Best PAL

Ball Exercise Sound Tracking PAL

AXEL SUNDKVIST

JOAKIM HELLBERG
BEST PAL
Ball Exercise Sound Tracking PAL

Bachelor Thesis

AXEL SUNDKVIST
JOAKIM HELLBERG

Bachelor’s Thesis at ITM
Supervisor: Jessica Landström
Examiner: Nihad Subasic

TRITA-ITM-EX 2018:66
Abstract

The PAL (Practise and Learn) Original is a ball board consisting of three wooden boards placed in a triangle, developed to practise football players’ passing ability and first touch. The former Swedish international footballer Jessica Landström observed that these ball boards can, if they are improved, help footballers to develop even more skills while practicing alone. Landström’s idea was to put lamps on top of the ball boards which light up when a certain ball board expects to receive a pass. This would force the player to look up instead of looking at the ball and hence improve their vision.

We concluded that speaking also is important within football. So our objective became to follow up on the development of the simple PAL Original to a ball board which rotates towards a sound source. We wanted to achieve this without configuring the PAL Original's construction.

With the purpose of executing the idea we needed to estimate the angle between a sound source and a face of the ball board, rotate the ball board with an electric motor, communicate wirelessly between units and detect a ball hit when the ball board receives a pass.

The final prototype consists of two systems, one system executing the sound source localization and rotation and the other system executing the detection of ball hit and wireless communication.

The first system uses time difference of arrival (TDOA) between incoming sound for three sound sensors to calculate an angle, which in turn is communicated to a DC motor that executes the rotation.

The other system combines an LED to light up when a pass is expected, an accelerometer to detect a pass, and radio transceivers to communicate with each other. When at least three of these devices are used a randomizing algorithm decides which one should light its LED next when the first one detects a pass.

**Keywords**— mechatronics, microcontroller, sound source localization, wireless communication, DC motor controlling
Referat

Ljudlokaliserande smart bollplank för individuell fotbollsträning


Den slutgiltiga prototypen består av två system, ett system som utför lokaliserings av ljudkälla samt rotation och ett system som utför detektering av bollträff samt hanterar trådlös kommunikation. Det första systemet utnyttjar tidsskillnad för ankomst, TDOA (Time Difference of Arrival), mellan inkommande ljud till tre ljudsensorer för att beräkna en vinkel, som i sin tur kommuniceras till en likströmsmotor som utför rotationen.

Det andra systemet kombinerar en lysdiod som lyser när en passning förväntas, en accelerometer för att detektera att passning mottagits och radiosändare samt mottagare för trådlös kommunikation. När minst tre sådana enheter används, bestämmer en slumpgenerator vilken enhet som ska tända sin lysdiod när den första detekterar en passning.

Nyckelord— mekanik, mikrokontroller, ljudlokalisering, trådlös kommunikation, likströmsmotorstyrning
Acknowledgements

We would like to thank Jessica Landström for her idea and her support throughout this project. She has helped us with knowledge about football and also with reading and improving this thesis paper.

We would also like to thank our colleagues and friends Erik Blomqvist and Niclas Bassili for their help and support. Philip Stenhammar for his work with our representation film and Lina Sundkvist Strindberg for the graphical design of our poster.

Without them this would not have been possible.

Axel Sundkvist & Joakim Hellberg
Stockholm, May, 2018
# Contents

1 Introduction 1
1.1 Background ................................................. 1
1.2 Purpose ...................................................... 2
1.3 Scope .......................................................... 2
1.4 Method .......................................................... 3
1.5 Similar Projects .............................................. 3

2 Theory ......................................................... 5
2.1 Microcontroller .............................................. 5
2.2 Sound and Sound Source Localization ...................... 5
2.3 DC Motor ...................................................... 7
2.4 Gear Transmission .......................................... 7
2.5 Relays .......................................................... 7
2.6 Sensors ........................................................ 8
2.7 Accelerometers .............................................. 8
2.8 Wireless Communication .................................... 9

3 Demonstrator .................................................. 11
3.1 Problem Formulation ....................................... 11
3.2 Overall Construction ....................................... 11
  3.2.1 Modular Construction .................................. 12
  3.2.2 Mechanics ................................................. 13
3.3 Hardware and Electronics .................................. 14
  3.3.1 Arduino ................................................... 14
  3.3.2 Sound Sensors ........................................... 14
  3.3.3 Accelerometer ............................................ 15
  3.3.4 Wireless Communication Module ....................... 16
  3.3.5 Display .................................................... 16
  3.3.6 DC Motor and Driver ................................... 17
  3.3.7 Power Supply ............................................ 17
3.4 Software ...................................................... 18
  3.4.1 Sound Source Localization Algorithm ................ 18
  3.4.2 Communication Algorithm .............................. 20
3.5 Verification and Testing .................................... 22
  3.5.1 Sound Source Localization Test ....................... 23
  3.5.2 Sound Source Distance Test ............................ 23
  3.5.3 Rotation System Test ................................... 23
## 4 Results

4.1 Sound Source Localization Test ........................................ 25
4.2 Sound Source Distance Test ........................................... 27
4.3 Rotation System Test .................................................. 27
4.4 Precision ......................................................... 28

## 5 Discussion and Conclusion  ........................................ 29

5.1 Discussion ..................................................................... 29
  5.1.1 Sound Source Localization Test ................................. 29
  5.1.2 Sound Source Distance Test ................................... 30
  5.1.3 Rotation System Test ............................................ 30
  5.1.4 Final Prototype and its Performance ....................... 31
5.2 Conclusion .............................................................. 32

## 6 Recommendations and Future Work  .......................... 33

Bibliography .................................................................. 35

Appendices .................................................................. 36

A Arduino Code ................................................................ 37
  A.1 Communication System ........................................... 37
  A.2 Sound Source Localization System ............................ 50

B Circuit Diagrams ..................................................... 57

C Test Data .................................................................. 59
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>PAL Original ball board [1]</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>Sound localization. The figure is drawn in Microsoft Paint.</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Schematic illustration of the function of relays. NC means normally closed, NO means normally open. Drawn in draw.io.</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>CAD pictures of final prototype, <em>Best PAL</em>, rendered in SolidWorks.</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Picture of communication device. This device contains an <em>Arduino</em> and all necessary electronics. Rendered in SolidWorks.</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Picture of communication device hanging on one of the PAL Original ball boards. Rendered in SolidWorks.</td>
<td>12</td>
</tr>
<tr>
<td>3.4</td>
<td>CAD picture of sound localization and rotation system. The sound sensors are placed in the top three corners just under the black spheres. Rendered in SolidWorks.</td>
<td>13</td>
</tr>
<tr>
<td>3.5</td>
<td>Sketch of bearings. The axle in the middle is standing still while the outer pipe is connected to the motor and rotates with the PAL Original ball board. Made in SolidWorks.</td>
<td>13</td>
</tr>
<tr>
<td>3.6</td>
<td><em>Arduino Uno</em> [12].</td>
<td>14</td>
</tr>
<tr>
<td>3.7</td>
<td>Sound sensor module [13].</td>
<td>14</td>
</tr>
<tr>
<td>3.8</td>
<td>Oscilloscope measurements of sound sensors detecting clapping and sending either a high or a low digital signal, depending on how the potentiometer is set. One unit corresponds to 500 microseconds.</td>
<td>15</td>
</tr>
<tr>
<td>3.9</td>
<td>Illustration of sound sensor placement. Drawn in Microsoft PowerPoint, Computer Aided Design (CAD) pictures rendered in SolidWorks.</td>
<td>15</td>
</tr>
<tr>
<td>3.10</td>
<td><em>MEMSIC 2125</em> accelerometer [14].</td>
<td>16</td>
</tr>
<tr>
<td>3.11</td>
<td><em>nRF24L01+</em> wireless communication module [15].</td>
<td>16</td>
</tr>
<tr>
<td>3.12</td>
<td><em>Parallax 2x16 Serial LCD</em> display for user interfacing [16].</td>
<td>16</td>
</tr>
<tr>
<td>3.13</td>
<td>Sketch of motor driver circuit. NC means normally closed and NO means normally open. Drawn in draw.io.</td>
<td>17</td>
</tr>
<tr>
<td>3.14</td>
<td>Flowchart of sound source localization algorithm. Drawn in draw.io.</td>
<td>19</td>
</tr>
<tr>
<td>3.15</td>
<td>Communication pattern. The display on <em>Best PAL 0</em> is not illustrated in this figure. Drawn in Microsoft PowerPoint, CAD pictures rendered in SolidWorks.</td>
<td>20</td>
</tr>
<tr>
<td>3.16</td>
<td>Flowchart of <em>Best PAL 0</em> software algorithm. Drawn in draw.io.</td>
<td>21</td>
</tr>
<tr>
<td>3.17</td>
<td>Flowchart of software algorithms of <em>Best PAL 1</em> and <em>Best PAL 2</em>, notice the similarity. Drawn in draw.io.</td>
<td>22</td>
</tr>
<tr>
<td>3.18</td>
<td>Illustration of sound source localization test. 10 sound peaks were generated at each angle for all three sides of the ball board. Drawn in Microsoft Powerpoint.</td>
<td>23</td>
</tr>
</tbody>
</table>
4.1 Plot of measured and estimated angles for AB-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in Matrix Laboratory (MATLAB). 25

