Development of a Small-Scale, Educational, Embedded, Audio Radar System

Presented by:

Dewan Pieterse

PTRDEW001

Department of Electrical and Electronics Engineering

University of Cape Town

Prepared for:

Dr. Mohammed Yunus Abdul Gaffar

Department of Electrical and Electronics Engineering

University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering

October 14, 2019

Key words: Education, Radar, Audio, Embedded Systems, Low-Cost

Terms of Reference

Title

Development of a small-scale, educational, low-cost embedded audio radar system for demonstration purposes.

Description

Radar systems are generally a concept that is not accessible to the masses without prior knowledge about the workings of radar. The high frequencies that radar usually work at are not audible and therefore, not tangible for the layperson. In an attempt to bridge the gap between the inexperienced and the university student with basic knowledge of how radar operates, the low-cost audio radar system would be developed. The radar system operated in the audible range to produce a tangible result.

Deliverables

The items that are expected upon the completion of the project include:

1. a working audio radar system to be used at university open-days.
2. a manual on how to build and program the radar to be easily reproducible by university students.
3. a manual on how to operate the completed radar system.
4. a full report that outlines the design and development of the system as a whole and all subsystems.

Requirements and Skills

Interfacing between software and hardware (electronics), Programming, Radar Theory.

Area

Education, Radar Systems, Embedded Systems and Technology

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

. . . .
D. Pieterse

14 October 2019
. . . .
Date

Acknowledgements

The success and undertaking of this project would not be possible without an instrumental role and support by many people.

I would like to sincerely thank Dr M.Y. Abdul Gaffar for the countless hours of supervision, encouragement and being a source of knowledge and ideas that played such a significant role in the process of conducting this project. Without your guidance, the project would not have been a success.

To my co-supervisor, Mr J. Son, thank you for taking in an additional student and for aiding the project by offering advice while being a soundboard for ideas.

Thank you to Mr J. Pead for the guidance on electronics selection and practical questions that I had. You provided me with the necessary tools to succeed. Thank you to Chiao-Shing 'Charlie' Lin for your quick email responses and ideas that helped shape the project into what it is today. Also thank you to Denielle Rawthee and Riselle Rawthee for doing work over your holiday period to aid my decision making during project.

Without the support and help of my colleagues, the project would have gone a different direction. I would like to thank each friend that offered a listening ear when something inevitably went wrong.

To my family and friends, thank you for the confidence that you displayed in my abilities. My parents for making this all possible, Matthew who helped and supported me whenever I needed help and ideas. Thank you to Mia who stood by and supported me throughout the project, my studies and life.

Abstract

To invoke interest in students at open days concerning radar systems, a short-range radar system which uses audio would be used as an entry point.

Short-Range, Embedded, Audio Radar is an accessible and tangible radar system that can be used to spark the interest of school and university students alike. The small and portable form factor makes the project appealing for open days because it can easily be demonstrated. The radar system intends to be as affordable as possible using breakout boards that are readily available to increase the uptake of interest in radar systems as a whole.

The report aims to develop an audio radar system using only off-the-shelf components to drive the cost down and to also be accessible to interested persons around the world. The project is completely open-source, using only open-source software in its implementation, breakout boards from local and international manufacturers and operated under GNU General Public License.

The radar has two methods in its implementation. The Continuous Wave radar, which plays out a constant tone of frequency set by the user, is used to measure radial velocity to- and from the radar. A spectrogram of the recorded sound is used to analyse the signal and display the relative velocity between a moving object and the radar over time. This method relies on the Doppler Effect.

The radar also implements a Pulsed-Doppler radar. The system plays out multiple pulses consisting of a 'chirp' each. A chirp can be described as a linear increase in frequency throughout the pulse where the amount of frequency increase is the bandwidth. Matched filtering is used to match up the transmitted and the received signals. The output from the Pulsed-Doppler radar shows the velocity and the range of an object at a specific point in time.

The continuous wave radar was found to be better suited towards objects moving at higher velocities. Pulsed-Doppler radar is limited to very low velocities but the range of the object is known more precisely. The output from both of the radar methods is only displayed to the user without any analysis of the data. The output figure, however, does offer to show the user a clear depiction of what happened during the sound output from the radar.

Finally, this report shows the successful implementation of a small-scale, low-cost, educational, audio radar system using only easily accessible parts and microcontrollers. The radar is easily reproducible anywhere in the world with comparable results.

Abbreviations and Initialisms

ADC	Analogue to Digital Converter
CPU	Central Processing Unit
CW	Continuous Wave
DAC	Digital to Analogue Converter
dBsm	Decibels square meter
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
HAT	Hardware Attached on Top
ISM	Industrial, Scientific and Medical frequency band
MDF	Medium Density Fibreboard
MIT	Massachusetts Institute of Technology
PCB	Printed Circuit Board
PRI	Pulse Repetition Interval
PRF	Pulse Repetition Frequency
Radar	Radio Detection and Ranging
RAM	Random Access Memory
RF	Radio Frequency
SAR	Synthetic Aperture Radar
SNR	Signal to Noise Ratio
SSH	Secure Shell
THD	Total Harmonic Distortion
UART	Universal Asynchronous Receiver/Transmitter
UCT	University of Cape Town
USB	Universal Serial Bus

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This report details the research, design, implementation and testing of a Low-Cost Embedded Audio Radar System. It was made as a part of a final year undergraduate project.

1.1 Background to the Study

Radar Systems, where it stands for *RAdio Detection And Ranging*, historically, have been used only as part of military operations or where the military and civilian paths cross such as airports [10]. Radar is also used to observe the weather. Since radar operates using radio waves and usually over a long distance, high frequencies are used.

Radar has been an area of research over the last century, but only at Masters Degree level. The first mention of radar systems is only at the third-year level in most universities where Signals & Systems courses are taught. Some attempts have been made to make radar more accessible such as the uRad Arduino and Raspberry Pi systems [11], but they still operate at very high frequencies and a vast amount of background should be known before understanding the system.

No other system exists to fill the gap and start students off at high school and university level and get them interested in the major research field that is Radar Systems.

1.2 Objectives of the Study

The objective of the project is to produce a simple, easy to use and informative radar system to be as accessible as possible. The system should be reproducible by interested parties with limited experience in radar and embedded systems.

1.2.1 Problems to be Investigated

Difficulties that need to be addressed in the project include:

- The principles of radar systems need to be understood by investigation and research.

- Investigate the different components of the radar system and choose appropriate electronics.
- Plan and design the algorithms for audio signal processing for radar implementation.
- Design a user interface for the user to interact with the radar system.
- Design and build the audio radar system as an enclosed product.

1.2.2 Purpose of the Study

The importance of radar systems in everyday life are recognised in the project and can be linked to weather radar, airport surface and control space radar systems up to military defence radar systems. Getting students interested in Science, Technology, Engineering and Mathematics (STEM) can be a tough challenge as is, but the specialised field of radar systems poses an even greater challenge. To generate interest in an easy and accessible way, the audio radar can fill that gap and get even generate interest from the general public in the workings and possibilities of radar. Furthermore, the audio radar is built using off-the-shelf components and open-source software to show just how attainable and open radars can be.

1.3 Scope and Limitations

1.3.1 Scope

The scope of the project includes the design and implementation of an audio radar system comprising of electronic components and embedded systems easily accessible and low cost. The following is included in the scope:

- Investigate and implement a suitable embedded platform to generate, transmit, receive and process audio radar signals using the Python programming language.
- Building the interfaces between the transmit, receive and processing components of the system using a suitable communications protocol.
- Design and implement the user interface of the system that renders the product to be as user-friendly as reasonably possible.
- Design and build a portable enclosure to protect the system and assure repeated use in a portable manner.

- Report on the entire process that consists of the design, implementation and results of an embedded audio radar system.

1.3.2 Limitations

The limitations of the project include:

- The time limitation on the project is 12 weeks in total.
- A total budget for all of the components is R 1500.
- Analysis of obtained data. For the CW radar, only a spectrogram would be produced and for Pulsed-Doppler only a Range vs Doppler Frequency (velocity) plot would be produced.

1.4 User Requirements

The user requirements for this project have been derived from the scope of the project. The requirements can be seen in Table 1.1 below.

Reference ID	**Requirement**
UR1	Low-Cost implementation of an audio radar
UR2	Simple user interface for layperson to interact with system
UR3	Measure distance and radial velocity from the radar
UR4	Portable design allows for usage at open days

Table 1.1: User Requirements

1.4.1 User Requirements Analysis

Given the requirements in Table 1.1 above, the following functional requirements in Table 1.2 have been developed with the end user in mind.

1.4.2 Functional Requirements

The basic overview of the functional requirements can be seen in Figure 4.2 below.

Reference ID	Requirement	Derived From
FR1	Operate the radar within the audible spectrum	UR1
FR2	Use easily accessible and low-cost, off-the-shelf components	UR1
FR3	Portable design allows for usage at open days	UR4

Table 1.2: Functional Requirements

1.4.3 Technical Specifications

Reference ID	Requirement	Derived From
TS1	Operate 5000 Hz to 12000 Hz in the audible spectrum	FR1
TS2	Measure distance within 15 m and 5 m and velocity up to 40 km/h	FR2
TS3	Fit within an A4 paper box to be portable	FR3

Table 1.3: Technical Specifications

1.4.4 Constraints

Constraints that pose a threat to the project include:

- R1500 budget of all the components and materials used.
- The 12 week time limit to complete the project and report.
- To only use off-the-shelf components to render the project reproducible by anyone with the tutorial to experience a low cost radar.
- The radar needs to be small enough to carry around.

1.5 Plan of Development

The paper will start by addressing the detailed user and technical requirements that the audio radar must follow. The audio radar would be developed by first doing a comprehensive review of existing works and research and other similar products.

Following the review of the literature, the embedded audio radar will be designed. This includes the process of selecting different hardware components, as well as the embedded platforms for processing and capturing the signal data. It also includes the classical engineering process followed to test and validate each component to assure that the candidates yield the required final results.

Furthermore, the final choices of components are laid out and the final design and implementation of the audio radar will follow. The results obtained using the selected hardware will be shown whereafter it will be discussed in detail.

The report will conclude whether the audio radar system satisfies the initial user and technical requirements. Following the conclusion, recommendations will be made based on the discussion concerning the results.

1.6 Report Outline

The table below outlines the report structure following the sections as set out in the Plan of Development above.

Chapters	Project Stage	Description
2	Requirements	This chapter sets out the requirements of the audio radar.
3	Literature Review	The chapter reviews previous research conducted into radar systems regarding short range, audible frequencies or low-cost implementations.
4	Theory of Radar	This chapter covers all the relevant topics regarding radar that the audio radar utilises. The chapter excludes any theory that is not directly used by the audio radar.
5, 6	Design, Implementation	Subsystems are tested and decided on and the detailed design of the complete radar is laid out in these chapters.
7	Results, Discussion, Conclusion	Results of the implemented audio radar is shown. Discussion follows about whether the requirements were met and possible improvements and recommendations.

Table 1.4: Report Outline

Chapter 2

Literature Review

This literature review will focus on educational radar applications, systems developed by students and finally commercial products available on the market either satisfying the low-cost, educational or small scale criteria. Short-range radar systems are mostly the focus of this literary review.

2.1 Educational Radar Systems

Given that the purpose of the study is to investigate a small-scale, educational radar, existing offerings have to be considered for its strengths and drawbacks that it poses and how the project can improve and use existing technology.

2.1.1 MIT Radar

The Massachusetts Institute of Technology (MIT) developed a course where students build a small radar system that is capable of measuring range, Synthetic Aperture Radar Imaging (SAR) and velocity using the Doppler Frequency phenomenon. The purpose of the course is to develop a relatively inexpensive radar system that would get students motivated to become resilient when working through challenging courses [1].

The radar system also serves to introduce students to a wide variety of research fields including electromagnetics, digital signal processing (DSP), analogue circuit design, radio frequency (RF) design and radar systems as a whole. A completely built MIT S-Band radar can be seen in Figure 2.1.

The radar system is capable of measuring range, velocity and producing Synthetic Aperture Images. The project used the popular method of FMCW again because of the lower cost associated with the technology and the simple construction of the radar [1]. The radar operates in the industrial, scientific and medical (ISM) band with $2.4GHz$ being a common frequency. The transmitting power of the system is approximately $10mW$ with a maximum range of 1km for $10dBsm$.

The radar makes use of two coffee cans as antennae where they both transmit and receive. The waveform generator generates a low power signal and is then amplified

Figure 2.1: Completed MIT S-Band Radar
[1]

by the transmitter. The receiver passes the received signal through a filter where after the signal is amplified and fed into the analogue-to-digital (ADC) converter. After the signal is captured, the provided MATLAB code is used to process the data.

The MIT radar opened up possibilities to students that were not previously possible. The whole radar was developed with reproducibility in mind. It is part of MIT's Open Courseware and the schematics, plans and instructions are completely open-source. The radar is also relatively low cost considering that radars for sea-going vessels can reach upwards of $3000 [12]. The sea-going vessels offer an easily accessible and tangible example of a working radar system. The MIT radar is also an accessible option where students can engage with how radar works and the processing behind it. Results obtained by a team attending the course at MIT can be seen in Figure 2.2. The figure shows a Time vs Doppler plot with cars slowing down at a traffic light.

The radar system has a total cost of $359.96 [13] which makes the field of radar more accessible yet excludes the likes of high school students and university students without a formal course with funding. Apart from the cost involved with the radar, other drawbacks of the system are that it operates in the ISM band. This in itself is not a weakness but for an educational radar package, it would be beneficial to see or hear something tangible yield results. The physical size of the completed radar can also be viewed as a drawback since it is a cumbersome shape with two large coffee cans on its edge. The device is also prone to stop working with being dropped or knocked since RF components are rather

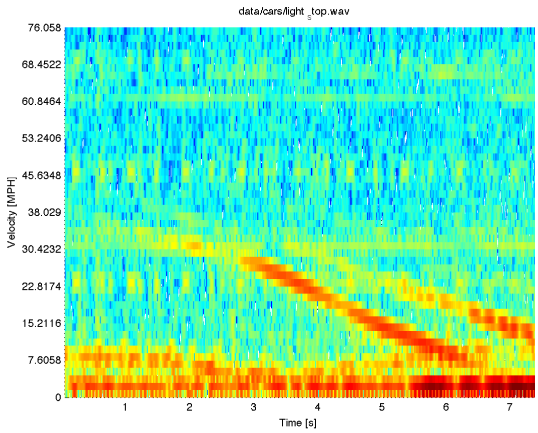

Figure 2.2: MIT Radar Velocity Measurement [1]

sensitive.

2.1.2 Other Educational Radar Programmes

Several universities around the world offer undergraduate and postgraduate courses introducing the workings of radar systems. One such course is offered by the Johns Hopkins University's Whiting School for Engineering. The course covers everything needed to get a solid fundamental understanding of radars including antennae, the range equation, matched filtering and pulse Doppler to name a few.

However, the course is offered at a prestigious university and attending a single course is not possible. The course also has prerequisites that prior knowledge of stochastic processes, MATLAB, DSP and electromagnetism making the course exclusive. There is also no practical aspect of the course [14].

Other universities do offer a single semester course in radar systems such as the University College London. The course is a four-day short-course focusing on the principles of modern radar systems and the signal processing associated with it. The course is for graduate-level engineers and a background in electronics and physics is required. It is possible to sign-up for this course as a standalone course or towards a masters level qualification.

The drawbacks of this course are once again that no practical and tangible results are obtained. The cost of the course is also expensive at £1500 for four days. This course is not aimed at a high school or university students to spark interest in the radar research field.

2.2 Student-Developed Radar Systems

The University of Cape Town (UCT) has a research centre specialising in Radar Systems. The centre drives research in the field especially in final year projects for engineers. One such engineer is Chiao-Shing Lin who completed his project in 2018. His project investigated the design and implementation of an audio radar for hand gesture recognition. He used an STM32F4 microcontroller to sample the audio between $8Khz$ and $12KHz$. All of the components needed to build the radar was developed from the ground up. The signal processing took place on a PC using MATLAB [15].

