td>30</td>
<td>200</td>
</tr>
<tr>
<td>Naked</td>
<td>after Amp</td>
<td>30</td>
<td>1430</td>
</tr>
<tr>
<td>Sleeve</td>
<td>Naked</td>
<td>30</td>
<td>550</td>
</tr>
<tr>
<td>Sleeve</td>
<td>before Amp</td>
<td>30</td>
<td>740</td>
</tr>
<tr>
<td>Sleeve</td>
<td>after Amp</td>
<td>30</td>
<td>2300</td>
</tr>
<tr>
<td>Obstacle allowance [cm]</td>
<td>0 [single point]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obstacle</td>
<td>L. Cuboid</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total distance: 194</th>
<th>Measured distance from:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>robot [cm]</td>
<td>transmitter [cm]</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total distance: 194</th>
<th>Measured distance from:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>robot [cm]</td>
<td>transmitter [cm]</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>112</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obstacle allowance [cm]</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>L. Cylinder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total distance: 194</th>
<th>Measured distance from:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>robot [cm]</td>
<td>transmitter [cm]</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>115</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obstacle allowance [cm]</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>S. Cuboid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total distance: 194</th>
<th>Measured distance from:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>robot [cm]</td>
<td>transmitter [cm]</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obstacle allowance [cm]</th>
<th>45.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>S. Cylinder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total distance: 139</th>
<th>Measured distance from:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>robot [cm]</td>
<td>transmitter [cm]</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obstacle allowance [cm]</th>
<th>49.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>S. Cylinder</td>
</tr>
</tbody>
</table>

Other geometries for total distance 139 cm: No configuration found where they are allowed.

Figure C.1. Raw data from the Obstacle allowance test
### Figure C.2.
Specifications of the dimensions of objects used when testing obstruction allowance

<table>
<thead>
<tr>
<th>Object</th>
<th>Width [cm]</th>
<th>Height [cm]</th>
<th>Depth [cm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Cuboid</td>
<td>6</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>L. Cuboid</td>
<td>7.8</td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>S. Cylinder</td>
<td>6.5</td>
<td>10</td>
<td>6.5</td>
</tr>
<tr>
<td>L. Cylinder</td>
<td>8</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>Point</td>
<td>x(cm)</td>
<td>y(cm)</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-93,4</td>
<td>89,6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-101,7</td>
<td>122,8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-99,9</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-113,4</td>
<td>159,3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-92,1</td>
<td>198,6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-84,5</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-75</td>
<td>180,5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-64,2</td>
<td>178,5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-59,4</td>
<td>185,6</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-53,5</td>
<td>207,5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-43,6</td>
<td>211,3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-37,4</td>
<td>229,8</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-22,1</td>
<td>238,8</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-11,2</td>
<td>243,6</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>254,5</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>7,3</td>
<td>249,2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>11,7</td>
<td>238,7</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>29,4</td>
<td>228,2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>39</td>
<td>227,4</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>45,8</td>
<td>203,6</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>49,2</td>
<td>201,2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>53,4</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>57,4</td>
<td>200,4</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>60,3</td>
<td>184,7</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>69,8</td>
<td>183,7</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>79,3</td>
<td>195,8</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>90,7</td>
<td>194,6</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>105,2</td>
<td>194,7</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>119,2</td>
<td>201,4</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>129,3</td>
<td>202</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>135,4</td>
<td>192,6</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>130</td>
<td>169</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>127,7</td>
<td>152,3</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>126,3</td>
<td>132,8</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>113,5</td>
<td>119,9</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>111,8</td>
<td>104,6</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>108</td>
<td>91,9</td>
<td></td>
</tr>
</tbody>
</table>

**Figure C.3.** Raw data from the test to determine the allowed area
Appendix D

Circuit Schematics
Figure D.1. Transmitter circuit PCB, made in the *Eagle* software
Figure D.2. Receiver circuit PCB, made in the *Eagle* software
Appendix E

Hardware Schematics
Figure E.1. Holder used for attaching H-bridge to robot as well as transmitter circuit to Sleeve Base, created in the SolidEdge, Academic version software.

APPENDIX E. HARDWARE SCHEMATICS

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

NAME
Olsson, Hasenson

DATE
05/15/17

MF133X

REVISION HISTORY

REV DESCRIPTION DATE APPROVED

NAME
Olsson, Hasenson

DATE
05/15/17

REVISION HISTORY

REV DESCRIPTION DATE APPROVED
Figure E.2. US Holder, created in the SolidEdge, Academicversion software
Figure E.3. Sender Screen, created in the SolidEdge, Academicversion software.
Figure E.4. Receiver Screen, created in the SolidEdge Academic version software.
Figure E.5. Handle, created in the SolidEdge, Academicversion software
Figure E.6. Bottom Hinge, created in the SolidEdge, Academicversion software
Figure E.7. Top Hinge, created in the SolidEdge, Academicversion software
Figure E.8. Receiver Base Holder, created in the SolidEdge, Academicversion
Figure E.9. Receiver Holder Offset, created in the SolidEdge, Academic version
Figure E.10. Sender Lid, created in the SolidEdge, Academicversion software
Figure E.11. Sleeve Attachment, created in the SolidEdge, Academic version software.
Figure E.12. Sleeve Base, created in the SolidEdge, Academic version software.
Figure E.13. Sleeve Base Offset, created in the SolidEdge, Academic version software.
Figure E.14. Sleeve Bottom, created in the SolidEdge Academic version software.
Figure E.15. Sleeve Ledge, created in the SolidEdge, Academic version software
Figure E.16. Sleeve Lid, created in the SolidEdge Academic version software.
Figure E.17. Sleeve Side, created in the SolidEdge, Academic version software.
Figure E.18. Sleeve Top, created in the SolidEdge Academicversion software.
Figure E.19. Receiver Holder, created in the SolidEdge, Academicversion software