4.2 Plot of measured and estimated angles for BC-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in MATLAB. 26

4.3 Plot of measured and estimated angles for CA-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in MATLAB. 26

4.4 Illustration of sound source distance test. The red circles to the left represent the sound sensor of the Best PAL. At each distance it is presented how many of the 10 hand claps that were detected. Drawn in Microsoft PowerPoint. 27

5.1 Oscilloscope measurements of a sound peak detected by a sound sensor. Note that this sensor is being used the inverted way compared to in this project, that is sending a digital high signal when a sound peak is registered. One unit corresponds to 10 microseconds. 29

5.2 Oscilloscope measurements of sound sensors detecting human voice and clapping. One unit on the horizontal axis corresponds to 500 microseconds. 30

B.1 Circuit diagram for the wireless communication system Best PAL 0, Best PAL 1 and Best PAL 2. For Best PAL 0 the display is connected to pin number 3 on the Arduino. Drawn in Fritzing. 57

B.2 Circuit diagram for the sound localization and rotating system. Drawn in Fritzing. 58
List of Tables

4.1 Requested and actual angle in the linear model in rotation system test. Table created in LATEX. .................................................. 27

4.2 Requested and actual angle after third degree adjustment in rotation system test. Table created in LATEX. .................................................. 27

C.1 Measured and estimated angle between sound sensors A and B. Table created in LATEX. .................................................. 59

C.2 Measured and estimated angle between sound sensors B and C. Table created in LATEX. .................................................. 60

C.3 Measured and estimated angle between sound sensors C and A. Table created in LATEX. .................................................. 60
List of Abbreviations

CAD  Computer Aided Design  
CPU  Central Processing Unit  
DC  Direct Current  
I/O  Input/Output  
KTH  KTH Royal Institute of Technology  
LED  Light Emitting Diode  
MATLAB  Matrix Laboratory  
PAL  a Swedish company, Practise and Learn  
PWM  Pulse Width Modulation  
TDOA  Time Difference of Arrival

Nomenclature

\( d \) the distance between sound sensors  
\( \Delta t \) time difference  
\( I \) electrical current  
\( M \) torque  
\( n \) number of revolutions  
\( \omega \) angular velocity  
\( t \) rotation time  
\( \theta \) angle  
\( U \) voltage  
\( u \) gear ratio  
\( v \) speed of sound (343 m/s)
Chapter 1

Introduction

This project studies the technology of localizing a sound source using simple electronics and a microcontroller. The main focus of the project is the sound localization, but wireless communication between several microcontrollers, and ball hit detection is also included.

1.1 Background

The original idea came from former professional football player Jessica Landström. She wanted to use electronics to build a "smart" ball board for football players to use when they practise their passing control and vision. Jessica cooperates with a Swedish company, Practise and Learn (PAL), who offers practise equipment for football players. PAL produces a ball board consisting of three wooden boards, placed in a triangle, see Figure 1.1. The PAL Original is a ball board consisting of three 600x600 mm wooden boards placed in a triangle, connected to each other by three metal rods, on the football field. The PAL Original weighs 16 kg.

![PAL Original ball board](image1.jpg)

Figure 1.1: PAL Original ball board [1].

During practise the football player can use several of these ball boards as teammates when they try to pass the ball to them. Jessica’s idea is to use electronics to be able to
light up one ball board at a time and then detect when the player has passed the ball to the corresponding ball board. The player then has to use his or her vision to see the next lit up ball board and pass the ball as fast as possible.

We wanted to elaborate on Jessica’s idea, and try to involve mechatronics, more specifically sound source detection. In many different sports, and especially football, there are three key parts to becoming a good player - that is technique, vision and communication. By including sound source detecting technology the player can use his or her voice to make the ball board turn towards him or her. This means that the returning pass will always come straight back to the player.

By using mechatronics in football practise we allow the player to practise passing control, vision, and communication all on their own.

1.2 Purpose

The purpose of this study is to find an effective way of detecting and localizing a sound source from 360 degrees. The main focus is to answer the following questions:

- How accurately can a sound source be localized using a microcontroller?
- At what distance can a human voice or clapping be distinguished from the background noise outside?
- How fast and accurately can the ball boards be rotated using an electrical motor and transmission?

The target with this project is to create a prototype consisting of a sound source localization system, and three wireless communication units. The sound source localization system should be able to calculate the angular position of a sound source providing a sound level above a certain threshold, and then rotate the construction to make one side of the PAL Original ball board face the sound source. The communication prototype units will furthermore be able to communicate with each other and light a Light Emitting Diode (LED) when it expects a ball to hit it. It will also be able to detect a ball hit.

1.3 Scope

This project will focus on sound source localization, motor controlled rotation, ball hit detection and wireless communication. The sound source localization is about estimating the angle from a certain plane to a sound source, the motor controlled rotation is about rotating a rig a certain angle in two directions, and the wireless communication is about how information can be communicated between several microcontrollers.

This project will not cover any voice or sound recognition, nor distance measuring.
1.4 Method

In this project there are a few distinctively separate steps. The first step is to localize a sound source, specifically a sound from a person, and then rotate the ball board. This step involves the mechatronic part of this project, therefore the main focus. The second step is to detect a ball that is passed to and reflected off of the ball board. The last step is to combine a number of these devices, to be able to light an LED when the player should pass to the corresponding ball board and then communicate with the other devices. The goal is to combine these steps into a fully working practise tool that can be used by footballers at all ages and levels of skill.

1.5 Similar Projects

The past few years students at KTH Royal Institute of Technology (KTH) have done similar projects for their bachelor thesis. One of them is a sound following robot, a robot that tracks and follows a sound emitting controller [2]. Another project uses sound source localization to find a sound source and shoots a projectile towards the sound source [3]. These projects involve sound detection in different surroundings and purposes.
Chapter 2

Theory

2.1 Microcontroller

The main components of a microcontroller is a Central Processing Unit (CPU), a memory, and Input/Output (I/O) ports. By establishing the correct connections between these three main components and some common electrical components such as resistors and capacitors, a microcontroller can be constructed [4]. A microcontroller can be programmed to execute the desired program if the memory is large enough, the CPU is fast enough, and there are as many I/O ports as the program needs.

The performance limit of a microcontroller is primarily set by the aforementioned three main components. The microcontroller is regularly connected to lamps, sensors and motors and the program on the microcontroller regulates these as requested. A microcontroller can process two kinds of signals, analog and digital. The analog signal is a continuous signal in time, measured as different voltages in a certain interval at a certain sampling rate. The digital signal is either high, 1, or low, 0, and will only exist at definite moments in time.

2.2 Sound and Sound Source Localization

Sound is mechanical waves transferring through a transition medium, if there is no medium (vacuum) sound will not exist. The sound waves cause momentary pressure deviations in the medium. The sound propagates in a spherical pattern from a sound source if the medium is homogeneous. The speed of sound depends on the medium and its temperature [5]. By knowing the speed, the sound can be used to calculate the angle to a sound source with two sound sensors. This is called acoustic localization, see Figure 2.1.
CHAPTER 2. THEORY

Figure 2.1: Sound localization. The figure is drawn in Microsoft Paint.

Via the Time Difference of Arrival (TDOA) method, where the time between sound detection in two sound sensors is measured [6]. The angle to the sound source is calculated according to equation 2.4 or 2.5 depending on whether the calculated angle is positive respectively negative,

$$\Delta t = \frac{\sqrt{(x_a - x)^2 + (y_a - y)^2} - \sqrt{(x_b - x)^2 + (y_b - y)^2}}{v}$$ \hspace{1cm} (2.1)

$$x = \pm \sqrt{\frac{4y_y y_b - y^2 - y_b^2 + \Delta t^2 v^2}{4}}$$ \hspace{1cm} (2.2)

$$y = \pm \sqrt{\frac{4x_x x_b - x^2 - x_b^2 + \Delta t^2 v^2}{4}}$$ \hspace{1cm} (2.3)

$$\theta = \frac{\pi}{2} - \arctan \frac{y}{x} \quad \forall \arctan \frac{y}{x} \geq 0$$ \hspace{1cm} (2.4)

$$\theta = -\frac{\pi}{2} - \arctan \frac{y}{x} \quad \forall \arctan \frac{y}{x} < 0$$ \hspace{1cm} (2.5)

Where the symbols indicate time difference ($\Delta t$), speed of sound ($343$ m/s) ($v$), angle ($\theta$) and ($x, y$) is ($x_{sound}, y_{sound}$) in figure 2.1, the rest of the variables follows the declarations in figure 2.1.