The project proved to be a success with results showing that using the audible range of frequencies are indeed possible to recognise hand gesture such as the *come-here* movement with a finger or *spraying-water* gesture. Figure 2.3 shows results where one hand moved toward the radar and another away from it.

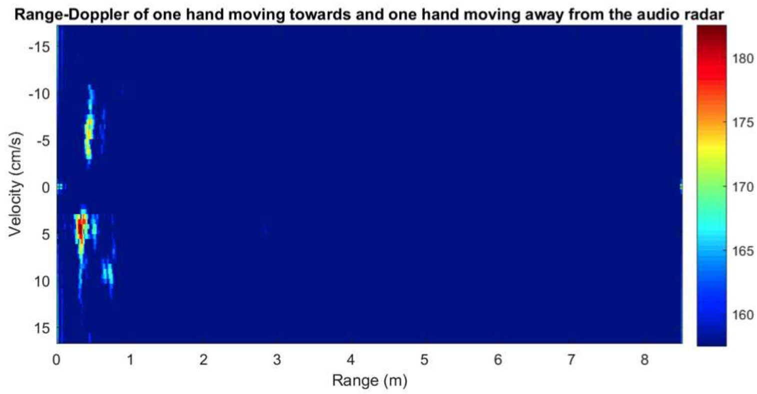

Figure 2.3: Audible Radar Results
[15]

The project proved to be stricken for time as all of the components were built from scratch. It causes issues as prior advanced knowledge of circuitry should have been known to build the radar. The use of a computer to process the data is also essential, making the project too large for open days. The reproducibility of the project is low since the circuitry can be difficult to understand and build.

A similar project at UCT for hand-gesture recognition was built by Rishad Ali Yasin. He focused on building a radar but in the ultrasonic sound spectrum. The radar was also built from the ground up but only using ultrasonic frequencies ($40KHz \pm 1KHz$) as the centre frequency. The signal processing also took place on a PC and the STM32F4 was also used in the project [16]. Figure 2.4 shows the physical implementation of the system.

The project yielded successful results in recognising gestures such as moving towards and away from the transmitter/receiver with a corner reflector. Results from a test where

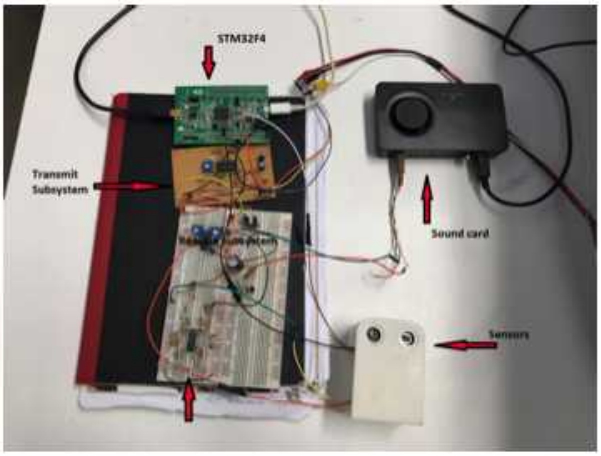

Figure 2.4: Ultrasonic Implementation of Radar
[16]

hands and fingers were moving can be seen in Figure 2.5.

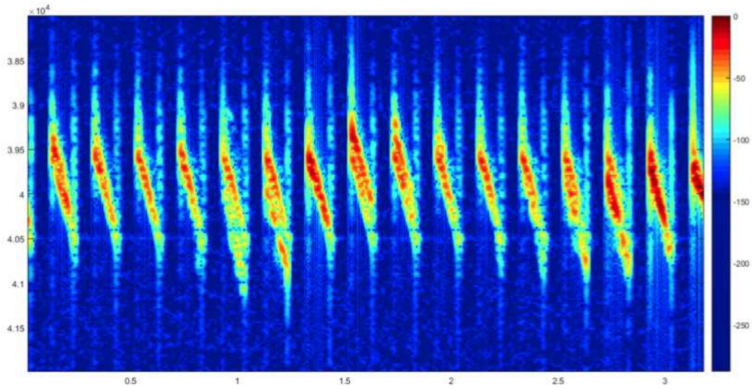

Figure 2.5: Ultrasonic Radar Results
[16]

Both of the above-mentioned projects used the STM32F4 microcontrollers as an embedded platform but the need for the PC running MATLAB was still required to process the signals. This is a drawback in the context of presenting the data as well as the portability of the system.

2.3 Commercial, Small-Scale Radar Systems

Following the student developed radar systems, the commercial sector has some small-scale radar systems on offer for use in education and industry.

2.3.1 uRad Radar System

uRad is a Spanish start-up which is specifically focused on development small-scale embedded radar systems. They have developed a system that can be used as a *Hardware-attached-on-top* (HAT) on a Raspberry Pi or as a module to attach to an Arduino Uno Revision 3. The option for a universal uRad radar also exists.

The system makes use of either FMCW in the form of a triangular wave as well as a sawtooth wave for modulating the transmitted signal. Both of these signals would be used to track the range of objects. The sawtooth modulated wave offers no velocity measurement but high accuracy in measurement of distance. The triangular modulated wave offers both velocity and distance measurement with high accuracy.

The uRad system also offers a Doppler CW mode that is used exclusively for radial velocity measurement and offers the best accuracy and very low complexity. The Arduino compatible uRad board can be seen in Figure 2.6 [11].

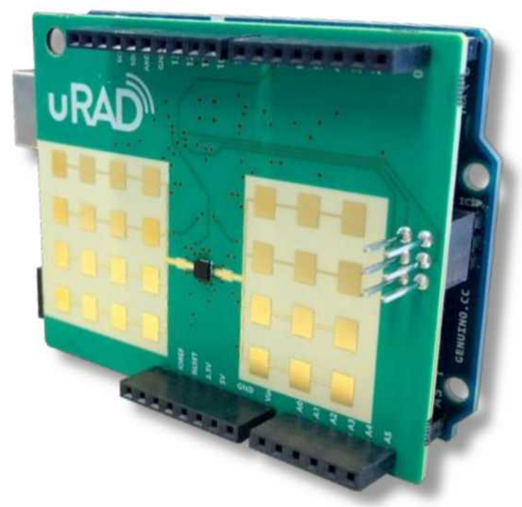

Figure 2.6: uRad Arduino compatible board [2]

The pulse repetition frequency (PRF) in the uRad system does not explicitly state but it can be considered to be fairly low given that the range of measurement is below 100 m. The uRad radar system operated within the noise-free ISM frequency band of 24 GHz [17]. The system operates with a minimum bandwidth of 24 GHz and a maximum bandwidth of 24.25 GHz giving the system bandwidth of 0.25 GHz. The targets used in the uRad radar system range from humans and birds to buildings and cars. It can detect objects within line of site but it is not a necessity. It can detect objects through walls as well. Therefore, it can be used as a security device to detect a presence within a building.

The uRad system is a very well packaged and accessible radar with a wide range of functionality ready for a commercial or *tinkering* project. It also uses a generally available platform (Raspberry Pi and Arduino) which makes the option attractive. The radar yields very reliable results on a user friendly interface as can be seen in Figure 2.7. It is low-cost regarding radar equipment however, it excludes the generally interested parties of experimenting with radar systems since they are expensive to high school and university students. The system costs €199 for the add-on onto an existing Raspberry Pi or Arduino Uno.

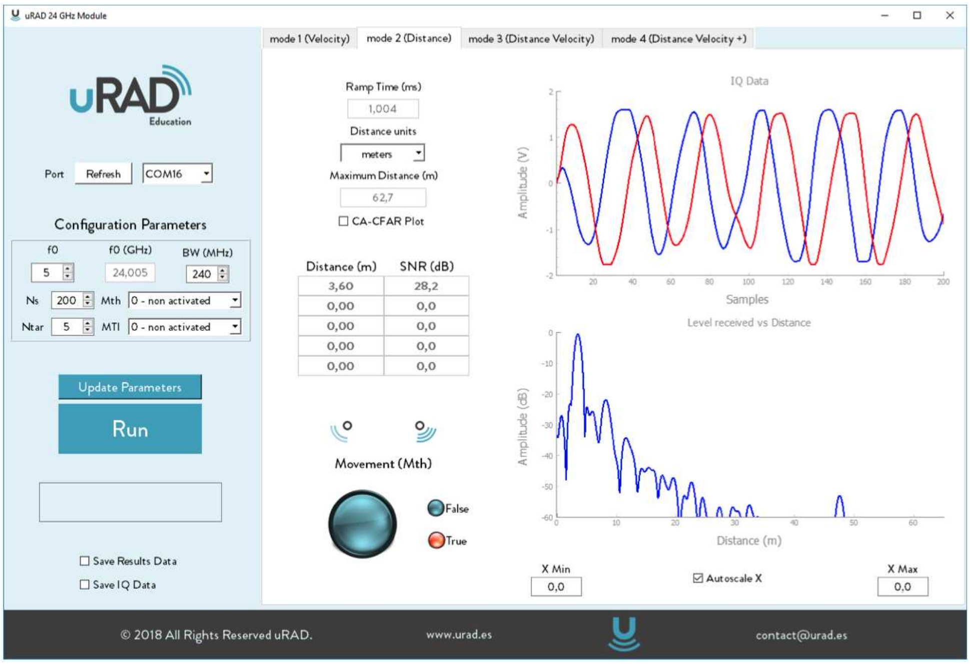

Figure 2.7: uRad Results [2]

2.3.2 Low-Cost Mini Radar

Another research project is the monostatic radar developed by Tarchi et al. The Low-Cost Mini Radar uses FMCW and the reason for using FMCW was to keep the physical size of the package small. FMCW is a relatively old technology, hence it can be made into a relatively small footprint [3]. The small package is necessary since the project would be used on unmanned moving platforms. The Low-Cost Mini Radar uses a particularly high PRF of 26.2610 MHz. The high PRF offers acceptable range being relatively unambiguous at 100 m but the high PRF also allows for accurate velocity measurement. The radar operates in the K_u band with centre frequency 17.2 GHz. The bandwidth of the system is a maximum of 1.4 GHz. The Radar was used to detect parking signs, light poles, parked cars and an external staircase. A physical implementation of the radar system can be seen in Figure 2.8.

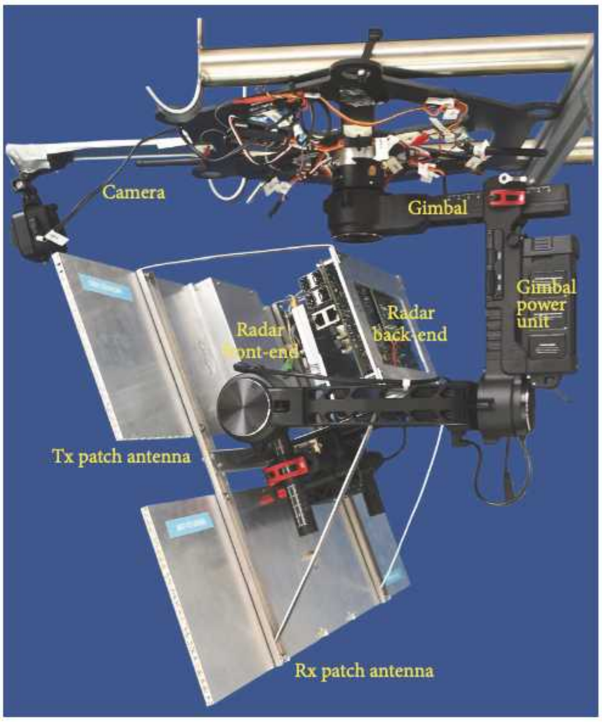

Figure 2.8: Low-Cost Mini Radar for unmanned moving platforms [3]

Results obtained by the radar when tested in a parking lot (seen in Figure 2.9) can be seen to identify parked cars, the tree line and a building wall in Figure 2.10.

2.3.3 Conclusion

In conclusion to the work laid out above, the Small-Scale, Embedded, Educational Audio Radar uses aspects from various previous works. The radar would need to implement a range and velocity measuring capabilities. Furthermore, the hardware and the implementation of radar systems are onerous and tend to be inaccessible to most. The radar systems investigated within this review are no different. The audio radar would improve on these short-comings in the following ways.

The MIT Radar offers an educational radar but still at a premium. The audio radar would reduce the cost and make radar technology even simpler without the need for expensive RF components. The accessibility of the designs and resources of the MIT radar would be utilised and mimicked in the audio radar to get more people involved in radar systems. The portability of the radar would be improved over the MIT radar but the simplicity of the radar would be maintained.

The educational courses offered worldwide attempt to provide an introduction to radar systems but the cost associated with it make them exclusive and difficult to attend given their physical location. The audio radar would attempt to make the data readily available

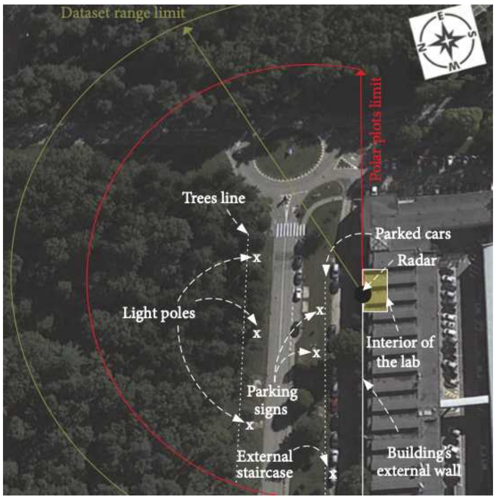

Figure 2.9: Testing ground for Low-Cost Mini Radar [3]

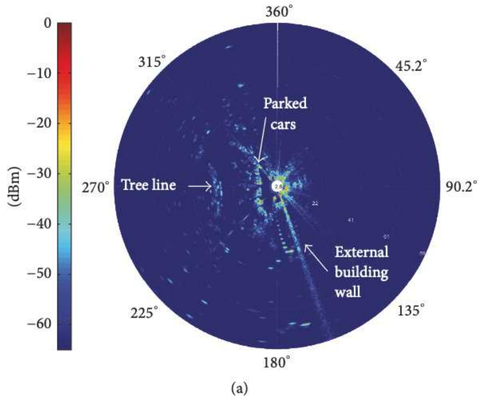

Figure 2.10: Results for Low-Cost Mini Radar [3]

and remove the barrier to the basic understanding of how these systems work.

Student developed radar systems form a vital part in the preparation and execution of this project. Previous research is used and refined upon in their work and would be done in this project as well. The processing of the signals in the student developed radars were all done after the data was collected and not in a timeous manner. The audio radar will process the data as soon as received to display results within seconds. The project would also make use of readily available *breakout* boards to render the project even more accessible and usable as a learning tool.

The uRad offers a lot of similarities to the audio radar where it also detects the same objects at similar ranges. It offers up to 100 m range, but the technology has the same focus. It is in a usable form factor using existing hardware[1]. The uRad system has a user friendly interface that the audio radar would build on. Most radar systems for low-cost implementation, including the uRad system, utilised the FMCW radar method since it is an older technology that is refined at this point but is more complex compared to CW. The audio radar would not implement FMCW but the theory behind it is still valuable to this project. The project would also improve on the portability of the radar and offering a PC free environment for obtaining results. This would be done using a web server linked to the back-end of the radar.

[1]Raspberry Pi and Arduino

Chapter 3

Radar Theory

Radar technology is a broad field, and aspects of radar theory covered below only hold significance to the audio radar and is not an attempt to reproduce a radar textbook but to aid in the understanding of the design process followed in Chapter 4.

3.1 Principles of Radar Operation

3.1.1 Range Concepts

To measure range with radar, the speed of type of signal through the medium it is travelling needs to be considered. In the audio radar context, an audible signal travels through the atmosphere at a speed of 343 $m.s^{-1}$. A chirp[1] is transmitted out from an antenna (speaker) and then the receiver (microphone) listens for the return echo that would have reflected back after hitting the target. In order to measure the distance to the target, Equation 3.1 is used.

$$\tau = \frac{2R}{c} \tag{3.1}$$

Where τ is the time delay between the start when the transmitting pulse was transmitted and the returning echo. R is the range that to the target and it is multiplied by 2 since the pulse travels to the target and back. The speed of sound is c. The range concept using the Equation 3.1 can be better explained using Figure 3.2.

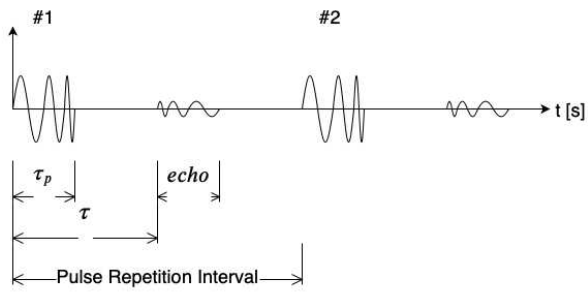

Figure 3.1: Pulse Repetition Interval[4]

[1]See section 3.2.1

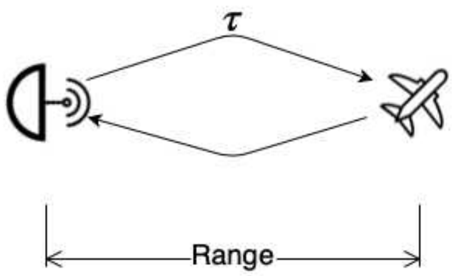

Figure 3.2: Radar Range Concept [4]

3.1.2 Range Resolution

Range resolution can be described as the minimum distance two or more targets need to be apart to be distinguishable from each other [5]. If two targets are too close together, they will show up as a single object on the resulting range line. To better explain this concept, Figure 3.3 depicts the concept visually.

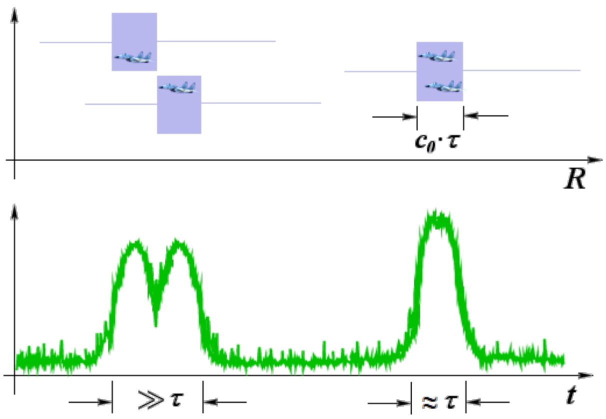

Figure 3.3: Range Resolution [5]

The range resolution in Figure 3.3 shows that when the two fighter jets are flying above each other, they are perceived as a single target on the spike in the received signal. Equation 3.2 governs the range resolution and is in terms of the speed of sound (c) [2] and the bandwidth (B) of the chirp.

$$\Delta R = \frac{c}{2B} \tag{3.2}$$

Where ΔR is the minimum distance that two objects can be apart to be able to detect them as two distinct targets.

[2] Working in audio spectrum

3.1.3 Unambiguous and Ambiguous Range

The concept of the unambiguous and ambiguous range goes hand in hand with each other. The unambiguous range of a radar is the maximum distance that a transmitted pulse can travel to a target and back before the next pulse is transmitted [18]. In Figure 3.4, the echo is only received after the second pulse is transmitted. The radar would consider this echo to be associated with a target much closer than in actuality. To calculate the

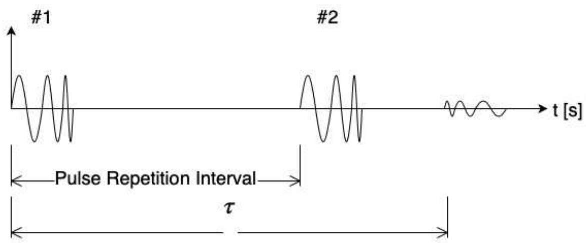

Figure 3.4: Unambiguous Range

maximum range that the radar can detect a target unambiguously Equation 3.3 is used. Consider Figure 3.3, and assume a target 225 m away[3] with $\tau = 1.312\ s$ and $PRI = 1\ s$

$$\begin{aligned} R_u &= \frac{c}{2PRF} \\ &= \frac{343}{2(1)} \\ &= 171.5\ m \end{aligned} \tag{3.3}$$

In actual fact the radar perceives the target to be at $225\ m - 171.5\ m$ = **53.5 m**which can be classified as the ambiguous range of the radar.

3.1.4 Doppler Frequency and Effect

Doppler Frequency can be obtained from the phenomenon of the Doppler Effect discovered by Christian Doppler [19]. The Doppler Effect is the perceived change in the frequency of a source depends on whether the radial distance between the source and observer changes. If the source is moving towards the listener, the frequency is higher since the waves consecutive waves travelling towards the listener would have a shorter distance to travel as time progresses. This is assuming that the source frequency and wavelength and the speed of sound remain constant. To calculate the Doppler Frequency f_D, Equation 3.4 is shown.

[3]Using Equation 3.1

$$f_D = \frac{2v}{\lambda} \tag{3.4}$$

Where v is the radial velocity of the source and λ is the wavelength of the source.

The Doppler Effect can be described using Equation 3.5

$$f = \frac{c + v_t}{c - v_s} f_c \tag{3.5}$$

where f_c is the centre frequency, f is the observed frequency, c is the speed of sound, v_t is the velocity of the target, and v_s is the velocity of the source of the sound.

3.2 Radar Frequencies and Bandwidth

Considering that only CW and Pulse-Doppler radar techniques will be used in this project, only the frequencies and bandwidth

3.2.1 The Chirp Signal

The Chirp Signal can be described as a form of frequency modulation being used in applications such as radar, sonar, image processing just to name a few. The idea behind a chirp signal is to have a sinusoidal signal that either ramp up or ramp down in frequency in a linear manner[4].

The chirp rate can be calculated using the frequency where the chirp should start and the frequency where it should end [20].

$$k = \frac{f_1 - f_0}{T} \tag{3.6}$$

Where k is the chirp rate, f_1 and f_0 is the end frequency and starting frequency respectively, and T is the time duration for the entire chirp.

The chirp signal can, therefore, be calculated as

$$y(t) = cos\Big(2\pi(\frac{k}{2} + f_0)\ t\Big) \tag{3.7}$$

The above equation results in a signal as seen in Figure 3.5 and the FFT of the signal showing the frequency spectrum ranging from 1 Hz to 30 Hz in Figure 3.6.

[4]Quadratic and exponential chirps also exist but only the linear version will be discussed here.

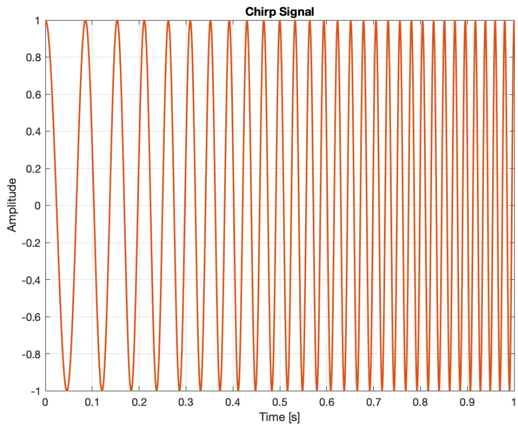

Figure 3.5: Chirp Signal - Time Domain 1 Hz to 30 Hz

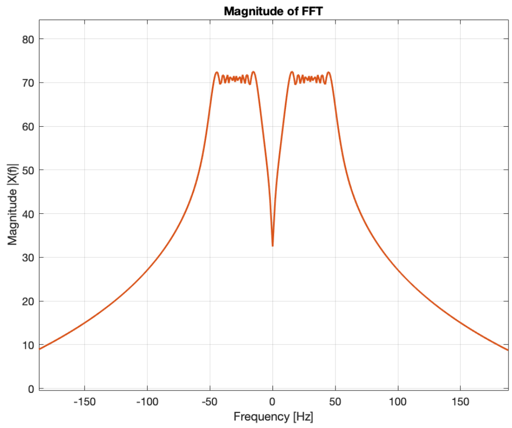

Figure 3.6: Chirp Signal - Frequency Domain 1 Hz to 30 Hz

3.2.2 Bandwidth

The bandwidth of the signal can be described as the difference between the higher frequency and the lower frequency in the chirp as depicted in Figure 3.6.

3.3 Waveform Types for Radar Applications

3.3.1 Pulse-Doppler Radar

Pulse-Doppler radar is when pulses are transmitted at a certain frequency or with a chirp which starts at a specific frequency and goes up to a higher frequency with the bandwidth. A pulse is transmitted after which a listening time where no sound is played. The listening time is used to wait for the echo from the target to return. The distance is then determined by taking into account the time delay from the transmission of the pulse to receiving the echo. Since the speed of sound in air is known, the range of the target can be determined using the formula below. [4]

$$R = \frac{t_d c}{2} \tag{3.8}$$

Where R is the range of the target, t_d is the time delay before an echo is heard and c is the speed of sound in air.

The Doppler Frequency is used to determine whether or not the target is moving with a radial velocity relative to the radar transmitter. The Doppler Frequency is determined by checking by how much the target has moved over the number of pulses.

3.3.2 Continuous Wave (CW)

Continuous Wave (CW) radar is when a constant frequency tone is played out for a set amount of time or indefinitely and direct measurement of the Doppler shift of the return signal. The wavelength of the returned signal will be different from the transmitted signal. This is known as the Doppler Frequency. CW radars can not be used for range estimation or measurement but only for velocity measurement. There is no basis for measuring the time delay. In Figure 3.7 below, the time delay in the wavelength can be seen graphically. [21]

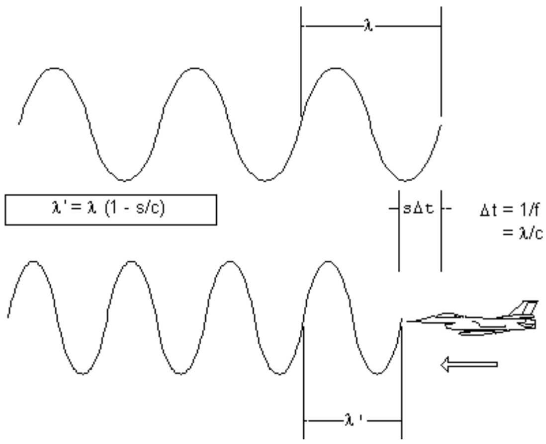

Figure 3.7: Illustration of the Wavelength Change of the Return Signal

3.3.3 Frequency Modulated Continuous Wave (FMCW)

Frequency Modulated Continuous Wave (FMCW) radar is used to measure range using the CW radar above but by modulating the frequency. The frequency is varied linearly and the transmitted signal is therefore 'timestamped' by the specific frequency that it is at. This timestamp is then used to determine the time delay between transmit and receive and in turn used to calculate the range of the target. The theory can be seen in Figure 3.8 below. [21]

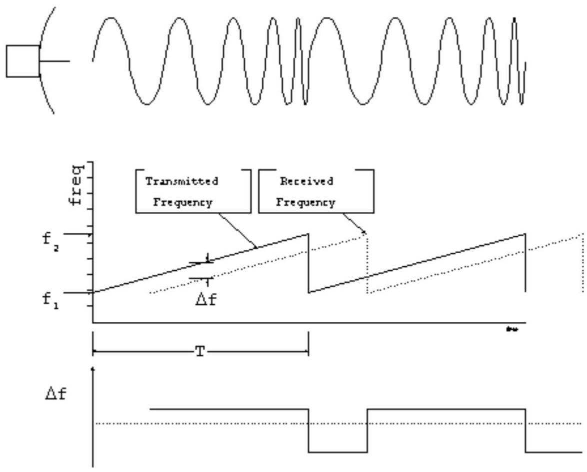

Figure 3.8: Illustration of the Frequency Modulation within FMCW

3.4 Windowing Functions

The only windowing function used in the processing of the signals is the Hamming Window. Figure 3.9 shows the time domain plot of a Hamming Windowing function and also the FFT of the signal. The Normalised Equivalent Noise Bandwidth of the Hamming

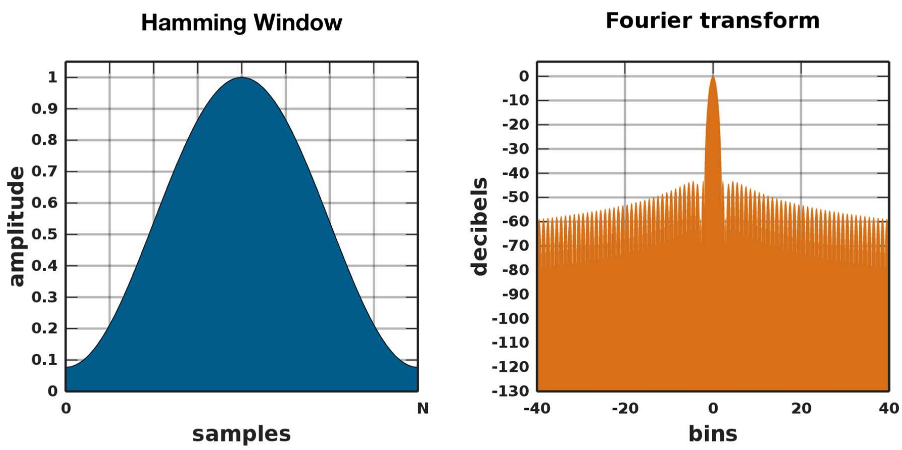

Figure 3.9: Hamming Windowing function
[22]

Windowing function (NENBW) is 1.3628 bins. The theory behind the windowing function would not be investigated since it falls outside the scope of this project. However, it is important to note the windowing function used in the signal processing. The Hamming Window can be obtained using mathematical equation

$$w[n] = a_0 - (1 - a_0) \cdot cos\left(\frac{2\pi n}{N}\right) \tag{3.9}$$

where $a_0 = 0.54$ and this categorises the windowing function as a Hamming Window.

3.5 Signal Processing

3.5.1 Downmixing and Low Pass Filter

Downmixing of a signal involves the process where the signal is downmixed and the results is a complex version of the signal in its baseband. Sine and cosine waves can be considered the same signal but the cosine signal has a phase shift of 90 from the sine. Multiplying the received signal by sine and cosine at the same centre frequency results in a spectrum with components at $2f_c$ and $-2f_c$ and the added components at 0 Hz. Passing the complex signal through a digital low pass filter results in the downmixed signal at baseband in its In-Phase and Quadrature components. An illustration of the entire process can be seen in Figure 3.10 and the theory behind the low pass filter can be seen in Figure 3.11.

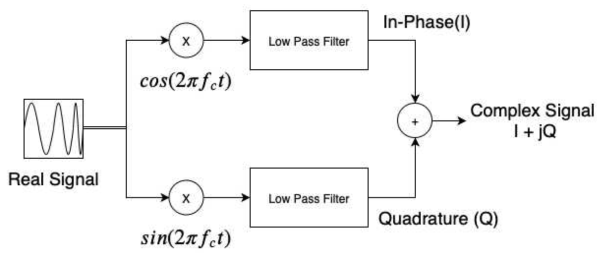

Figure 3.10: Downmixing Signal to Base band

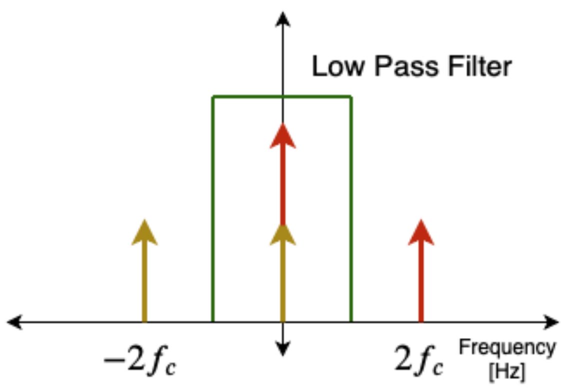

Figure 3.11: Low Pass Filter within Downmixing Function

3.5.2 Matched Filter

A matched filter is the method to increase the signal to noise ratio (SNR) of a received signal. The known transmitted pulse is used to correlate with the received signal. The transmitted pulse is used to detect the presence of the pulse within the unknown received signal. An illustration of the matched filter can be seen in Figure 3.12.