With the knowledge of which sound sensor detected the sound peak first it is possible to determine which of the two sound sensor is closest, and therefore the sign of the angle. The measured TDOA then makes it possible to calculate the angle to the sound source as

$$\theta = |\arcsin \frac{\Delta tv}{d}|$$ \hspace{1cm} (2.6)
2.3 DC Motor

Where the distance between sound sensors (d) is needed. Equation 2.6 is an approximation of equation 2.1, 2.2, 2.3, 2.4 and 2.5. With the use of three sound sensors this method can be extended to localize a sound source from 360 degrees.

2.3 DC Motor

A Direct Current (DC) motor is a motor powered by direct current electricity. The motor has a stator, the stationary part, and a rotor, the rotating part. There are different types of DC motors, but often the stator consists of permanent magnets and the rotor consists of a coil where the electrical current is supplied. Magnetic forces between the stator and the rotor gets the rotor spinning.

The speed and torque of a DC motor relates to the incoming voltage and current as presented in equation 2.7 and 2.8,

\[ n \propto U \]  
\[ M \propto I \]  

(2.7)  

(2.8)

Where the symbols indicate number of revolutions (n), voltage (U), torque (M) and electrical current (I) [7]. This allows control of the DC motor by simply changing the incoming current and voltage. There are other ways to control a DC motor, for example with Pulse Width Modulation (PWM) via an H-bridge, but it is not necessary in this project.

2.4 Gear Transmission

To achieve greater torque or greater speed a gear transmission can be used. To achieve greater torque with a gear transmission it is at expense of the speed, which is defined by the gear ratio, equation 2.9 and 2.10,

\[ u = \frac{\omega_{in}}{\omega_{out}} = \frac{M_{out}}{M_{in}} \]  
\[ \omega_{in}M_{in} = \omega_{out}M_{out} \]  

(2.9)  

(2.10)

Where the symbols indicate gear ratio (u), angular velocity (\( \omega \)) and torque (M). In and out is refers to the input and output gear [8].

2.5 Relays

Electrical relays are often used when a larger current or voltage needs to be controlled by and galvanically separated from a smaller control current or voltage. A transistor is an
example of a semiconductor relay which is capable of changing its state extremely fast. An electromagnetic relay consists of a switch with a magnet which is being controlled by an electromagnetic coil. This means that the switch can be opened or closed by changing the current in the coil, see Figure 2.2a and 2.2b. The electromagnetic relays have to wait for the physical switch to open and close, which means it is not as fast as a transistor [7].

![Unaffected relay.](image1) ![Affected relay.](image2)

Figure 2.2: Schematic illustration of the function of relays. NC means normally closed, NO means normally open. Drawn in draw.io.

### 2.6 Sensors

The main task of a sensor is to measure the value of a signal corresponding to a specific physical quantity. A digital sensor detects changes and communicates when the change of the specified quantity in its environment is below or above a certain level. An analog sensor detects the physical quantity and provides a continuous analog signal [9]. A digital sensor on the other hand can only send either a high or low (1 or 0) signal.

There are numerous different sensors developed to measure separate matters. A frequently used digital sensor is the fuel tank level sensor which measures the fuel level in the tank of a car and communicates by turning on a lamp when the level is too low. Another sensor is a limit switch which is a simple example of a digital sensor that closes a circuit when for example the machine reaches one of its limits. An analog sensor can via the use of electronics be converted to a digital sensor, which analyzes the analog signal and sends a digital high or low signal when the analog signal is above or below a certain threshold respectively.

### 2.7 Accelerometers

Accelerometers are sensors constructed to convert changes in velocity (acceleration) into stress deviations or deflections. To read the value from an accelerometer stress deviation or deflection is turned into an analog or digital electrical signal containing information of the acceleration [11].

Depending on how often the acceleration is measured it is possible to detect different kinds of acceleration. If the acceleration is measured more hundred times a second it is possible to detect impulses as well as uniform acceleration.
2.8  Wireless Communication

Wireless communication is the transfer of data over a certain range without connection with electrical conductors between two or more communicating sources, most often computers. The range of wireless communication varies from just a few meters to thousands of kilometres.

The communication can be achieved in different ways depending on how far the data has to travel and how fast the data has to be transferred between the communicating sources. One of the ways to communicate between sources is via electromagnetic radio waves. The frequencies which the radio waves travels with varies between 3 kHz and 300 GHz and they travel with the speed of light. A difficulty with radio technology is to receive and transmit the correct messages, since radio is a common use of wireless communication it is easy to read disturbances from other sources than desired [10].

Serial communication means that a message is sent with a carrier wave (3 kHz to 300 GHz) and the message consists of a start bit, a data word and a stop bit. The start bit tells the receiver to start reading the data word, which contains certain information, and the stop bit tells the receiver that the data word is over. Depending on the size of the information package the data can be transferred with different speed. How long one bit lasts in the message determines how much information can be sent during a certain time interval. It is therefore crucial for both the transmitter and the receiver to know which data transfer frequency is being used [7].
Chapter 3

Demonstrator

3.1 Problem Formulation

In order to build the final prototype there are a few different areas that need to be covered. Firstly a sound source needs to be localized, secondly the whole construction needs to be rotated towards the sound source and lastly there has to be some kind of communication between separate devices.

3.2 Overall Construction

The final prototype is a rotating ball board, which we call Best PAL (Ball Exercise Sound Tracking PAL), driven by a DC motor. The construction consists of a steel foot welded to an axle. The axle is connected to a frame via roller bearings and the PAL Original is easily mounted to the construction, see Figure 3.1a and 3.1b.

![Image](a) Without PAL Original mounted.

![Image](b) With PAL Original mounted.

Figure 3.1: CAD pictures of final prototype, Best PAL, rendered in SolidWorks.

The final prototype also consists of several communication devices, see Figure 3.2. The
CHAPTER 3. DEMONSTRATOR

communication devices can communicate with eachother and detect ball hit. They can also be connected to a DC motor driver and three sound sensors in order to enable sound source localization and rotation. The whole construction was manufactured by hand in the school workshop and at home.

Figure 3.2: Picture of communication device. This device contains an Arduino and all necessary electronics. Rendered in SolidWorks.

3.2.1 Modular Construction

Since the construction has two main focuses it was built in two separate systems, as stated earlier. One focusing on the communication between different devices and another focusing on sound source localization and rotation. Both the communication and the localization units can use the same Arduino. The communication unit can be hung on one of the PAL Original ball boards, see Figure 3.3. The communication devices are built in three different implementations, Best PAL 0, Best PAL 1 and Best PAL 2, with almost identical code, to be able to communicate with each other, further explanation in section 3.4.2.

Figure 3.3: Picture of communication device hanging on one of the PAL Original ball boards. Rendered in SolidWorks.

The main components of the communication system are an Arduino, an accelerometer, a radio transceiver and a LED. The main components of the sound localization and rotation device are an Arduino, a DC motor, 12 V batteries, a worm gear and three sound sensors, see Figure 3.4. The two systems can be combined since the program code is written to simply connect the parts to different I/Os on the Arduino.
3.2. OVERALL CONSTRUCTION

Figure 3.4: CAD picture of sound localization and rotation system. The sound sensors are placed in the top three corners just under the black spheres. Rendered in SolidWorks.

3.2.2 Mechanics

The bearing between the axle and the frame consists of a radial ball bearing and a tapered roller bearing, see Figure 3.5. The motor, battery, and microcontroller rotates together with the ball board, which means they can be connected during the rotation without wires being twisted. This also means that it doesn’t need to have room for motor and battery under the rotation rig, allowing the construction to become low as possible.

Figure 3.5: Sketch of bearings. The axle in the middle is standing still while the outer pipe is connected to the motor and rotates with the PAL Original ball board. Made in SolidWorks.

The DC motor is connected to a worm gear which in turn is connected to the axle via a pipe and two breakpins. A breakpin is a small pin which is used to transfer a rotational motion from an axle to a hub. The breakpin is placed in a drilled hole through the axle and the hub, forcing them to rotate together. The breakpin solution is simple and easily assembled, but it is also very imprecise. This means that there is a play of approximately ±7 degrees.

The construction was made with the fundamental idea to make it easily assembled, repaired and modified. We also made the construction to be compatible with the PAL Original.
3.3 Hardware and Electronics

In this section all hardware and electronics are presented.

3.3.1 Arduino

The microcontroller used in this project is an Arduino UNO which is a single-board microcontroller based on the 8-bit microprocessor ATmega328P, see Figure 3.6. In this project up to 12 I/Os are being used at the same time.