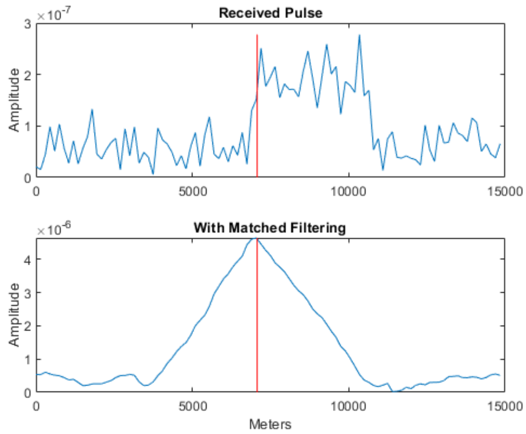

Figure 3.12: Illustration of Matched Filter [6]

The SNR of the received signal is greatly improved after applying the matched filter making object detection easier. The matched filter improves the probability of detection $P_d \approx 90$ % and the probability of false detection $P_{fa} \approx 10^{-5}$. It is possible for the match filtering to take place in the time domain but the easier implementation is done in the frequency domain as is the case for this system.

3.5.3 Range Line vs Time

The matched filter essentially translates to the range line needed to display the distance to detected targets. The array containing all of the range data is currently in a vector form and the vector needs to be broken up into the different number of pulses that were played out initially. The `np.reshape()` function is used to reshape the vector into a matrix of shape n x m where n is the rows consisting of the number of pulses and m is the range measured to the target. The general function of going from the rangeline to a matrix is illustrated in Figure 3.13.

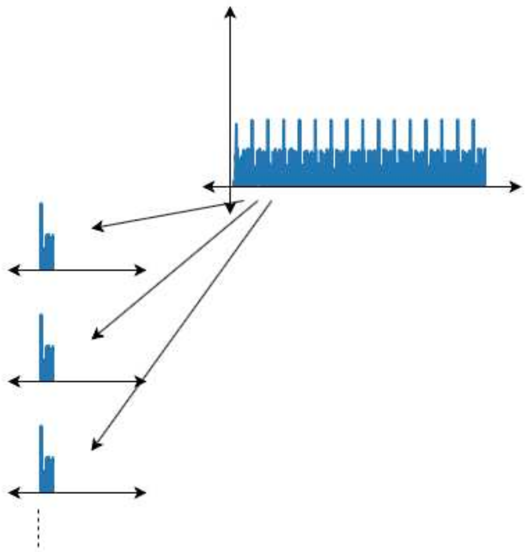

Figure 3.13: Illustration of Rangeline to Range Matrix

Chapter 4

Methodology, Design and Implementation

The methodology followed to design and test the implemented radar follows the V-Model shown in Figure 4.1. The system would be broken down into sections and subsystems and tested individually before the integration of the system. Finally, the system as a whole would undergo acceptance testing per the user- and functional requirements and technical specifications. The requirements and specifications can be viewed in Section 1.4.

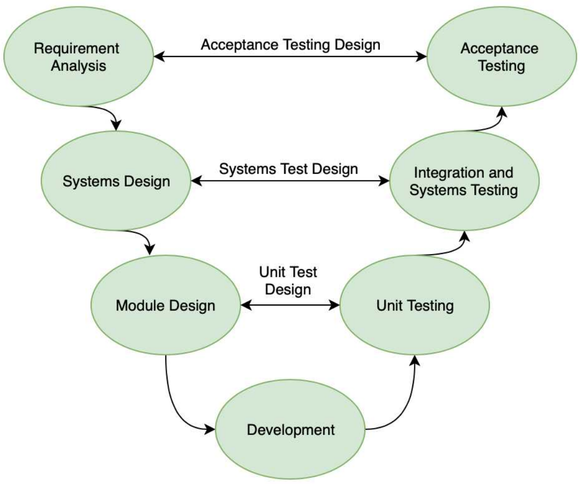

Figure 4.1: V-Model used for Design, Implementation and Testing [7]

4.1 Systems Design of the Radar

The system needs to consist of the basic components set out in Figure 4.2. In the following sections, all of the individual subsystems would be tested and an appropriate component would be chosen for the final radar. Figure 4.2 displays the basic layout of the system and different variations of this system would be experimented with.

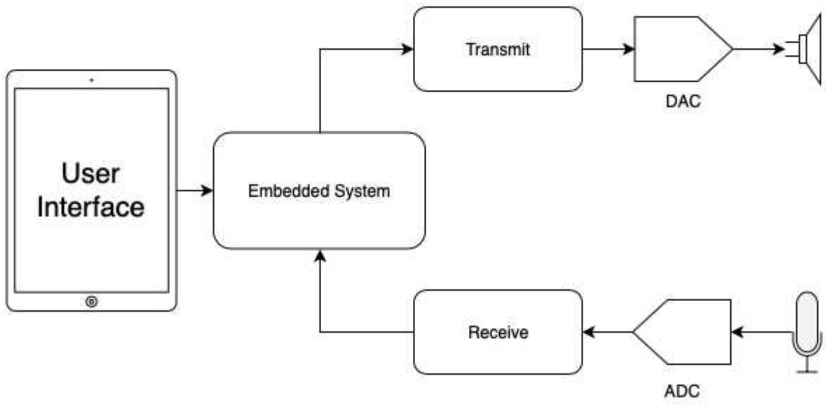

Figure 4.2: Overview of Technical Requirements

4.2 Module Design

4.2.1 Embedded Platform

The transmission and receiving sections of the radar can be accommodated on the Teensy 3.6 microcontroller. The Teensy board has a 32 bit 180 MHz ARM Cortex-M4 processor with the ability to process FFT's in real-time. The Teensy with an Audio Shield sold separately, offers I^2S audio output and input. The Teensy 3.6 can be seen in Figure 4.3 and the Audio Shield in Figure 4.4.

Figure 4.3: Teensy 3.6

Figure 4.4: Audio Shield for Teensy 3

The audio is of CD-quality being sampled at 44.1 KHz and 16-bits and streams automatically as the Arduino script executes. The Teensy board was benchmarked against other Teensy and Arduino boards where it came out on top. The results can be seen in Figure 4.5[23].

The Teensy board could be used to interface to a speaker using the on-board digital-to-analogue converter (DAC) to play the audio and also to use the on-board ADC to record the returning sound using a microphone. The board can perform FFTs in real-time which is needed to process the received signal. However, the Teensy only has 256 kb of on-board random access memory (RAM) and hosting a user interface without the need for

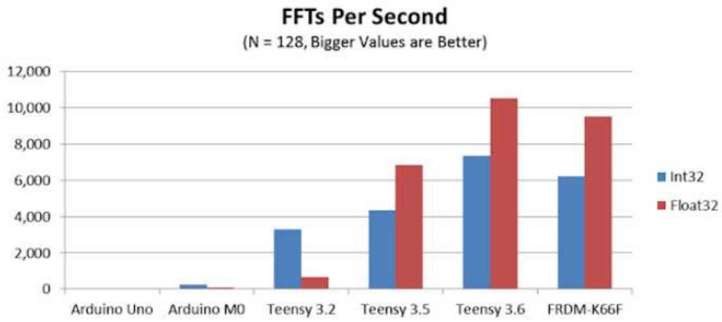

Figure 4.5: Overview Teensy performance in FFT

an external PC would be difficult.

The second option is to use the Raspberry Pi 3B (RPi) as an embedded platform. The RPi is a credit card-sized mini-computer with Quad Core 1.2GHz Broadcom BCM2837 64bit central processing unit (CPU) and 1GB of RAM. The RPi is not inherently a real-time operating system (RTOS) like Arduino or Teensy boards. Yet it offers faster processor speeds and more RAM for signal processing and a user interface. The RPi does not have an onboard DAC or ADC. Figure 4.6 shows the Raspberry Pi 3B.

Figure 4.6: Raspberry Pi 3B

The two platform's key components and performance are compared head to head in Table 4.1.

The boards are both similarly priced. Although the Raspberry Pi does not have on-board audio support, the complexity of the Teensy Audio interfaces is not conducive to an educational device. The readily available breakout boards to add the audio capabilities are also affordable with a large community for support. Furthermore, the Teensy 3.6 board does not have enough RAM and no easy and apparent way to host an interface, the obvious choice is the Raspberry Pi 3B. The RPi offers the ability to host a web server as well as the on-board HDMI port if a monitor is needed to be connected. The superior CPU and RAM on the RPi would aid in doing multiple signal processing algorithms one

	Teensy 3.6 [24]	**Raspberry Pi 3B [25]**
Processor	180 MHz ARM Cortex-M4	1.2 GHZ quad-core ARM Cortex A53
RAM	256 KB	1 GB SDRAM
Communication Protocols	USB, Serial, SPI, I2C, CAN Bus	I2C, SPI, UART, Ethernet, Bluetooth
Audio Ability	2 ADCs 13 bit 2 DACs 12 bit	PWM 3.5mm socket
Power Consumption	80 mA[26]	730 mA[27]
Price	$29.95	$35
Shortcomings	Not enough RAM for webserver. No ability to add monitor.	Only PWM ∼11bit audio out. No on-board ADC

Table 4.1: Comparison: Raspberry Pi 3B and Teensy 3.6

after another. The result would be displayed to the webserver without much delay.

4.2.2 Transmitting Hardware

Speakers

On the transmitting side of the radar, the speaker needs to have a flat frequency response over the operating range of audible frequencies used by the radar. Three different speakers were tested and their frequency response analysed to obtain the speaker with the flattest response.

Firstly, the Visaton Oval Speaker offered a different shape to the usual round speakers. This leads to a wider beam of sound in the narrower axis speaker dimension. Since the radar would be used in louder environments like open-days, it is worth experimenting with. Secondly, the RS Pro unflanged speaker offers the same nominal power to as the Visaton Oval Speaker but has a waterproof PET dome. The speaker doesn't need to be waterproof but it is yet an affordable option. Finally, the RS Pro flanged speaker with a nominal power of 1 W was chosen for the convenience of mounting the speaker and to test whether a less powerful speaker would be sufficient. Table 4.2 shows the specifications of each speaker.

	Impedence	Nominal Power	Price	Shape
Visaton (Flanged)	8Ω	$2W$	R 140.74	Oval
RS Pro (Unflanged)	8Ω	$2W$	R 59.82	Round
RS Pro (Flanged)	8Ω	$1W$	R 63.45	Round

Table 4.2: Comparison: Speakers

The frequency response of each speaker was tested in a lab and analysed with a single microphone[1] across all of the speakers. Test conditions were as follow:

- Signal generator set to 1 V_{pk-pk}
- MAX4466 powered with 5 V with bench power supply
- MAX4466 gain set to 125x [28]
- Respective speaker placed on chair 1 m from microphone suspended from the roof [2]
- Oscilloscope connected to OUT pin on MAX4466
- Frequency stepped by 500 Hz starting at 5000 Hz to 15000 Hz

The results for the speaker frequency response testing for the Visaton speaker can be seen in Figure 4.7. RS Pro 2W frequency response in Figure 4.8 and the RS Pro 1W in Figure 4.9. The frequency responses for the RS Pro speakers are extremely similar because they follow a similar design except for the nominal power rating that differs (therefore the higher gain associated with it) and the 2W speaker does not have a flange for mounting. Both the RS Pro speakers show a relatively flat response over the intended operating frequency of 8000 Hz to 12000 Hz. The Visaton speaker shows an erratic response over the operating frequency having a sharp decline at 8000 $-$ 9500 Hz.

The RS Pro 1 W proved to be more than sufficient for the operating frequencies and the flanged design makes for easy mounting in a finished product. The speaker is not the most powerful but the response is comparable to that of the 2 W speakers.

[1] MAX4466 Microphone Pre-Amp Audio Evaluation Board

[2] See Appendix C

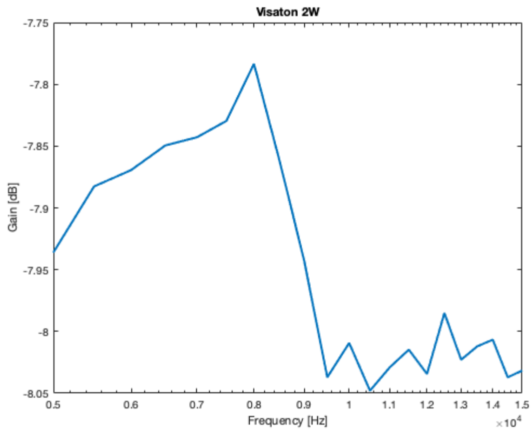

Figure 4.7: Frequency Response: Visaton 2W

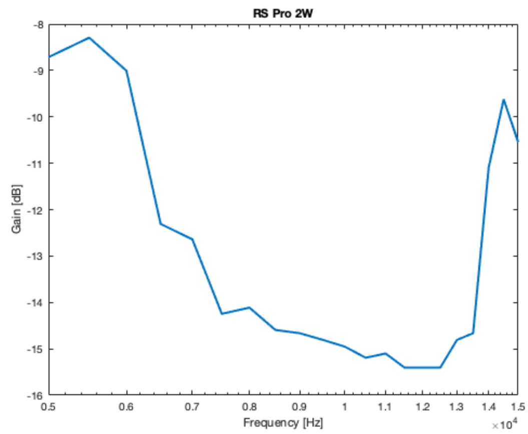

Figure 4.8: Frequency Response: RS Pro 2W

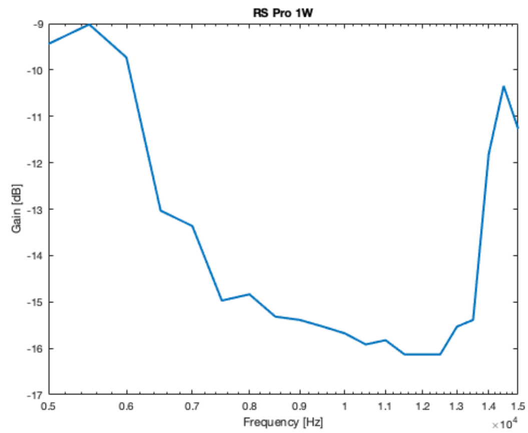

Figure 4.9: Frequency Response: RS Pro 1W

Digital-to-Analogue Converter (DAC)

Following the speaker tests, the DAC needed to be investigated to see that we have a linear and accurate response from the devices. The DACs need to take in a digital signal and convert it to an analogue signal for playing over a speaker. The DAC investigated is the Adafruit MAX98357 I^2S Class-D Mono Amp that features a I^2S digital audio device specifically made for microcontrollers. If the microcontroller has the ability of I^2S digital audio, like the Raspberry Pi and Teensy both support, the breakout board takes in digital audio, converts it to analogue, and amplifies it. When the device is powered with 5 V, the output is a maximum of 3.2 Won 4Ω impedance and 10% total harmonic distortion (THD) or 1.8 W on 8Ω with 10% THD [29].

The amplifier used is the device which would introduce non-linearities to the system. The frequency response of the device could not be tested individually since it is part of a complete breakout board. The datasheet, however, does support the hypothesis that the response is roughly linear in the operating range. Figure 4.10 shows the normalised frequency response of the MAX98357A amplifier when supplied with 5 V and the gain set to 12 dB.

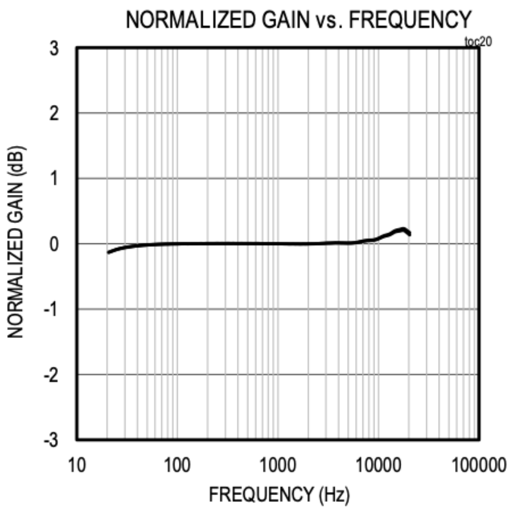

Figure 4.10: Frequency Response: MAX98357A Amplifier [8]

4.2.3 Receiving Hardware

Microphone

The receiving side of the radar, the microphone forms a vital part and its operation is crucial to the workings of a radar. Therefore, the frequency response is important since the signal processing would require accurate frequency pick up by the microphone. A similar approach to testing the speakers was followed to test the microphones.