![Arduino Uno](image)

Figure 3.6: Arduino Uno [12].

3.3.2 Sound Sensors

To detect sound and estimate the location of the sound source, Arduino compatible sound sensor modules are used, see Figure 3.7. The sound sensor modules consists of a condenser microphone, which must be powered with 5V, and an amplifier. The sound sensors can detect sound from approximately a hemisphere. The placement of the sound sensors and the fact that the ball board is low (600 mm high) makes it possible to detect a human voice or hand clapping if the person is standing up.

![Sound sensor module](image)

Figure 3.7: Sound sensor module [13].

The sound sensors are digital, with an adjustable potentiometer which makes it possible to adjust at what sound level the signal will be interpreted as high and low respectively. The potentiometer also allows the user to change whether a sound peak will result in either a high or a low digital signal, this is represented in Figures 3.8a and 3.8b.
3.3. HARDWARE AND ELECTRONICS

In this project the alternative in Figure 3.8b was used, which means that the microcontroller awaits a low edge from the sound sensor signal. The placement of the sound sensors is illustrated in Figure 3.9. The placement of the sound sensors makes it possible to localize a sound source from 360 degrees, further explanation in section 3.4.1.

3.3.3 Accelerometer

To detect when a ball hits the construction Arduino compatible accelerometer modules are used. The module used in this project is MEMSIC 2125, see Figure 3.10. The accelerometer is placed in the middle of the communication devices, which makes it possible to detect a ball hit.
This module is a dual axis accelerometer capable of measuring acceleration, tilt, vibration and rotation. The modules consist of six pins, one to measure the differentiation on the x-axis and one on the y-axis, two ground pins and one power pin. The modules can be powered with 3.3 V to 5 V DC. The MEMSIC 2125 module also has a function to measure temperature but that is not relevant in this project.

### 3.3.4 Wireless Communication Module

To send information via wireless communication nRF24L01+ Arduino compatible modules are used, see Figure 3.11. The wireless communication modules can be set at different power amplifier levels and data rates to achieve a satisfactory and reliable wireless communication in a certain environment. This module communicates with a radio frequency of 2.4 GHz.

![nRF24L01+ wireless communication module](image)

**Figure 3.11: nRF24L01+ wireless communication module [15].**

### 3.3.5 Display

For one of the communication devices Best PAL 0 a display is used to communicate with the user. The display that is used is a backlit Parallax 2x16 Serial LCD and it is being controlled via an arduino compatible driver, see Figure 3.12.

![Parallax 2x16 Serial LCD display](image)

**Figure 3.12: Parallax 2x16 Serial LCD display for user interfacing [16].**
3.3. HARDWARE AND ELECTRONICS

### 3.3.6 DC Motor and Driver

The rotation system is powered by a 24 V DC motor and a worm gear. The motor is a 24 V, 0.64 Nm, 2800 rpm DC motor from *Exmek Electric*. The worm gear has a gear ratio of 1:30 and is self-locking, which means that it takes 30 motor revolutions for the worm gear to complete one rotation. The motor is in turn powered by a 12 V motorcycle battery and is being controlled via two relays. The driver circuit allows control in both directions, see Figure 3.13a, 3.13b and 3.13c. A 12 V battery is used because the DC motor would be too fast and strong if it was working with its full capacity.

![Motor Driver Circuit](image)

**Figure 3.13:** Sketch of motor driver circuit. NC means normally closed and NO means normally open. Drawn in draw.io.

This driver circuit allows simple motor control in both directions, with no risk of short circuit. Soft start and PWM controlling is not necessary for this application and the speed of the motor can be controlled by changing the voltage over the motor by series circuit with some sort of resistor, in this case a car lamp is used to reduce the voltage over the motor. The relays are both cheap and easily controlled from the *Arduino*.

To rotate a certain angle the rotation first needs to be calibrated. The calibration means that we run the motor in one direction for a certain time, and then measure the rotated angle. With this information it is possible to create a linear model for the rotation, see equation 3.1,

\[
t_{\text{commanded}} = \frac{t_{\text{calibration}}}{\theta_{\text{calibration}}} \cdot \theta_{\text{commanded}}
\]

where the symbols indicate rotation time \( t \) and rotation angle \( \theta \). Commanded means the wanted angle and the time that the motor will be run, and calibration means the calibration time and the measured calibration angle. This linear model can be adjusted to match the reality better. In section 5.1.3 we introduce a third degree adjustment which increases the precision.

### 3.3.7 Power Supply

The 24V DC motor is powered by a 12V motorcycle battery. The speed of the motor rotation, which is proportional to the input voltage, is being reduced by series connection...
CHAPTER 3. DEMONSTRATOR

of the motor and a regular car lamp. The car lamp also works as a current limiter. The Arduino on the other hand is powered by a 9V battery pack.

3.4 Software

The software algorithms are presented here. The software is made as similar as possible for each device type and the same Arduino pins are used for the same electronic devices as far as possible. All the Arduino code is presented in appendix A.

In this paper the sound source localization system is separated from the communication system, but they can easily be combined so that a communication system also includes the sound source localization. Here they are separated to make it easier to explain and understand.

3.4.1 Sound Source Localization Algorithm

The sound source localization algorithm is built up by a few cases. The first case awaits signals from the three sound sensors detecting a sound peak. The second case is initiated when two sound sensors have detected a sound peak. At each of these two detections the time in microseconds is being measured. Depending on which two of the three sound sensors detected a sound peak there are three possible cases to enter, in each of these three cases we neglect the sound sensor that didn’t detect a sound peak. Depending on which sound sensors detected a sound peak and depending on which of them detected it first the TDOA is calculated and the desired rotation direction is decided. The TDOA is then used to calculate the angle to the sound source. Flow chart of the algorithm is showed in Figure 3.14.
The placement of the sound sensors allow us to only focus on the two sound sensors that detect a sound peak first, and the maximum needed rotation of the construction is 60 degrees (worst case scenario means that the triangle points towards the sound source). If
the time measured exceeds the maximum TDOA the system jumps back to the first state and awaits a new signal. The maximum TDOA is the time it takes for the sound to travel the shortest way between two sound sensors. The reason for the maximum TDOA being exceeded is that the distance between the sound sensors also means that the amplitude of the sound will be different for each sound sensor. This means that if one sound sensor detects a sound peak it doesn’t necessarily mean that another one also will.

3.4.2 Communication Algorithm

The communication algorithm is built up from two cases, either the unit’s LED is on and the unit awaits ball hit, or the LED is off and the unit awaits a radio signal. The communication pattern is a circular loop where every unit receives signal from one unit and sends signal to another. This means there is no need for a master unit, and it simplifies the programming. The radio message is an information package consisting of two integers - the address of the information (where to) and the present score. The communication pattern is illustrated in Figure 3.15.

A unit which is awaiting a ball hit reads the signal from its accelerometer and when a large enough deviation is detected the Arduino interprets it as a ball hit. When a ball hit is detected the Arduino generates a random number to decide which unit should expect a ball hit next and calculates the score. A higher score is achieved if the player always passes the ball to the ball board expecting a pass as fast as possible. The algorithms of Best PAL 0, Best PAL 1 and Best PAL 2 are illustrated in Figure 3.16, 3.17a and 3.17b.
3.4. SOFTWARE

The *Best PAL 0* unit has a display to present the score to the player, it also tells when the game starts and ends. Therefore is *Best PAL 0* always starting in the case where it awaits ball hit, unlike *Best PAL 1* and *Best PAL 2* which start in the case where they await
a radio message.

![Flowchart of software algorithms of Best PAL 1 and Best PAL 2](image)

Figure 3.17: Flowchart of software algorithms of Best PAL 1 and Best PAL 2, notice the similarity. Drawn in draw.io.

Note that the only difference between the algorithms of Best PAL 1 and Best PAL 2 is their respective addresses, and who they send and receive data from. The algorithm of Best PAL 0 is identical to the Best PAL 1 and Best PAL 2 with the only difference of starting in another case, and also different addresses, according to Figure 3.15.

### 3.5 Verification and Testing

In this section the performed tests will be presented. Three tests were executed in order to answer the research questions.
3.5. VERIFICATION AND TESTING

3.5.1 Sound Source Localization Test

To answer the first research question “How accurately can a sound source be localized using a microcontroller?” a sequence of sound peaks were measured at different angles. The test is illustrated in Figure 3.18. At each angle (-60, -45, -30, 0, 15, 30, 45, 60) ten sound peaks were generated for each of the three sides of the triangular ball board. This means that 360 degrees of sound source localization was covered in the test. To generate distinct and consistent sound peaks drumsticks were used as a sound source.