Two different microphones were tested where the first is a breakout board with an adjustable gain set with a screwdriver on the back called the MAX4466 also used for testing frequency response of the speakers. The second microphone is the MAX9814 breakout board with an automatic gain. This ensures that the output from the microphone never clips the signal. Both of these microphones can be supplied with $2.4\,V - 5.5\,V$ making them both suitable for use with the RPi or the Teensy 3.6.

To ensure consistency in the testing of the microphones, a single speaker[3] was used at a distance of $1\ m$. The Set up can be seen in Appendix C. The OUT pins from both microphones were connected to an oscilloscope to measure their frequency response between $5000\ Hz\ -\ 15000\ Hz$ in steps of $500\ Hz$. Figure 4.11 shows the frequency response of the MAX4466 with its gain set to a maximum. Figure 4.12 shows the frequency response of the MAX9814 with automatic gain.

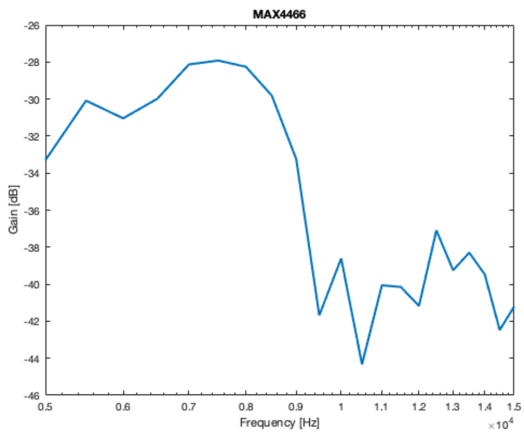

Figure 4.11: Frequency Response: MAX4466 Microphone

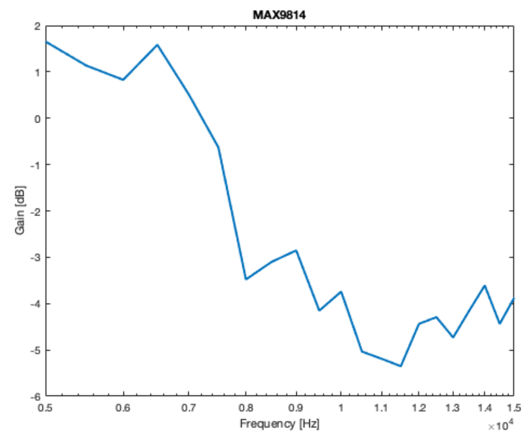

Figure 4.12: Frequency Response: MAX9814 Microphone

Both of the microphones have a relatively flat frequency response over the radar's operating frequency. However, there are some inconsistencies in both microphones especially with

[3]MacBook Pro 2015 right-hand speaker

the MAX4466 at higher frequencies. It should also be noted that the MAX4466 output voltage is significantly lower than that of the MAX9814 since the automatic gain of the microphone amplifies the quieter signal from the computer a lot more efficiently compared to the single setting gain. Therefore, MAX9814 microphone would be prefered to the adjustable gain microphone to work better in noisy environments where the radar might be used. All of the recorded data sets to verify the testing of the microphones and speakers can be found along with all the screenshots and photos taken, here.

Analogue-to-Digital Converter (ADC)

To convert the analogue signal obtained by the microphone, an ADC is required for the microcontroller to be able to process the data as a digital signal. The Nyquist sampling rate [30], which states that a signal must be sampled at or above at least twice the highest frequency that needs to be observed, need to be satisfied. To sample in the audible frequency range with a maximum of 20 000 Hz, the sampling rate should be at least 40 000 Hz. Ideally, the sampling rate of 44 100 Hz is desirable since this frequency is CD quality and sound frequencies are identifiable. Three options were considered for this process.

The first ADC is the MCP3008 I/P which offers a sampling rate of 200 kilo samples per second (ksps) when powered by a 5 V supply. The device offers a 10 bit resolution over 8 channels. The device is compact and widely used with microcontrollers and the low cost of the device makes it a competitive choice. A sampling resolution of 10 bits are not necessarily desirable but given the uncertainty regarding sampling rates and equally timed samples, this remains an option. Figure 4.13 shows the physically small size of the device.

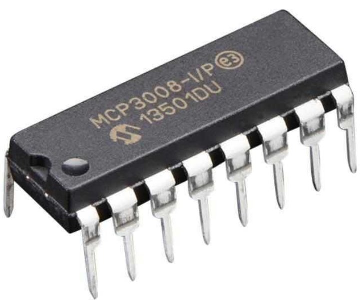

Figure 4.13: MCP3008-I/P physical package [9]

Initial tests using the Raspberry Pi and the SPI interface that the MCP3008 uses only offered a sampling rate of about 8000 Hz. This was as a result of each sample with the MCP3008 took 24 clock cycles to read the value. Therefore, this option delivered insufficient results.

Secondly, the Teensy 3.6 microcontroller was considered having a 16 bit ADC on-board. It can communicate with a Raspberry Pi through the USB port using UART. The Teensy board is flashed using the Teensyduino plugin and programmed using the standard Arduino IDE in C. This lends itself to be an easily programmable board with vast amounts of support online through forums and communities. The Teensy was used in conjunction with the RPi and sound recorded using the MAX9814 microphone. The code for the Teensy can be found in Appendix B. Initial results with the Teensy proved that the recorded sound reflected the tone that was generated using MATLAB. The code is also available in Appendix B. Figure 4.14 shows the tone received and Figure 4.15 shows the FFT of the tone. The tone generated was set to 10 000 Hz.

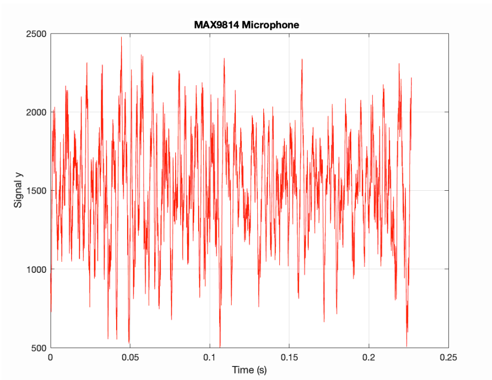

Figure 4.14: 10 000 Hz Tone Recorded by Teensy

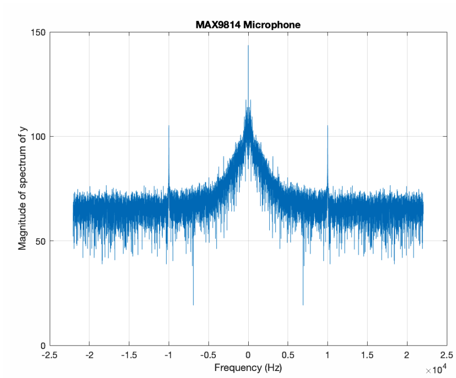

Figure 4.15: FFT of Tone Recorded by Teensy

Figure 4.15 shows that the frequency component is peaking at exactly 10 000 Hz. However, it is important to note that the maximum duration to stream in real-time, using the PySerial package in Python through USB, was only worth 0.224 s of sound data. This renders the Teensy, is only used as an ADC, expensive and not sufficient for recording tones long enough for processing in a radar.

Finally, a USB sound card compatible with Raspberry Pi was tested. The Astrum 3D Sound Card features a mono microphone input socket for audio recording. It also has an audio out but the radar requires that the microphone record and the speaker play at the same time and this device does not allow for simultaneous recording and playing. The sound card samples audio through the input jack at 44 100 Hz at 16 bit sample resolution. This is a sufficient sampling rate and CD quality. Therefore, suitable for the radar. Figure 4.16 shows a visual representation of a spectrum of frequencies (9 000 Hz in this case over 1 s) called a spectrogram and shows how accurately the sound was recorded.

The spectrogram at 9 000 Hz shows that the ADC sound card samples the data at a sufficient rate with a resolution exceeding the requirements. Table 4.3 shows the direct

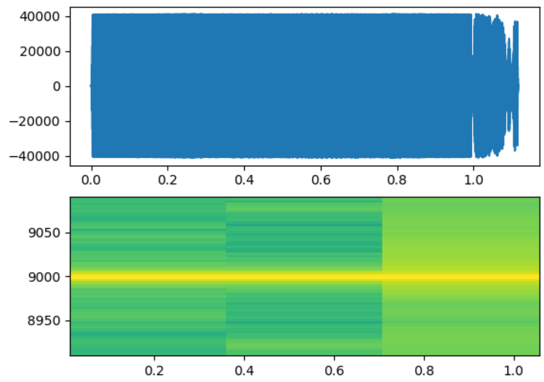

Figure 4.16: Received tone and spectrogram of 9 000 Hz tone

comparison between the most important parameters of the three options considered.

	Practical Sampling Rate	**Sampling Resolution**	**Price**	**Ease of implementation**
MCP3008	8 $ksps$	10 bit	R 27.11	Decent
Teensy 3.6	44.1 $ksps$	16 bit	\$ 29.95	Hard
Astrum USB Sound Card	44.1 $ksps$	16 bit	R 99	Easy

Table 4.3: Comparison: ADC parameters

Therefore, it can be concluded that the USB sound card, given the low-cost, CD-quality sampling ability, and the ease of use, is the ADC of choice for the radar project.

4.2.4 Software

The software used to host a web app needs to have the ability to be easily understandable as well as easy to implement given the requirements of the project. Two options that were investigated were Flask and Django, and both have proved to be widely popular under python web developers.

Django

Django is considered a full-stack web development platform. It offers developers the ability to implement common web site protocols without much effort. It does, however, lack some common features that Python offer. Django is considered to be an environment with all the add-ons and functionality already included in its source, leaving nothing out to develop a fully operational web application. It offers everything from an admin panel to database structures right out of the box.

It is considered to be a 'batteries included' approach to web application development [31].

Flask

Flask offers the minimum tools to get a web application working and if additional functionality is sought after, the flask framework is open to third-party applications and packages. It is more simplistic in its design and it lets you decide how, and whether you want a certain functionality implemented.

Flask is considered to be a more explicit implementation of Python and that makes it easier to read the code and understand what is happening [31].

Software Comparison

Table 4.4 shows the direct comparison between Flask and Django.

	Django	**Flask**
Resource Intensive	Medium footprint	Light footprint
Simplicity	Harder to read, implement	Easily readable and implemented
Built-in Functionality	All-inclusive functionality	Not a lot of functionality out-of-the-box
Expandability	Medium	High
Support	2631 StackOverflow questions [31]	575 StackOverflow questions [31]

Table 4.4: Software Comparison

Therefore, for the simplicity and the functional working of Flask, it was chosen as the preferred software package to use to implement the web app. As stated in [31], Flask is also more focused on the experience and educational aspect of development. The Raspberry Pi is also not the most powerful computer and hence the lightweight footprint of the software contributes to the choice.

Software Implementation and Testing

A Flask web application was set up to run on the Raspberry Pi. An HTML template was generated using Bootstrap Studio to improve on the user experience. The packages imported into Python include:

- `request` used to process forms submitted to the app.
- `render_template` used to render the HTML templates for the user interface.
- `Flask` is used to implement and initiate the Flask web application.

Seven web pages were developed to aid in both a technical and non-technical implementation of the Continuous Wave and the Pulsed-Doppler radar modes. One of the pages is for a home (landing) page, one for an error handling and the other a results page. Each web page is scalable to fit the width of 4 different devices[4]. Figure 4.17 shows the web application implemented on a tablet-sized screen.

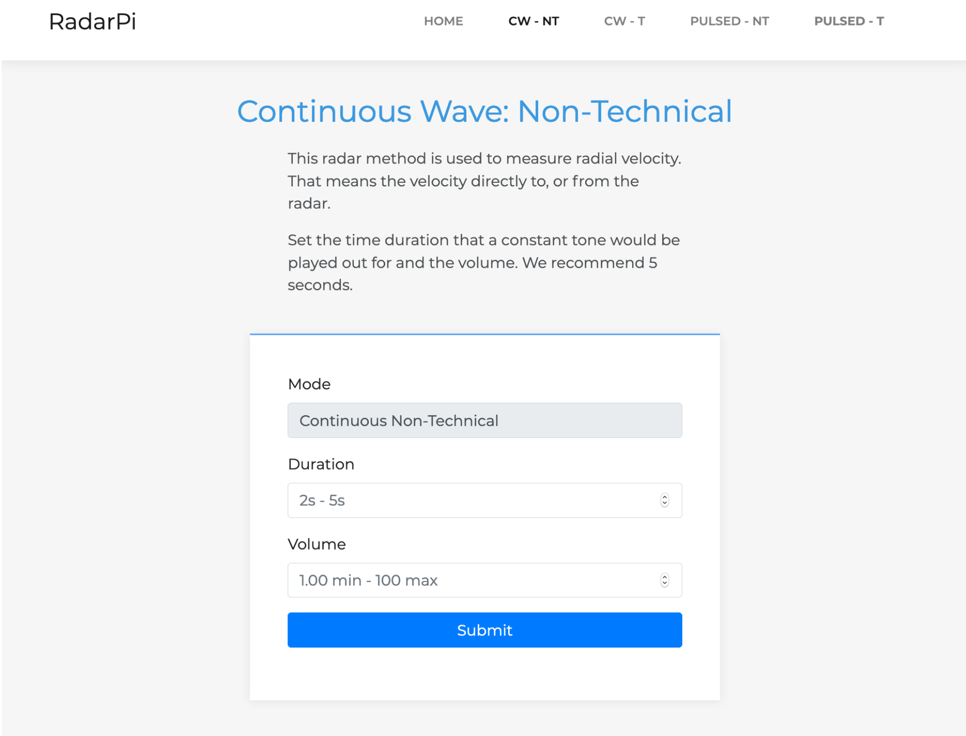

Figure 4.17: Website User Interface: Continuous Wave (Non-Technical)

Figure 4.18 shows the web application displaying results after testing the continuous wave radar by rapidly approaching and retreating a hand to and from the radar.

[4]Widescreen desktop, desktop, tablet and phone

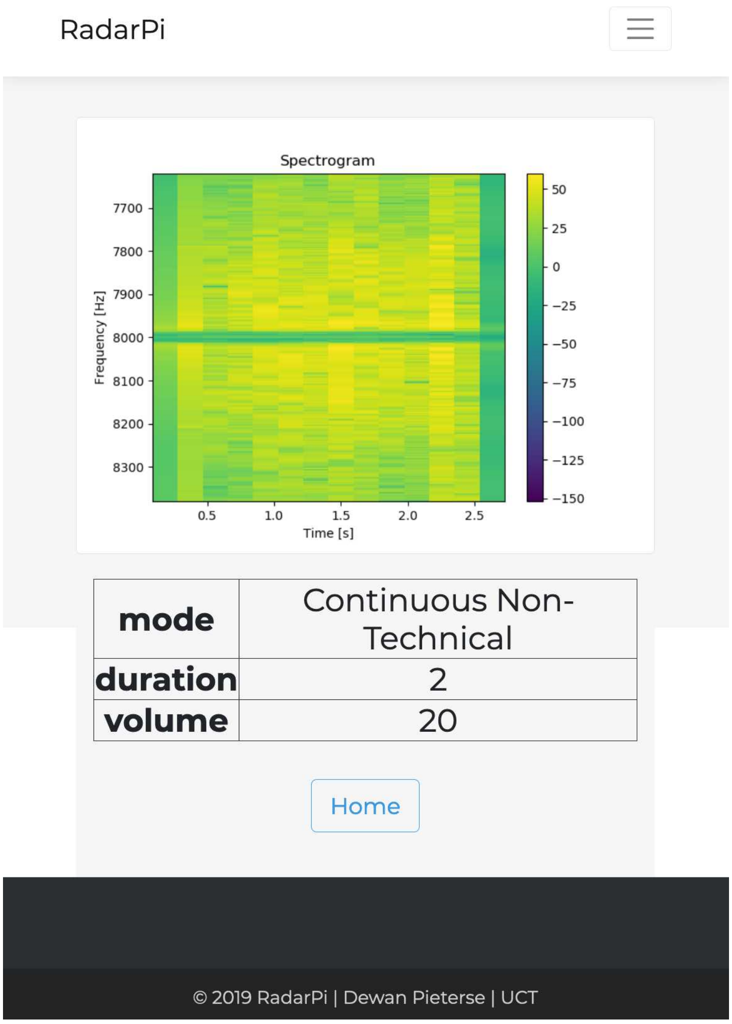

Figure 4.18: Website User Interface: Results

4.2.5 Enclosure Design

A simple enclosure was sought after to merely protect the components and render the radar to be portable. Figure 4.19 shows the basic design for the enclosure from the front with the lid on while Figure 4.20 shows the components visible with the lid removed.

4.3 Selection Summary

Table 4.5 shows the summary of all the selected subsystems after being tested and scrutinised. The budget constraint is also satisfied in the table.

It is important to note that the remaining budget would be partly spent on the enclosure, although only a small amount. The low cost is a user requirement of the radar as specified in Section 1.4.

Figure 4.19: Enclosure with Lid Installed

Figure 4.20: Enclosure with Components Visible

4.4 Signal Processing of the Radar

The signal processing of the radar is the process where the signals are created to be played out and the analysis and processing of the received signal as received by the microphone. The processes followed will use the received data to ultimately generate a Range-Doppler Map where the radial distance to an object will be visible as highlighted colours and the velocity of an object in the case of the Continuous Wave Radar.

The Range-Doppler map would be obtained with generating a set number of pulses consisting of chirps where the velocity map would be obtained with a constant tone using the Doppler Effect (both methods as described in Chapter 3). The following sections show the design of the signal processing used for both applications.

Component	**Selection**	**Price**
Embedded Platform	Raspberry Pi 3B	R 534.25
Speaker	RS Pro 1 W (Flanged)	R 63.45
DAC	Adafruit MAX98357 I^2S Class-D Mono Amp	R 85
Microphone	Adafruit MAX9814	R 228.25
ADC	Astrum USB Sound Card	R 99
Web App Software	Flask	Open-source
		R1009.95

Table 4.5: Final component selection

4.4.1 Range-Doppler Radar Signal Processing and Implementation

Generate Transmit Pulse and Signal

The transmit pulse and transmit signals are generated using the Numpy Python package and a cosine function. The time element is generated by the user entering the bandwidth of the chirp where the duration is calculated as:

$$T = \frac{100}{B} \tag{4.1}$$

The chirp rate μ (mu), is calculated using the duration and the bandwidth as:

$$\mu = \frac{B}{T} \tag{4.2}$$

The single chirp is then generated with the code:

```
y = (np.cos(2 * np.pi * (f0 * t + 0.5 * mu * t**2)))
```

The individual chirp is then duplicated using the `np.tile()` function to generate the pulse train needed for transmission. The transmit pulse of bandwidth 4000 Hz, unambiguous range of 10 m, the range resolution of 0.05 m, and centre frequency of 8000 Hz can be seen in Figure 4.21. The transmit signal can be seen in Figure 4.22 with 32 pulses that are just the repeated pulse from the transmitted pulse.

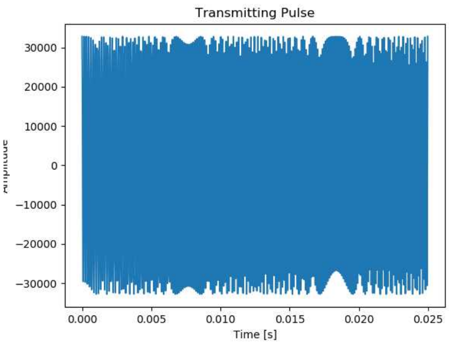

Figure 4.21: Transmitting Pulse

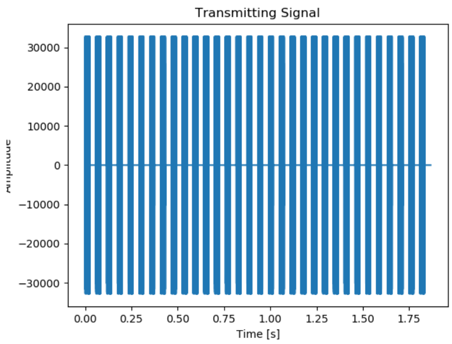

Figure 4.22: Transmitting Signal

Downmixing and Low Pass Filter

Next, a function was written to implement the downmixing and low pass filtering of the received signal. The function written to implement the downmixing takes in sampling time, the bandwidth of the chirp, centre frequency, the signal itself and the time over which it has been sampled. It returns the downmixed and low pass filtered complex signal in its baseband. The code snippet below shows the function written in Python.

```
def downMix(ts,B,fc,signal,time):
    fs = 1/ts
    pi = np.pi
    [numtaps, f] = 101, (fc + B/2)
    coeffsLowPass = scipy.signal.firwin(numtaps, f, pass_zero=False, fs=fs)
    cos = np.cos(2 * pi * fc) * time
    I_tp = np.multiply(signal, np.transpose(cos))
    I_tp_LPF = scipy.signal.convolve(I_tp, coeffsLowPass)
    sin = -np.sin(2 * pi * fc ) * time
    Q_tp = np.multiply(signal, np.transpose(sin))
    Q_tp_LPF = scipy.signal.convolve(Q_tp, coeffsLowPass)
    return I_tp_LPF + (1j * Q_tp_LPF)
```

Matched Filter

The received signal in Figure 4.23 is downmixed and them the matched filter is applied in Python using the code below. The function takes in the complex downmixed received signal and the complex downmixed transmitted pulse and returns the range line that is sought. It has the output that can be seen in Figure 4.24.

```
def matchedFilter(tp,r):
```

```
N1 = len(tp)
N2 = len(r)
N3 = N2 - N1;
tp_ZP = np.pad(tp, (0, N3), 'constant')
FFT_conj_tp_ZP = np.conj(fft(tp_ZP))
FFT_r = fft(r)
fftTpR = ifft(FFT_conj_tp_ZP * FFT_r)
return fftTpR
```

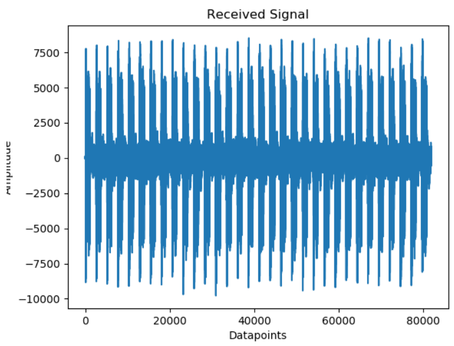

Figure 4.23: Received Signal

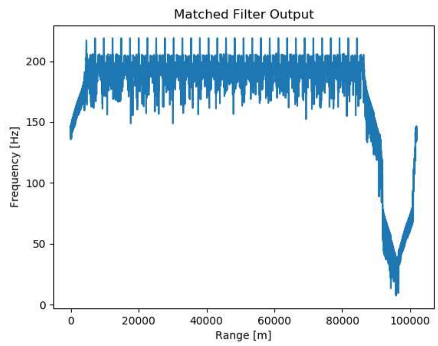

Figure 4.24: Matched Filter Output

Range Line vs Time

A range matrix follows the matched filter output as explained in Chapter 3. After obtaining the range matrix, the whole matrix undergoes a windowing function as explained in Section 3.4. In addition to windowing the matrix, the range matrix is also corrected for phase leakage that is introduced by the speaker and microphone playing and recording a sine wave. The first column of the range matrix is considered the 0° datum that the other phases would be measured against. The phase (angle) of the first column is used to generate a complete matrix with the same dimensions as the range matrix duplicating the first column. The range matrix is then multiplied by the conjugate (to cancel the phase leakage) of the phase leakage matrix and also multiplied by the windowing function. The matrix also needs to undergo an FFT that needs to be applied in the slow time. This refers to the rows (or the number of pulses) of the matrix. The output of the slow time FFT applied to the matrix is the range map.

The signal processing of the Pulsed-Doppler radar was executed and tested on a computer running MATLAB. The resultant range line from a test using real-world data can be seen in Figure 4.25 showing the successful implementation of the range map using a real speaker and microphone. The end result of the embedded radar would be compared to this result.

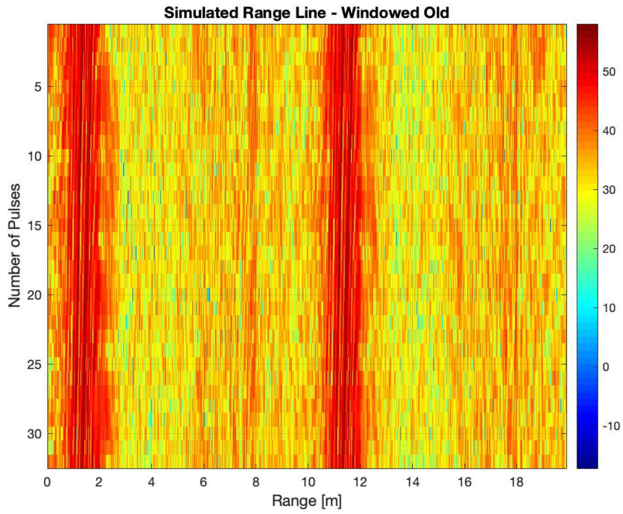

Figure 4.25: Range Map from MATLAB

The function (in Python) that windows and gets rid of the phase leakage that the radar experiences, and FFT the matrix can be seen below. It takes in the received signal matrix and returns the windowed, phase leakage corrected matrix and FFTed matrix that is the range map.

```
def hamming(RxSignalMatrix):
        n = RxSignalMatrix.shape[0]
    m = RxSignalMatrix.shape[1]
    h = signal.hamming(n)
    Window = np.transpose(np.matlib.repmat(h, m, 1))
    phaseLeakageVector = np.angle(RxSignalMatrix[ : , 1])
    phaseLeakMatrix = np.transpose(np.matlib.repmat(phaseLeakageVector, m, 1))
    RangeMatrix_Win_PhLeak = RxSignalMatrix * Window * np.conj(phaseLeakMatrix)
    RangeMatrix = fft(RangeMatrix_Win_PhLeak, axis=0)
    return RangeMatrix
```

Figure 4.26 shows the applied range map as implemented on the radar system. The system was tested and the setup can be seen in Figure 4.27.

The radar was handheld to reduce the clutter from the floor. As seen in the figure, the wall registered at roughly 3 m as was measured by the tape measure. The radar was powered by a power bank with a minimum power output of 7.5 W.

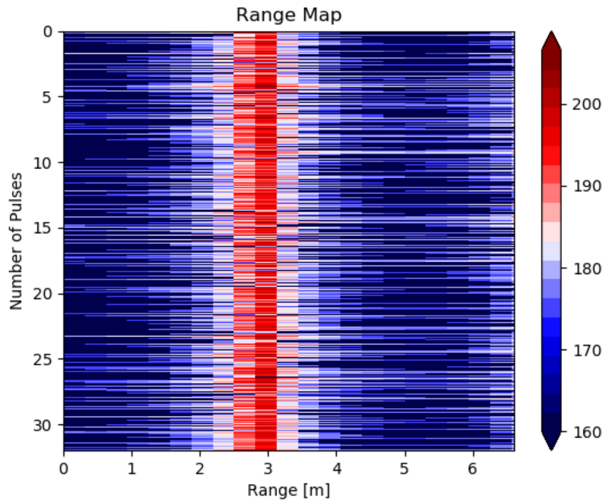

Figure 4.26: Range Map

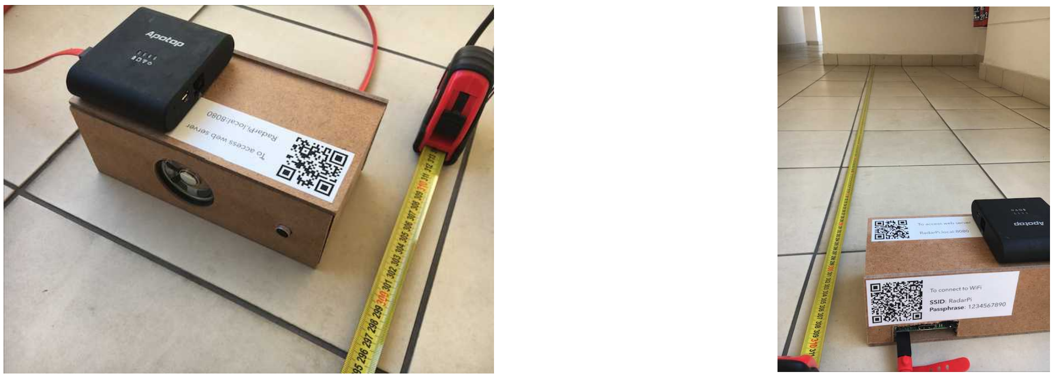

Figure 4.27: Setup for Testing Range Map

4.4.2 Continuous Wave Radar Signal Processing and Implementation

Frequency variation and Doppler Effect

The Continuous Wave radar implementation starts with generating a constant tone. The tone is generated using the Numpy package in Python and a sine function. The user would enter the duration of the tone to be played as well as the frequency. The function to generate the `wave` file can be seen below as implemented on the Raspberry Pi.

```
def waveGenerator(duration, frequency=8000):
    fs = 44100
    t = np.linspace(0, duration, fs * duration)
    y = np.int16(np.sin(frequency * 2 * np.pi * t) * 32767)
    path = "/home/pi/Documents/RadarPi"
    name = (os.path.join(path, "static", (str(int(frequency)) + 'Hz.wave')))
    wavfile.write(name, fs, y)
```

After the sound has been played (using the same function as the Pulse-Doppler radar) and recorded, the recorded signal undergoes a bandpass filter to get rid of static noises and clutter. The bandpass filter has cut-off frequencies 7.5 % lower and higher than that of the frequency on the tone generated. A bandpass filter lower- and higher cut-off frequencies for an 8000 Hz tone would be [7600, 8400] Hz respectively. The filter is designed every request by the radar with the `scipy.signal.firwin()` function and implemented using the `scipy.signal.convolve()` function.

Spectrogram

A spectrogram is a representation of the spectrum of frequencies in a signal. The matplotlib.pyplot package is used to generate the spectrogram by running the following code.

```
Pxx, freqs, bins, im = plt.specgram(y, NFFT=16384, Fs=fs, noverlap=8092)
```

The `Pxx` is the power spectral density, `freqs` is the spectrum of frequencies, `bins` is the amount time that has passed, and `im` is the image file. Figure 4.28 shows the entire spectrum after the bandpass filter has been applied. `NFFT` is the number of slow time FFTs that are being done by the function to form the spectrogram. The frequency resolution depends on it. `noverlap` is the time resolution of the spectrogram. Figure 4.29 shows the zoomed-in version of the same spectrogram centred around the tone frequency. It is visible to see that the recorded sound features the tone frequency.