![Figure 3.18: Illustration of sound source localization test. 10 sound peaks were generated at each angle for all three sides of the ball board. Drawn in Microsoft Powerpoint.](image)

3.5.2 Sound Source Distance Test

To answer the second research question “At what distance can a human voice or clapping be distinguished from the background noise outside?” the following test was performed. Hand claps were used as a sound source to test the sound sensors’ distance capacity. The distance was discretely increased with 1 m from a start point of 2 m from the sound sensors. At each distance 10 separate hand claps were executed and every time at least two of the sound sensors detected a sound peak it was noted. This was repeated up until a distance where no clap were detected. During the test the ball board had one side pointing towards the sound source. Note that hand claps were used in this test, unlike the sound source localization test. The sound sensors had complications to detect human voice good enough to complete this test, so hand claps were used instead.

3.5.3 Rotation System Test

To answer the third research question “How fast and accurately can the ball boards be rotated using an electrical motor and transmission?” firstly a linear rotation model was implemented. The linear model was then examined by rotating the construction 15, 30 and 60 degrees multiple times in positive and negative direction. The rotation time was then
evaluated in the same way by using a third degree polynomial instead of the linear method. The two methods were then compared.
Chapter 4

Results

4.1 Sound Source Localization Test

The results of the sound source localization test are presented in Figure 4.1, 4.2 and 4.3. The test data is also presented in Table C.1, C.2 and C.3 in Appendix C. In this case (60 cm, 343 m/s) the maximum TDOA is 1750 microseconds. The sound sensors are called A, B and C and they are mounted in alphabetical order clockwise on the Best PAL.

Figure 4.1: Plot of measured and estimated angles for AB-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in MATLAB.
CHAPTER 4. RESULTS

Figure 4.2: Plot of measured and estimated angles for BC-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in MATLAB.

Figure 4.3: Plot of measured and estimated angles for CA-side of the ball board. The sound sensors are marked with A, B and C and the measured angles are marked with red crosses. Plotted in MATLAB.
4.2 Sound Source Distance Test

The results of the sound source distance test are illustrated in Figure 4.4. The test was performed outside during day time. At each marked distance 10 claps were performed according to the description earlier. Human voice was not detected at any distance.

![Illustration of sound source distance test. The red circles to the left represent the sound sensor of the Best PAL. At each distance it is presented how many of the 10 hand claps that were detected. Drawn in Microsoft PowerPoint.](image)

4.3 Rotation System Test

The result of the rotation system is presented in Table 4.1 and 4.2. During this test the angle was measured by rotating the rig as far as possible in the same direction of the built in play, to ensure that the actual rotated angle was measured.

<table>
<thead>
<tr>
<th>Requested angle x times rotated</th>
<th>60</th>
<th>30 x 2</th>
<th>15 x 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured angle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>50</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>47</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>47</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>48</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>48</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>60.0</td>
<td>48.0</td>
<td>48.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requested angle x times rotated</th>
<th>60</th>
<th>30 x 2</th>
<th>15 x 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured angle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>58</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>57</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>56</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>57</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>57</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>60.0</td>
<td>57.0</td>
<td>54.8</td>
</tr>
</tbody>
</table>
4.4 Precision

The precision of the angular measurements is estimated to be approximately ± 5 degrees. This means that it is less than the play of ± 7 degrees of the rotation rig.
Chapter 5

Discussion and Conclusion

5.1 Discussion

5.1.1 Sound Source Localization Test

The sound source localization test showed some interesting results. It is possible to localize a sound source using a microcontroller and simple sound sensors. The precision and consistency of the test is quite bad, but the mean value is not too far away from the desired value.

The fact that the sound source localization algorithm only cares about the first sound peak for each sound sensor should mean that the problem with sound reflections is not crucial. Since sound reflections have lower intensity than the original sound wave and since they also will have to travel a longer distance they should in theory not be the first detected sound peak, and therefore not influence the estimated angle. Another interesting phenomenon is represented in Figure 5.1.

![Figure 5.1: Oscilloscope measurements of a sound peak detected by a sound sensor. Note that this sensor is being used the inverted way compared to in this project, that is sending a digital high signal when a sound peak is registered. One unit corresponds to 10 microseconds.](image)

It is clear that the sound sensor has some initial noise for about 7 microseconds in which the *Arduino* has trouble deciding whether the signal is high or low. This means that we might interpret the signal as high sometime during this initial phase or when the signal has settled as high, this will obviously affect the calculated angle.
CHAPTER 5. DISCUSSION AND CONCLUSION

Another limitation is the sample rate of the measurements. A faster microprocessor and a faster program will result in a better resolution in the measured time difference, and therefore also a more accurately estimated angle. The calculation method is an approximation of the correct method and this will also generate an error.

5.1.2 Sound Source Distance Test

The sound source distance test was performed to measure the sound sensors capability of detecting a sound source moving further and further away. In the application of the product developed in this project it would be more desirable if one could use their voice instead of clapping, since it is meant to be used by football players when they practice. The problem with this is that the sound sensors don’t respond to human voice as well as they respond to clapping. This is illustrated in Figure 5.2a and 5.2b. In the oscilloscope measurements it is made clear that the sound sensors are more sensitive to clapping than to human voice.

![Oscilloscope measurements of sound sensors detecting human voice and clapping.](image)

(a) Human voice.

(b) Clapping.

Figure 5.2: Oscilloscope measurements of sound sensors detecting human voice and clapping. One unit on the horizontal axis corresponds to 500 microseconds.

5.1.3 Rotation System Test

Since the linear model of the rotation time was not satisfying a third degree polynomial was estimated. This removed some of the deviations between the requested rotation and the actual rotation,

\[ y = \frac{-1}{6912} \cdot x^3 + \frac{1}{192} \cdot x^2 + \frac{29}{24} \cdot x \]  \hspace{1cm} (5.1)
5.1. DISCUSSION

Where \( x \) is the proportionally calculated rotation time and \( y \) is the calculated rotation time after third degree adjustments. Equation 5.1 is a third degree polynomial which compensates the rotation time for the three measured angles (15, 30, 60) in the test. For a more accurate model more angles could be measured and a higher degree polynomial could be used to compensate.

One problem with the linear model is the start up and stopping phase of the motor. It takes a while for the motor to accelerate to the final speed and decelerate to full stop. This means that a running time that is far from the calibration time will result in a larger error, since there is a larger or smaller percentage start up and stopping time than for the calibration. This can be compensated by the use of some other model, for example a third degree polynomial. The more calibration points and the higher degree polynomial will result in higher precision.

After the implementation of the third degree polynomial the limiting factor of precision in the rotation is the built in play because of the breakpin construction in the linkage between the worm gear and the rotating rig. In the application in which the Best PAL will be used it is not necessary to have a much better precision.

The test was executed with a fully charged 12 V battery and the DC motor was supplied with 4.8 V because of the voltage reduction due to the car lamp in series circuit with the motor. To start the rotation the current needed was 7 A and as the rotation progressed it lowered to 4 A. This gave a Rotation speed of 0.8 seconds to rotate 60 degrees and 0.5 seconds to rotate 30 degrees. The voltage and current to the motor was measured to get an idea of how much power is being used - approximately 336 W in the start up phase and 192 W during constant speed.

When the battery is being used, the voltage level drops. This means that the rotation speed also will decrease, meaning that the calibrated angle/rotation time will have to be updated. The effect of this is not significant since the motor is turned off the majority of time during the use of Best PAL as a practise tool.

5.1.4 Final Prototype and its Performance

The final Best PAL prototype consists of a sound source localization system and a ball hit detecting radio communicating system.

The sound source localization system can localize a clapping sound from up to 6 m and rotate the ball board towards it. The maximum needed rotation is 60 degrees, since the PAL Original is triangular, and the rotation takes about 0.8 seconds. The precision of the sound source localization is better than the built in play of the rotation rig. The use of a self-locking worm gear is good for construction robustness and because it reduces the speed and increases the torque of the rotation. Since the rig is used for football practise there will be a lot of ball impacts on the rig. The self-locking worm gear prevents the outside force to damage the DC motor.

The communicating devices, Best PAL 0, Best PAL 1 and Best PAL 2, can be placed on the PAL Original ball board. It can detect ball hit with the accelerometer and communicate with three devices. When the device that expects a pass detects a ball hit it generates a random number which tells which of the devices should expect a pass next. This system has
some communication troubles when the devices are placed hanging over three separate PAL Original ball boards. This might be due to the PAL Original wooden boards interfering the radio signal. It might be fixed with the use of better antennas.

5.2 Conclusion

By using the described components the concluded answers to the research questions for this project are as following:

- At a distance of 2 meters a sound source can be localized with an accuracy of approximately $\pm 5$ degrees. Where the accuracy of the localization increases as the angle to the sound source reduces.

- The sound sensors used in this project can not distinguish human voices. The sensors can however distinguish hand clapping at a distance of 6 meters. The sensors will distinguish approximately every other clap at a distance of 6 meters.

- The ball boards can rotate consistently at an angular velocity of 75 degrees per second and the accuracy of the rotation is $\pm 6$ degrees.

The results of the testing and the performance of the final prototype including both the sound source localization and the communication devices demonstrate that the concept works. The wireless communication devices can communicate with each other, detect ball hits with the accelerometers and calculate a score. The sound source localization system can detect a sound source from 360 degrees and up to 6 m and rotate the ball board with satisfactory accuracy and consistency for the application.
Chapter 6

Recommendations and Future Work

The sound source localization part of this project could benefit from better and more suitable sound sensors. One large problem at the moment is the difficulties of detecting a sound peak consistently, especially from a human voice. The use of some sort of analog microphone would allow continuous calibration and filtering to get rid of background noise and other disruptions. A better suited microphone would also be able to pick up the interesting sound frequencies, to detect human voice. If the sound source localization is used as a practise tool for football players it is important that it can localize a human voice, since that is what football players use to communicate.

Another possible improvement is to use another calculation method for calculating the angle to the sound source. One calculation method is presented in a paper from Information Science and Technology University in Beijing and it makes use of TDOA for all three sound sensors [17]. This method uses more data power and is more difficult to implement, but its advantage of using more input information (all three sound sensors) means it will give a better result.

The rotating rig has potential of development with a better connection between the motor and the standstill axle. Since there is a lot of play in the connection at the moment it is impossible to achieve high precision in the rotation.

Power supply to the DC motor can also be improved so that it is more suitable for the application, and doesn’t need a voltage reduction and current limiting lamp in series circuit. The motor could also be changed to a smaller, more suitable size. The use of PWM motor controlling in order to be able to easily change the speed of the rotation and implement some sort of soft start and soft stop could also be examined. An encoder could be used if it would be desirable with feedback controlling of the rotation. A simple encoder with low resolution could also be used for the initial calibration of the rotation rig.

The wireless communication could be improved with better hardware, such as an antenna. At the moment the communication is a bit unreliable and for this product to be used as a practise tool it has to be more reliable.

Another future work would be to increase the number of devices in the wireless communication. Since the communication pattern is circular every device only needs two closest
communicating devices, regardless how many devices is included in the whole system. This means it can easily be expanded to many more devices.

The *Best PAL* communication devices would need a better attachment or hanging interface to the PAL Original. It is important that the product is easily mounted and that it doesn’t break when the ball hits the ball board. A smartphone application which could save the score would improve the product as well. It would mean that football players could practise on their own and compete and have fun with their friends on the same time.
Bibliography


Appendix A

Arduino Code

Here the Arduino code is presented. Corresponding connection diagrams is presented in appendix B.

A.1 Communication System

Listing A.1: Best PAL 0 Arduino code

```c
/*
by: Joakim Hellberg and Axel Sundkvist
Date: 2018-04-27
CMAST15
Bachelor’s degree mechatronics – MF133X

Best PAL module type 0 – radio communication, ball hit detection and display.

Connect microphones: A−2, B−3, C−4 (clockwise)
Connect LED: 5
Accelerometer: 6
RF device: 7, 8, 11, 12, 13
Display: 2
Buzzer: 3
*/

// libraries for radio communication and display
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <SoftwareSerial.h>

// define pin numbers
#define TxPin 2
#define buzz_sound 3
#define led 5
#define xPin 6
```
#define radioCE 7
#define radioCSN 8

// display
SoftwareSerial mySerial = SoftwareSerial(255, TxPin);

// switch-case variable
int state;

// variables to read the pulse widths:
int pulseX;
int pulseXx;
int acc_sense;

// For score
unsigned long time_recieved;
unsigned long time_hit;
unsigned long game_time;
int points;

// For where to send the information package
int randint;

// radio addresses
RF24 radio(radioCE, radioCSN); // CE, CSN
byte addresses[][6] = {"Nod1","Nod2","Nod3"};

// data package datatype
struct sendinfo
{
    int where_to; // Where to send
    int score; // Sum of points
};

// instance of data package datatype
sendinfo SENDINFO;

void setup() {

    // starting state - state 1 for BestPAL0
    state = 1;

    game_time = 60000; // set the game time in milliseconds

    // set pin modes
    pinMode(led, OUTPUT);
    pinMode(xPin, INPUT);
    pinMode(buzz_sound, OUTPUT);
    digitalWrite(led, HIGH);
A.1. COMMUNICATION SYSTEM

// radio communication
radio.begin();
radio.setPALevel(RF24_PA_MAX); // Power Amplification level
radio.setDataRate(RF24_1MBPS); // Data frequency
radio.setChannel(124);
radio.openWritingPipe(addresses[0]); // Write to BestPAL1
radio.openReadingPipe(1, addresses[2]); // Read from BestPAL2

// Random number generation
randomSeed(analogRead(0)); // If it is important for a
// sequence of values generated
// by random() to differ, on subsequent
// executions of a sketch, use randomSeed()
// to initialize the random number generator
// with a fairly random input, such as analogRead()
// on an unconnected pin.

--- display ---
mySerial.begin(9600);
delay(100);
mySerial.write(12); // Clear
mySerial.write(17); // Turn backlight on
delay(5); // Required delay
mySerial.println("Hello there!"); // First line
mySerial.write(13); // Form feed
mySerial.println("I'm your BESTPAL!"); // Second line
mySerial.write(212); // Quarter note
delay(3000); // Wait 3 seconds
mySerial.write(12); // Clear
mySerial.write(17); // Turn backlight on

// buzzer sound — game starts!
digitalWrite(buzz_sound, HIGH);
delay(500);
digitalWrite(buzz_sound, LOW);

// set score to 0
SENDINFO.score = 0;

// start score clock
time_received = millis();

// initial value from accelerometer
pulseX = pulseIn(xPin, HIGH); // read accelerometer
pulseXx = pulseX;
acc_sense = 300; // accelerometer sensitivity

}
void loop() {
  switch (state) {
    case 1:
      // awaiting ball hit
      pulseX = pulseIn(xPin, HIGH);  // read the accelerometer
      if (abs(pulseX - pulseXx) > acc_sense)  // ball hit detected
        {
          time_hit = millis();  // remember when ball hit was detected
          randint = random(9);  // generate random number to decide address
          if (randint == 0) {
            // back to me
            SENDINFO.where_to = 0;
          } else if (randint < 5) {
            // send to BestPAL1
            SENDINFO.where_to = 1;
          } else {
            // send to BestPAL2
            SENDINFO.where_to = 2;
          }
      // calculate the score
      SENDINFO.score = SENDINFO.score + 100 - (time_hit - time_received)/200;
      delay(5);
      if (!radio.write(&SENDINFO, sizeof(sendinfo))) {
        // information not sent
        mySerial.write(12);  // Clear
        mySerial.write(17);  // Turn backlight on
        delay(5);  // Required delay
        mySerial.print("Send fail...");  // First line
        mySerial.write(13);  // Form feed
        mySerial.print("-");  // Second line
        mySerial.write(212);  // Quarter note
        mySerial.write(220);  // A tone
        delay(3000);  // Wait 3 seconds
        mySerial.write(12);  // Clear
        mySerial.write(17);  // Turn backlight on
      } else {
        // information successfully sent!
        digitalWrite(led, LOW);
      }
  }
}
A.1. COMMUNICATION SYSTEM

```c
state = 2;
} // go to state 2

} // awaiting radio communication
radio.startListening(); // start listening
delay(50); // necessary delay
while (!radio.available()); // waiting to receive!
radio.read(&SENDINFO, sizeof(sendinfo)); // read radio message
if (SENDINFO.where_to == 0)
{
    // message was for me
    if (millis() > game_time)
    {
        // if game time is exceeded, game over! Go to state 3.
        state = 3;
    }
    else
    {
        // if message is to me, go to state 1
        digitalWrite(led, HIGH); // light the LED
        time_received = millis(); // remembered when message was received, for score
        state = 1; // go to state 1
        pulseX = pulseIn(xPin, HIGH); // read accelerometer
        pulseXx = pulseX;
        radio.stopListening(); // stop listening for radio
delay(5); // necessary delay
        mySerial.write(12); // Clear
        mySerial.write(17); // Turn backlight on
delay(5); // Required delay
        mySerial.print("Your score ... "); // First line
        mySerial.write(13); // Form feed
        mySerial.print(SENDINFO.score); // Second line
        mySerial.write(212); // Quarter note
        mySerial.write(220); // A tone
    }
}
else
{
    // if message is to someone else, send the information further
    radio.stopListening(); // stop listening to radio
delay(5); // necessary delay

    if (!radio.write(&SENDINFO, sizeof(sendinfo)))
    {
        // if radio message not successfully sent, go to state 6
    }
}
```