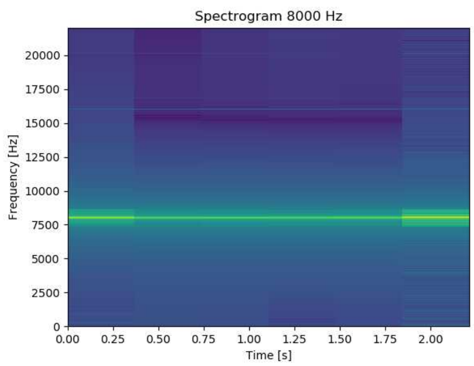

Figure 4.28: Spectrogram Displaying All Registered Frequencies

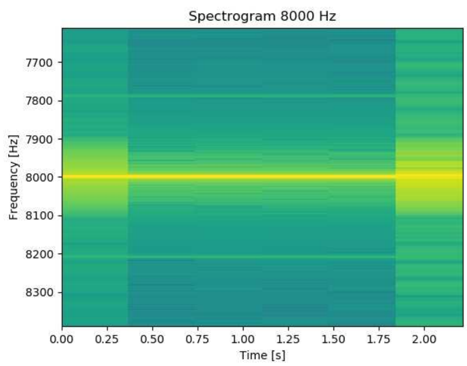

Figure 4.29: Spectrogram Displaying Only Around Tone Frequency

The implemented spectrogram in the radar features a notch filter to get rid of the stationary clutter contained in the received signal. It is not the prospect of the CW radar to display the tone frequency but rather display any Doppler Frequencies recorded by the microphone. This Doppler Frequency represents the target moving. If no movement is detected, only the tone frequency would exist and that is not desirable. Figure 4.30 shows a box moving toward the radar with the notch filter applied. It is important to note that the ambient noises are amplified and with the filters applied, softer noises register more profoundly on the scale and are, therefore, more pronounced.

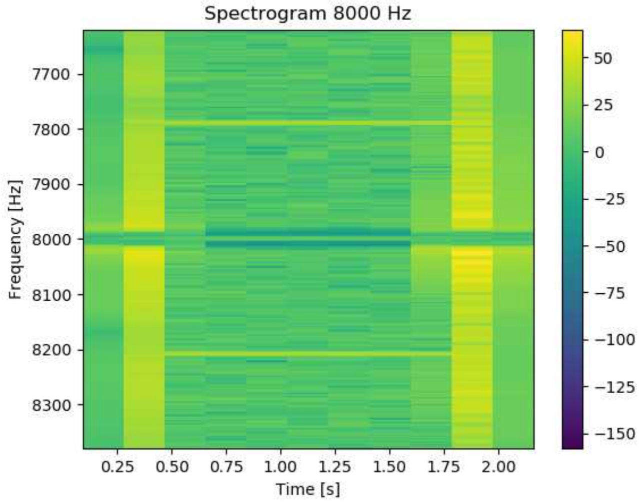

Figure 4.30: Spectrogram with notch filter applied and no movement.

The frequency as seen in Figure 4.30 in the third bin is a higher frequency than that of the played tone of 8000 Hz indicating that the box is moving toward the radar. The observed frequency is roughly 8030 Hz and using Equation 3.5, the velocity is approximately:

$$8030 = \frac{343}{343 - v_s}(8000)v_s = 1.28\ m.s^{-1} \tag{4.3}$$

4.5 Implementation of Chosen Hardware Designs

Using all of the chosen subsystems selected above, the following implementation of the design was chosen and can be seen in Figure 4.31.

It can be seen that the MAX98357A module is connected to the I^2S pins hosted on the Raspberry Pi 3B and directly driving the speaker. On the receiving side, the MAX9814 is connected to a 3.5 mm audio socket that connects to the USB sound card. A capacitor is added to the OUT pin of the microphone to act as a filter and block any DC current to flow. This allows for the input to the sound card to only see the changes in voltage as generated by the microphone. The Veroboard designed to accommodate this circuit can be seen implemented in Figure 4.32.

Figures 4.33, 4.34, and 4.35 show the practical implementation with the manufactured enclosure as designed in Section 4.2.5 combined with the circuitry layout from above. The enclosure was made using medium-density fibreboard (MDF) to add structural integrity and also protection from the outside world. The cost of the MDF amounted to R 50.00 which included two A4 sheets.

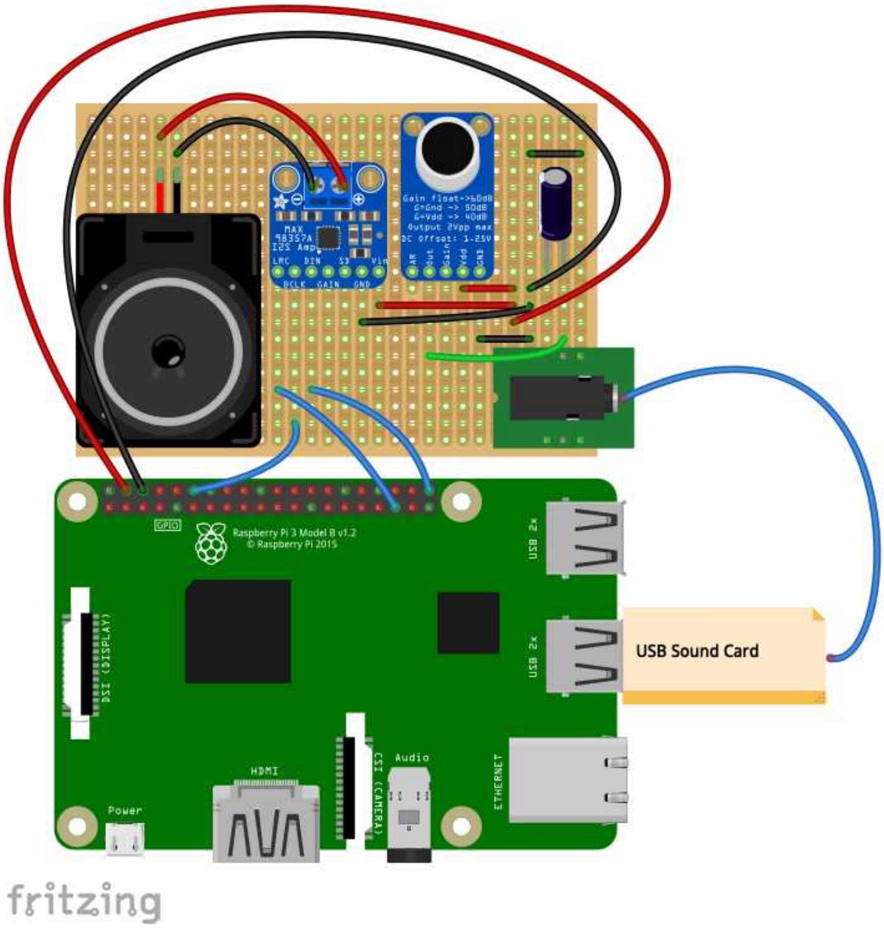

Figure 4.31: Enclosure with components visible

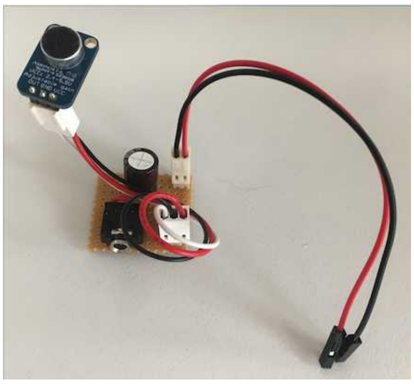

Figure 4.32: Veroboard implementation of microphone and 3.5 *mm* socket

The QR-codes on the enclosure is for connecting to the Wi-Fi as broadcast by the Raspberry Pi and for a quick link to visit the web page hosted on the Raspberry Pi. From the web page, all of the parameters for both methods of detection can be set and the results displayed.

Figure 4.33: Front Facing View of Enclosure

Figure 4.34: Rear Facing View of Enclosure

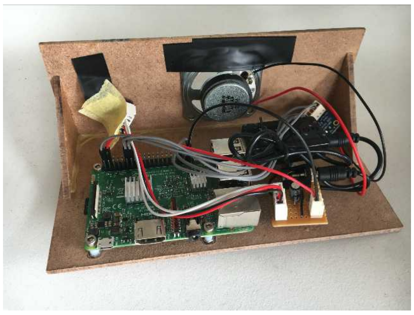

Figure 4.35: Rear Facing View of Enclosure (Open)

4.6 Energy Budget

The following energy budget has been set up as the 'worst-case' possible for all of the components. The table can be seen in Table 4.6.

Item	Criteria	Energy Consumption
Raspberry Pi 3B	400% CPU	730 mA
I^2S **DAC**	Playing at highest volume	360 mA
Speaker	Playing at highest volume	200 mA
USB Sound Card	Only recording	110 mA
Microphone	Quiet environment (maximum gain)	105 mA
		1505 mA Total

Table 4.6: Energy Budget

Chapter 5

Experimentation, Results and Discussion

To prove the concept of an educational, embedded, audio radar, the system as a whole needed to be tested in situations where the results can be useful, not only to the user but also to future projects. The two functionalities of the radar were tested not under lab conditions but real-world situations.

5.1 Continuous Wave

5.1.1 Test 1

Setup

The radar was set up next to a busy road (Marine Drive, Milnerton) on a pedestrian caution sign facing the traffic at an angle of $20°$. The tone was set to $7000\ Hz$ and the duration of $3\ s$. The radar was powered by a power bank and the web interface controlled using an iPad. Figure 5.1 shows the setup next to the road.

Figure 5.1: Setup of CW Radar

Result

The results of the tests were not as successful as originally hoped for. The velocity of the vehicles is roughly 60 km/h, since the speed limit on this section is 60 km/h, which translates to 16.67 $m.s^{-1}$ and furthermore equates to a Doppler Frequency of approaching vehicles to be 8408.7 Hz using Equation 3.5. The Doppler Frequency, and hence the velocity, is at an angle and the radial velocity toward the radar can be calculated using trigonometry. The radial velocity is calculated using Equation 5.1. The results obtained for this setup can be seen in Figure 5.2. The resulting spectrograms for two vehicles following each other can be seen in Figures 5.3 and 5.4.

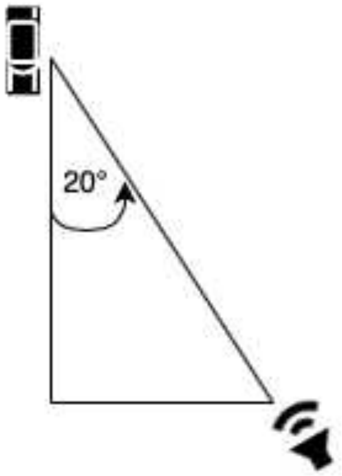

Figure 5.2: Trigonometry for Velocity Measurement

$$v_{actual} = v_{measured}\, cos(20^{\circ}) \qquad (5.1)$$

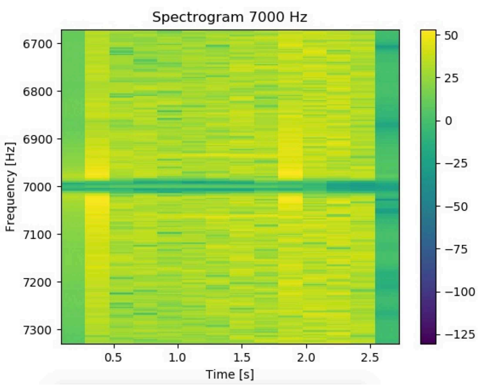

Figure 5.3: Spectrogram of Cars driving CW Test 1

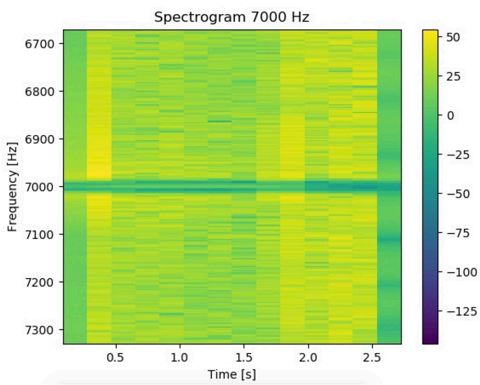

Figure 5.4: Spectrogram of Cars driving CW Test 2

Discussion

As shown above, the results are not as clear as initially intended. Figure 5.3 shows a faint line around 7200 Hz which translates to a rough estimate of the real velocity of 36 km/h. The line is decreasing in frequency and speaks to the fact that cars are speeding up as they accelerate from the traffic light down the road.

Figure 5.4 however, shows only faint lines in the spectrogram with no significant features. This can be attributed to the loud ambient noises surrounding the radar including car engines, nearby construction and the automatic gain of the microphone highlighting even the quietest noise at a distance.

5.1.2 Test 2

Setup

The second test performed using the continuous wave radar was measuring the velocity of a golf club as it hits the ball off of a tee. The radar was placed 1 m behind the tee and using a frequency of 7000 Hz and 9000 Hz and duration of 3 s to capture both the backswing and forward swing of a single stroke. The setup of the test can be seen in Figure 5.5.

Figure 5.5: Setup of radar behind golf ball

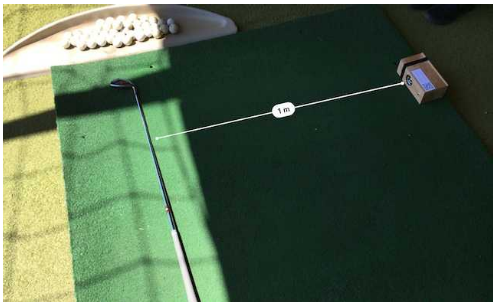

Figure 5.6: Radar 1 m behind the golf ball

Result

Figures 5.7 and 5.8 shows the results from the test described above. The results show two different attempts. Figure 5.7 shows the results obtained with 9000 Hz and Figure 5.8 shows the results as obtained with 7000 Hz.

Discussion

As shown in the results for this test, the resolution of the spectrogram proved to be difficult to read and make out exactly the velocity of the golf club. However, the backswing of the club is visible in the sixth bin and the forward hitting swing is also clearly visible toward the end of the

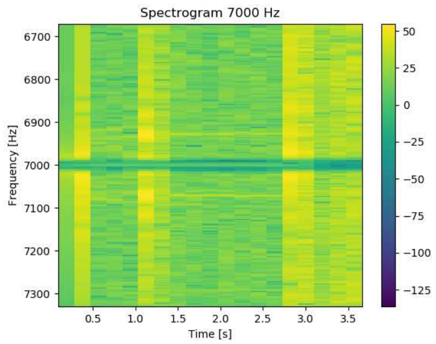

Figure 5.7: CW Results of Golf club velocity 1

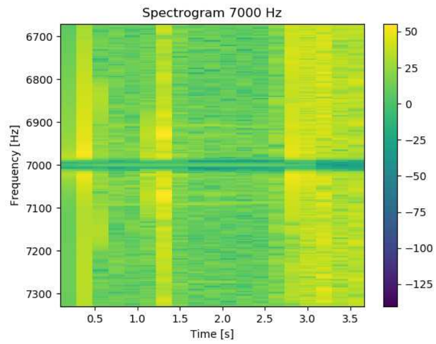

Figure 5.8: CW Results of Golf club velocity 2

spectrogram. The exact measurement of the club speed is not possible though. This can be due to the small size and small window that the golf club is within the beamwidth of the microphone.

5.2 Pulsed-Doppler

5.2.1 Test 1

Setup

The radar was set up on top of a stool approximately 1.5 m tall in the middle of a room on the top floor above a standard 2 vehicle garage. The dimensions of the room are 6 m x 7.2 m with a floor space of 43.2 m^2 and feature a staircase coming up from below. The floor plan can be seen in Figure 5.9.

Figure 5.10 shows the physical location of the radar.

Result

The test was conducted four times, once to each flat part of a wall. This should be enough to map the dimensions of a simple rectangular room. The results in Figures 5.11 and 5.12 only shows the distance to the northern wall and the Western wall. The remaining results can be seen in Appendix C.2.