APPENDIX A. ARDUINO CODE

```cpp
    state = 6;
}
break;
case 3:
    // game over, buzz three times and print the score on the display
    digitalWrite(buzz_sound, HIGH);
    delay(500);
    digitalWrite(buzz_sound, LOW);
    delay(500);
    digitalWrite(buzz_sound, HIGH);
    delay(500);
    digitalWrite(buzz_sound, LOW);
    delay(500);
    digitalWrite(buzz_sound, HIGH);
    delay(500);
    digitalWrite(buzz_sound, LOW);
    delay(500);
    digitalWrite(buzz_sound, HIGH);
    delay(500);
    digitalWrite(buzz_sound, LOW);
    // print score on display
    delay(100);
    mySerial.write(12);  // Clear
    mySerial.write(17);  // Turn backlight on
    delay(5);           // Required delay
    mySerial.print("Well done! Score:");  // First line
    mySerial.write(13);  // Form feed
    mySerial.print(SENDINFO.score);      // Second line
    mySerial.write(212);         // Quarter note
    mySerial.write(220);         // A tone

    while(true)
    {
        // infinite loop to wait for reset
    }
break;
case 6:
    // radio communication failure, print on display
    mySerial.write(12);  // Clear
    mySerial.write(17);  // Turn backlight on
    delay(5);           // Required delay
    mySerial.print("Send fail ...");  // First line
    mySerial.write(13);  // Form feed
    mySerial.print("RESET!");      // Second line
    mySerial.write(212);         // Quarter note
    mySerial.write(220);         // A tone

    while(true);  // infinite loop to wait for reset
    break;
}
```
Listing A.2: Best PAL 1 Arduino code

/*
by: Joakim Hellberg and Axel Sundkvist
Date: 2018–04–27
CMAST15
Bachelor’s degree mechatronics – MF133X

Best PAL module type 1 – radio communication and ball hit detection.

Connect microphones: A–2, B–3, C–4 (clockwise)
Connect LED: 5
Accelerometer: 6
RF device: 7, 8, 11, 12, 13
*/

// libraries for radio communication
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// define pin numbers
#define led 5
#define xPin 6
#define radioCE 7
#define radioCSN 8

// switch-case variable
int state;

// variables to read the pulse widths
int pulseX;
int pulseXx;
int acc_sense;

// For score
unsigned long time_received;
unsigned long time_hit;

// For where to send the information package
int randint;

// radio addresses
RF24 radio(radioCE, radioCSN); // CE, CSN
byte addresses[] = {"Nod1", "Nod2", "Nod3"};

// data package datatype
struct sendinfo
{
  int where_to; // Where to send
APPENDIX A. ARDUINO CODE

```cpp
int score; // Sum of points

// instance of data package datatype
sendinfo SENDINFO;

void setup() {
    // initial state - await radio communication
    state = 2;

    // set pin modes
    pinMode(led, OUTPUT);
    pinMode(xPin, INPUT);

    // radio communication
    radio.begin();
    radio.setPALevel(RF24_PA_MAX); // Power Amplification level
    radio.setDataRate(RF24_1MBPS); // Data frequency
    radio.setChannel(124);
    radio.openWritingPipe(addresses[1]); // writing to BestPAL2
    radio.openReadingPipe(1, addresses[0]); // reading from BestPAL0

    // random number generation
    randomSeed(analogRead(0)); // If it is important for a
    // sequence of values generated
    // by random() to differ, on subsequent
    // executions of a sketch, use randomSeed()
    // to initialize the random number generator
    // with a fairly random input, such as analogRead()
    // on an unconnected pin.

    // startup signal with LED
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay(500);
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay(500);
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay(500);
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);

    // initial value from accelerometer
    pulseX = pulseIn(xPin, HIGH); // read accelerometer
    pulseXx = pulseX;
    acc_sense = 300; // accelerometer sensitivity
}
```

44
A.1. COMMUNICATION SYSTEM

```c
void loop() {

switch (state) {

case 1:
    // awaiting ball hit
    pulseX = pulseIn(xPin, HIGH); // read the accelerometer
    if (abs(pulseX−pulseXx) > acc_sense) // ball hit detected
    {
        time_hit = millis(); // remember when ball hit was detected
        randint = random(9); // generate random number to decide address
        if (randint == 0) {
            // back to me
            SENDINFO.where_to = 1;
        }
        else if (randint < 5) {
            // address = BestPAL0
            SENDINFO.where_to = 0;
        }
        else {
            // address = BestPAL2
            SENDINFO.where_to = 2;
        }
        // calculate the score
        SENDINFO.score = SENDINFO.score+100−(time_hit−time_recieved)/200;
        digitalWrite(led, LOW); // turn LED off
        state = 2; // go to state 2
        if (!radio.write(&SENDINFO, sizeof(sendinfo))) {
            // if radio communication failed go to state 3
            state = 3;
        }
    }
    break;

case 2:
    // awaiting radio communication
    radio.startListening(); // start listening
    delay(50); // necessary delay
    while (!radio.available()); // waiting to receive!
    radio.read(&SENDINFO, sizeof(sendinfo)); // read radio message
    if (SENDINFO.where_to == 1)
```
APPENDIX A. ARDUINO CODE

Listing A.3: Best PAL 2 Arduino code

/*
 * by: Joakim Hellberg and Axel Sundkvist
 * Date: 2018-04-27
 * CMAST15
 * Bachelor’s degree mechatronics – MF133X

Best PAL module type 2 – radio communication and ball hit detection.

Connect microphones: A–2, B–3, C–4 (clockwise)
A.1. COMMUNICATION SYSTEM

Connect LED: 5
Accelerometer: 6
RF device: 7, 8, 11, 12, 13

/*
// libraries for radio communication
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// define pin numbers
#define led 5
#define xPin 6
#define radioCE 7
#define radioCSN 8

// switch-case variable
int state;

// variables to read the pulse widths
int pulseX;
int pulseXx;
int acc_sense;

// For score
unsigned long time_recieved;
unsigned long time_hit;

// For where to send the information package
int randint;

// radio addresses
RF24 radio(radioCE, radioCSN); // CE, CSN
byte addresses[][6] = {"Nod1", "Nod2", "Nod3"};

// data package datatype
struct sendinfo
{
    int where_to; // Where to send
    int score; // Sum of points
};

// instance of data package datatype
sendinfo SENDINFO;

void setup() {
    // initial state – await radio communication
    state = 2;

    // set pin modes
APPENDIX A. ARDUINO CODE

```c
pinMode(led, OUTPUT);
pinMode(xPin, INPUT);

// radio communication
radio.begin();
radio.setPALevel(RF24_PA_MAX);   // Power Amplification level
radio.setDataRate(RF24_1MBPS);   // Data frequency
radio.setChannel(124);
radio.openWritingPipe(addresses[2]); // writing to BestPAL0
radio.openReadingPipe(1, addresses[1]); // reading from BestPAL1

// random number generation
randomSeed(analogRead(0)); // If it is important for a
// sequence of values generated
// by random() to differ, on subsequent
// executions of a sketch, use randomSeed()
// to initialize the random number generator
// with a fairly random input, such as analogRead()
// on an unconnected pin.

// startup signal with LED
digitalWrite(led, HIGH);
delay(500);
digitalWrite(led, LOW);
delay(500);
digitalWrite(led, HIGH);
delay(500);
digitalWrite(led, LOW);
delay(500);
digitalWrite(led, HIGH);
delay(500);
digitalWrite(led, LOW);

// initial value from accelerometer
pulseX = pulseIn(xPin, HIGH);  // read accelerometer
pulseXx = pulseX;
acc_sense = 300;               // accelerometer sensitivity

void loop() {

switch (state) {

case 1:
    // awaiting ball hit
    pulseX = pulseIn(xPin, HIGH);  // read the accelerometer
    if (abs(pulseX−pulseXx) > acc_sense)
    // ball hit detected
    {
```
A.1. COMMUNICATION SYSTEM

time_hit = millis(); // remember when ball hit was detected
randint = random(9); // generate random number to decide address
if (randint == 0)
{
    // back to me
    SENDINFO.where_to = 2;
}
else if (randint < 5)
{
    // address = BestPAL0
    SENDINFO.where_to = 0;
}
else
{
    // address = BestPAL1
    SENDINFO.where_to = 1;
}
// calculate the score
SENDINFO.score = SENDINFO.score + 100 - (time_hit - time_recieved) / 200;
digitalWrite(led, LOW); // turn LED off
state = 2; // go to state 2
if (!radio.write(&SENDINFO, sizeof(sendinfo)))
{
    // if radio communication failed go to state 3
    state = 3;
}
break;
case 2:
    // awaiting radio communication
radio.startListening(); // start listening
delay(50); // necessary delay
while (!radio.available()); // waiting to receive!
radio.read(&SENDINFO, sizeof(sendinfo)); // read radio message
if (SENDINFO.where_to == 2)
{
    // if message is to me, go to state 1
    digitalWrite(led, HIGH); // turn LED on
time_recieved = millis(); // remember when message was received, for score
radio.stopListening(); // stop listening for radio
delay(5); // necessary delay
state = 1; // go to state 1
pulseX = pulseIn(xPin, HIGH); // read accelerometer
pulseXx = pulseX; // reset accelerometer values
}
else
{
    // if message is to someone else, send the information further
    radio.stopListening();  // stop listening to radio
    delay(5);               // necessary delay

    if (!radio.write(&SENDINFO, sizeof(sendinfo)))
    {
        // if communication failed, go to state 3
        state = 3;
    }
}
break;
case 3:
    // communication failed
    while (true)
    {
        // blinking to tell the user that communication failed
        digitalWrite(led, HIGH);
        delay(300);
        digitalWrite(led, LOW);
        delay(300);
    }
break;
}

A.2 Sound Source Localization System

Listing A.4: Sound source localization Arduino code

/*
by: Joakim Hellberg and Axel Sundkvist
Date: 2018-04-27
CMAST15
Bachelor’s degree mechatronics – MF133X

Best PAL with sound source detection.

Connect microphones: A-2, B-3, C-4 (clockwise)
Relays: 9,10
*/

// math library to calculate angle to sound source
#include <math.h>

// define pin numbers
#define relay1 9
#define relay2 10
A.2. SOUND SOURCE LOCALIZATION SYSTEM

// switch-case variable
int state;

// command angle
String command_angle_str; // for motor testing from computer
int command_angle;

// sound source detection variables
bool Rot(int Dir, double angle); // motor rotation function
int max_time;
int stateloc;
float t0A;
float t0B;
float t0C;
float Dt;
long angle;
int Dir;
float calibration_time;
float calibration_angle;
float distance;
float velocity;

void setup() {
  // set pin 2,3,4 as inputs
  DDRD = B11100011;

  // set relays as outputs
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);

  // initialize variables
  stateloc = 0;
t0A = 0;
t0B = 0;
t0C = 0;
  calibration_time = 1.0; // chosen value
  calibration_angle = 60.0; // measured from chosen calibration time
  max_time = 1750; // exclude measured time differences above this
  distance = 0.60; // distance between microphones
  velocity = 343.0; // speed of sound
}

void loop() {
  switch (stateloc) {
  
  
  
}
APPENDIX A. ARDUINO CODE

case 0:
// await sound peak
if ((PIND & B00000100) == 0 && t0A == 0)
{
    // signal from sound sensor A
    t0A = micros();
}
if ((PIND & B00001000) == 0 && t0B == 0)
{
    // signal from sound sensor B
    t0B = micros();
}
if ((PIND & B00010000) == 0 && t0C == 0)
{
    // signal from sound sensor C
    t0C = micros();
}
if (t0A*t0B != 0 || t0A*t0C != 0 || t0B*t0C != 0)
{
    // two sound sensors detected sound, go to next state
    stateloc = 1;
}
break;

case 1:
// choose next state depending on which sound sensor
// did not get a signal
if (t0A == 0)
{
    stateloc = 2;    // exclude A
}
if (t0B == 0)
{
    stateloc = 3;    // exclude B
}
if (t0C == 0)
{
    stateloc = 4;    // exclude C
}
if (t0A*t0B*t0C != 0)
{
    // if all sound sensor detected a sound peak
    // the calculations will not work
    stateloc = 0;
t0A = 0;
t0B = 0;
t0C = 0;
}
break;

52
case 2:
    // calculate time difference and decide rotation direction
    // sound sensor A did not detect sound peak
    Dt = t0B - t0C;
    if (abs(Dt) > max_time)
        {
            // maximum time exceeded
            stateloc = 0;
            t0A = 0;
            t0B = 0;
            t0C = 0;
        }
    else if (Dt > 0)
        {
            Dir = 1;
            stateloc = 5;
        }
    else
        {
            Dir = -1;
            stateloc = 5;
        }
    break;

case 3:
    // calculate time difference and decide rotation direction
    // sound sensor B did not detect sound peak
    Dt = t0A - t0C;
    if (abs(Dt) > max_time)
        {
            // maximum time exceeded
            stateloc = 0;
            t0A = 0;
            t0B = 0;
            t0C = 0;
        }
    else if (Dt > 0)
        {
            Dir = -1;
            stateloc = 5;
        }
    else
        {
            Dir = 1;
            stateloc = 5;
        }
    break;

case 4:
    // calculate time difference and decide rotation direction
    // sound sensor C did not detect sound peak
APPENDIX A. ARDUINO CODE

Dt = t0A - t0B;
if (abs(Dt) > max_time)
{
    // maximum time exceeded
    state_loc = 0;
    t0A = 0;
    t0B = 0;
    t0C = 0;
}
else if (Dt > 0)
{
    Dir = 1;
    state_loc = 5;
}
else
{
    Dir = -1;
    state_loc = 5;
}
break;
case 5:
// calculate angle to sound source and call rotation function

// calculation method for calculating angle to sound source
angle = asin(Dt/1000000*velocity/distance)*180/PI;
Rot(Dir, angle); // call rotation function

// Reset the time variables
 t0A = 0;
 t0B = 0;
 t0C = 0;
//state = 2;
break;
}

// function for rotation
// takes direction (-1 or 1) and desired rotation angle
// relay1 high --> positive rotation
// relay2 high --> negative rotation
bool Rot(int Dir, double angle){
    float rotTime;

    // calculate rotation time from calibration
    rotTime = calibration_time/calibration_angle*abs(angle);
    rotTime = rotTime*1000; // convert to milliseconds
if (Dir == 1)
{
    // positive direction
    digitalWrite(relay1, HIGH);
    delay(rotTime);
    digitalWrite(relay1, LOW);
    digitalWrite(relay2, LOW);
}
else if (Dir == -1)
{
    // negative direction
    digitalWrite(relay2, HIGH);
    delay(rotTime);
    digitalWrite(relay1, LOW);
    digitalWrite(relay2, LOW);
}
else
{
    // direction error – do nothing
}
return true;
Appendix B

Circuit Diagrams

Figure B.1: Circuit diagram for the wireless communication system Best PAL 0, Best PAL 1 and Best PAL 2. For Best PAL 0 the display is connected to pin number 3 on the Arduino. Drawn in Fritzing.
Figure B.2: Circuit diagram for the sound localization and rotating system. Drawn in Fritzing.
Appendix C

Test Data

Test data for sound source localization test is presented in tables C.1, C.2 and C.3 below.

<table>
<thead>
<tr>
<th>Actual angle</th>
<th>-60</th>
<th>-45</th>
<th>-30</th>
<th>-15</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>-57</td>
<td>-53</td>
<td>-43</td>
<td>-34</td>
<td>-6</td>
<td>0</td>
<td>13</td>
<td>28</td>
<td>41</td>
<td>57</td>
</tr>
<tr>
<td>-55</td>
<td>-55</td>
<td>-43</td>
<td>-34</td>
<td>-6</td>
<td>0</td>
<td>13</td>
<td>28</td>
<td>41</td>
<td>57</td>
</tr>
<tr>
<td>-55</td>
<td>-55</td>
<td>-43</td>
<td>-34</td>
<td>-6</td>
<td>0</td>
<td>13</td>
<td>28</td>
<td>41</td>
<td>57</td>
</tr>
</tbody>
</table>

Table C.1: Measured and estimated angle between sound sensors A and B. Table created in \LaTeX.
APPENDIX C. TEST DATA

Table C.2: Measured and estimated angle between sound sensors B and C. Table created in \LaTeX.  

<table>
<thead>
<tr>
<th>Actual angle</th>
<th>-60</th>
<th>-45</th>
<th>-30</th>
<th>-15</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Measured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>-54.0</td>
<td>-47.4</td>
<td>-34.4</td>
<td>-19.9</td>
<td>-3.9</td>
<td>12.5</td>
<td>25.6</td>
<td>38.2</td>
<td>53.8</td>
</tr>
</tbody>
</table>

Table C.3: Measured and estimated angle between sound sensors C and A. Table created in \LaTeX.  

<table>
<thead>
<tr>
<th>Actual angle</th>
<th>-60</th>
<th>-45</th>
<th>-30</th>
<th>-15</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Measured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>-50.8</td>
<td>-35.1</td>
<td>-20.5</td>
<td>-11.4</td>
<td>1.0</td>
<td>18.3</td>
<td>35.4</td>
<td>43.1</td>
<td>57.0</td>
</tr>
</tbody>
</table>