Discussion

The wall was measured to be approximately 3 m from the radar to the Northern wall and approximately 3.5 m to the Western wall. This corresponds to the floor plan and the measurements

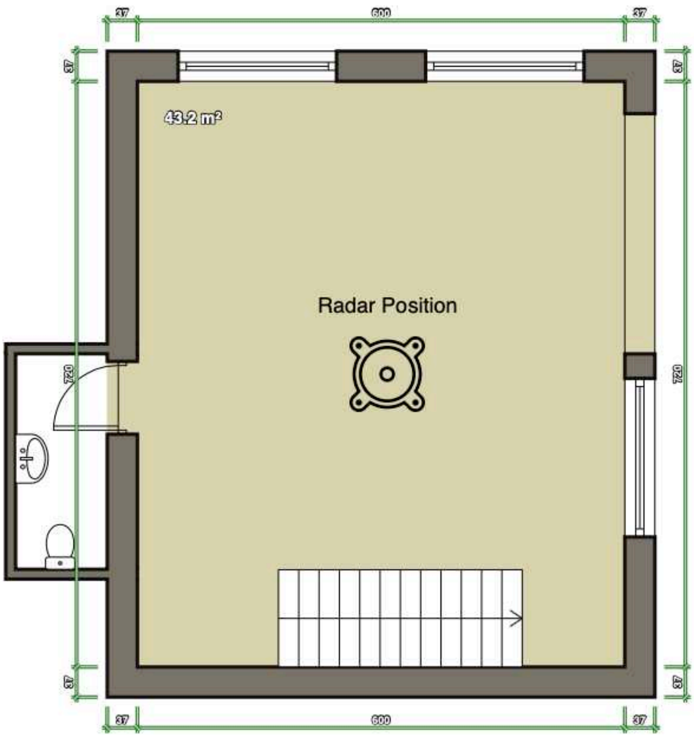

Figure 5.9: Floorplan setup for PD Radar Test 1

Figure 5.10: Radar setup for PD Radar Test 1

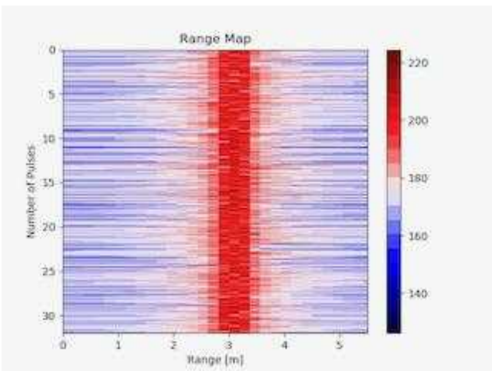

Figure 5.11: Measurement to the Northern Wall

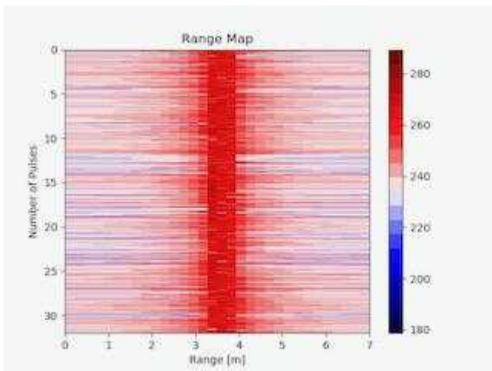

Figure 5.12: Measurement to the Western Wall

proved to be somewhat accurate although the range map shows quite thick amplification and an exact measurement proved to be difficult.

5.2.2 Test 2

Setup

The radar was set up on a bridge to measure the distance to the water. Figure 5.13 shows the bridge and Figure 5.14 the location of the radar on the bridge. The radar was handheld and operated using an iPad. The non-technical implementation of the radar was used with a maximum unambiguous range of 9 m and a volume of 100%.

Figure 5.13: Setup of Radar on bridge

Figure 5.14: Location of Radar on bridge

Result

The results of the process laid out above, the measurement was taken and the results can be seen in Figure 5.15.

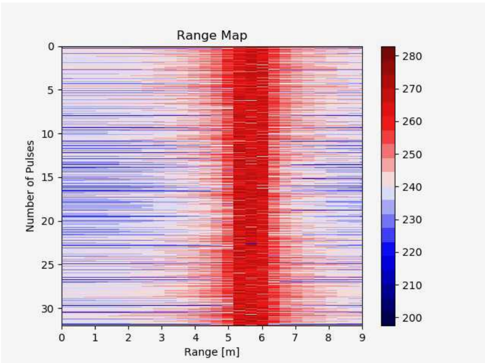

Figure 5.15: Distance to water measurement

Discussion

The results from above clearly show a range of $5 - 6\ m$. Once again, the resolution is not great and accurate measurement can not be obtained. The bridge was quite noisy since it is used by cars and the nearby construction on the new bridge made taking measurements difficult.

5.3 Consistency Tests

Consistency tests were conducted to verify that running the tests under the same conditions and the same targets would yield the same results.

5.3.1 Continuous Wave Radar

The setup was a box moving toward, back and forward again to the radar at high speed ($0.5\ s$ per movement over $1\ m$, i.e. $2\ m.s^{-1}$), the duration of $1.5\ s$ and a tone of the frequency of $10\ 000\ Hz$. The setup of the test can be seen in Figure 5.16.

Figure 5.16: Setup of Radar on box and corner reflector for consistency tests

Three tests were performed and the results are shown in Figures 5.17, 5.18 and 5.19.

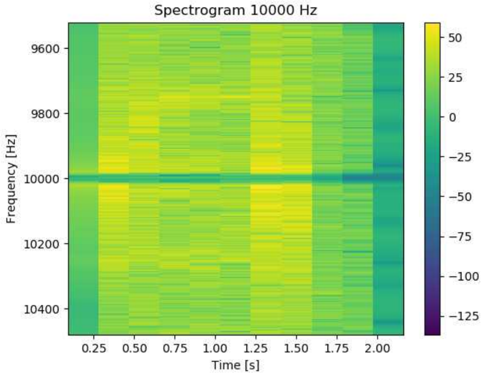

Figure 5.17: Front Facing View of Enclosure

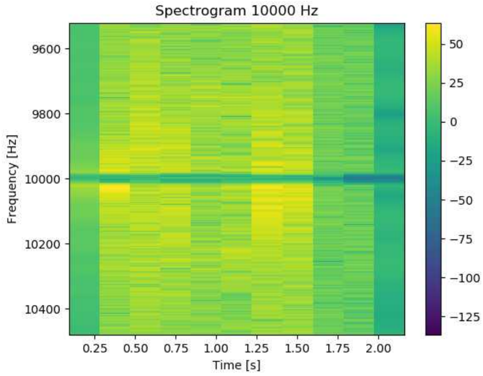

Figure 5.18: Rear Facing View of Enclosure

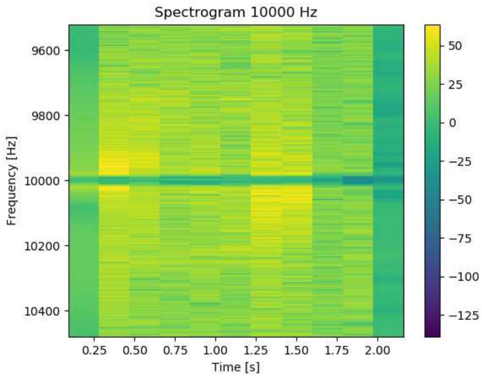

Figure 5.19: Rear Facing View of Enclosure (Open)

5.3.2 Pulsed-Doppler Radar

The setup was the radar hand-held at 1 m away from a wall inside a parking lot with high ceiling and no nearby walls to minimise unwanted noise reflections. The parking garage was also selected so that minimal wind noise would impact the results. The setup of the test can be seen in Figures 5.20 and 5.21.

Three tests were performed and the results are shown in Figures 5.22, 5.23 and 5.24.

5.3.3 Discussion

As can be seen from the consistency checks performed above, the results are similar for both the continuous wave radar and the Pulsed-Doppler radar.

The continuous wave radar is difficult to see exactly the velocity of the box moving toward and backwards from the radar but it differs significantly from a test where no movement has happened such as in Figure 4.30. The results show similar movements and disturbances to the spectrogram. Therefore, the radar performed consistently over the three tests.

Figure 5.20: View of the Radar to the wall

Figure 5.21: Distance measurement to the wall (7.39 m)

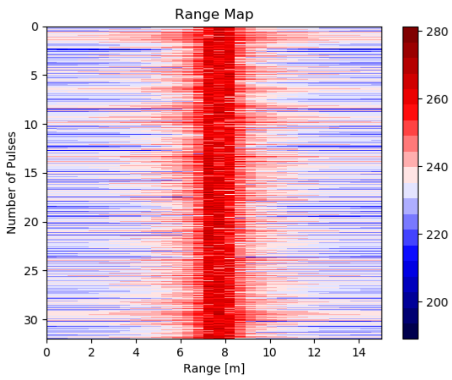

Figure 5.22: Pulsed-Doppler Consistency Test 1

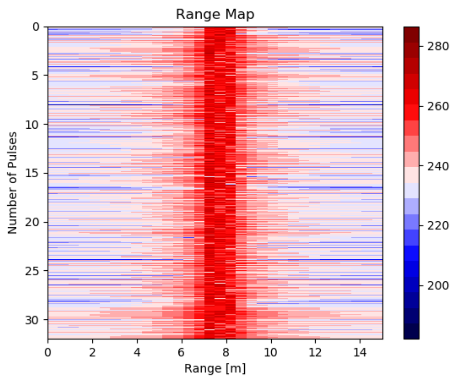

Figure 5.23: Pulsed-Doppler Consistency Test 2

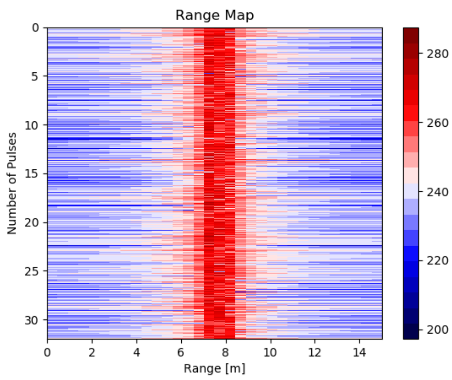

Figure 5.24: Pulsed-Doppler Consistency Test 3

The Pulsed-Doppler radar tests all yielded similar results where the wall is recognised with similar returned power for all three tests. The leakage power that registers on the Range Map can be attributed to the parking garage reflections of sound although most precautions were taken to minimise the unwanted effects.

Chapter 6

Conclusions

After the results and discussion have been shown in Chapter 5, the following conclusions can be drawn.

6.1 Continuous Wave Radar

The tests conducted using the continuous wave radar to measure velocity proved to be extremely tough. The microphone with automatic gain picked up the quietest of noises and when the sound that is intended to be recorded presents itself, the MAX9814 chip is just too slow to respond. This was proven by using an over the counter microphone with the ADC and the spectrogram would improve but the results were so quiet that they were barely recognisable and not usable outdoors. The set up was under laboratory conditions with no ambient noises.

Another major issue that the radar experienced was the poor resolution that was offered by the spectrogram by the Matplotlib package. The package allows for setting the number of FFTs that the function performs for a single spectrogram and this setting was set to 16384 and the resolution did not improve dramatically. This was proven by using a over the counter microphone with the ADC and the spectrogram did not improve overall. A different package was also tested and `scipy.signal.fftpack` offered worse results.

Overall, the radar does display some correlation to the real world where a significant increase in activity is visible to the end-user. However, the actual real velocity of the measured target is not clear.

6.2 Pulsed-Doppler Radar

The tests conducted using the Pulsed-Doppler radar to measure the distance or range to a target proved to be a relative success. The distances in the results were obtained successfully although a lot of prior set up was required and the radar did not yield the results that were hoped for immediately.

The reasons for the difficult operation of the Pulsed-Doppler radar were because of the sensitivity of the microphone once again. The room had to be completely quiet and even then the radar had issues. In the second test where the water height was measured, the results were obtained after numerous tests were run.

The radar produced acceptable results in this aspect but some room for improvement still exists.

6.3 Consistency Tests

The checks run to test whether the radar produced similar results, were largely successful. The radar was set up and without interference between tests in a quiet room, the tests were run. The results from all of the tests were very similar and proved that the radar can produce consistent results.

6.4 Overall

The project was a success in the eyes of the objectives originally set out in the Introduction (Chapter 1).

The project produced a simple, easy to use and informative radar system that is indeed very accessible where all the components were off-the-shelf and done in a reasonably low budget of R 1500. The radar is also a well packaged and compact design making it portable and easily usable in open days and information sessions.

The radar is reproducible in the sense that anyone with limited prior knowledge of electronics and embedded systems can reproduce this radar with little issues. Interest in radar systems can easily be realised by the project since the audio component of the radar makes the results more tangible, fun to play with and experiment with remote sensing techniques.

Chapter 7

Recommendations

Following the conclusion of the project, the following recommendations can be made where future projects can benefit and improve on the educational, embedded, audio radar.

1. Use a microphone with a greater sensitivity to quieter sounds without using automatic gain.
2. Improve on using the USB sound card by researching audio-specific, Raspberry Pi compatible, ADCs.
3. Improve the signal processing algorithms to improve speed when executing filters and FFTs.
4. Investigate the use of higher or lower frequencies in the audible spectrum for implementation to improve range and accuracy.
5. Investigate the usage of Notch, Band Pass, Low-Pass and High-Pass filters to improve their effectiveness in isolating movement in the Continuous Wave radar mode.
6. Develop printed circuit boards for all of the Veroboard circuits to improve the physical appearance and functionality of the radar.
7. Investigate other embedded platforms, like the Raspberry Pi 4B with up to 4GB RAM, to do the signal processing and improve overall speed.
8. Explore techniques used to develop a spectrogram function other than a build-in package's version to improve the resolution of the spectrogram.
9. Investigate how to suppress signals of power less than a certain value in different sound level conditions to improve the Range-Doppler Map.
10. Improve the webserver to display more interactive content with the specific mode's results that is showing on the results page.
11. Change the design to incorporate an Ethernet CAT5 cable to be internet-connected to the Raspberry Pi with the audio socket circuits fixed down. This would enable testing the radar while protecting the components.

Appendix A

RadarPi Procedures

A.1 Set-Up

The instructions followed in this section serves as a tutorial or a guide to assemble the Audio Radar in its entirety. It will follow initial setup of test equipment used, assembly of the transmitting and receiving hardware and finally the software component of the project will be addressed.

A.1.1 Raspberry Pi Setup

Complete setup instructions exist in many forms online however, this guide will serve the setup requirements needed only for the Audio Radar. Other functionality of the Raspberry Pi are ignored but it should be noted that it is not limited to only the functions that are used here.

It is assumed that the following is available to the user:

- Raspberry Pi 3 or newer[1] with Raspbian with Desktop already installed and SSH enabled[2].
- A computer with keyboard, mouse, monitor and SD card slot
- An Ethernet cable
- A local area network with Ethernet communication and WiFi.

The setup of the Raspberry Pi is covered extensively on the internet. **Here** is a guide to set up the Raspberry Pi.

A.1.2 Package Installation and Setup

The following packages need to be installed:

- Flask (**Link**)
- PyGame (**Link**)

[1] Models older than the Raspberry Pi 3 will also work but a USB WiFi dongle will be required.
[2] Numerous online resources exist to achieve this such as this.

- I^2S DAC Driver (**Link**)
- RaspAP (**Link**)
- If the radar is implemented using something other than the Raspberry Pi, additional packages need to be installed:
  - Numpy (**Link**)
  - Scipy (**Link**)
  - Matplotlib (**Link**)

A.1.3 Hardware Setup

The following links would help with setting up all the hardware needed for the radar.

- Microphone setup (**Link**)
- Speaker setup (**Link**)

A.1.4 Procedure

Clone repository that is linked in Appendix B and copy contents to `/home/pi/Documents/`Set up script to run `webapp.py` at start up of Raspberry Pi (**Link**).

A.2 Usage Tutorial

The radar was designed to be as simple as possible to use. Therefore, after initial set-up, the only instructions on how to use the radar are as follows:

1. Power the Raspberry Pi with a minimum 1.5 A power supply.
2. Wait for 10 seconds until the Wi-Fi network 'RadarPi' is registered.
3. Scan the QR-code with your device or connect to the Wi-Fi manually.
4. Scan the other QR-code to visit the web page or enter the web address manually.
5. Select a mode that you want to use i.e. Continuous Wave or Pulsed-Doppler. Both have a technical and non-technical option. The technical version is intended only for advanced users.
6. Enter the parameters requested by the web page.
7. Click 'submit' and wait for the resulting figure to appear. The Pulsed-Doppler has significant operations and may take longer than 8 s.
8. Click on the 'Home' button to return to the landing page or select another mode from the navigation bar.
9. When done using the RadarPi, simply disconnect the power cable from the Raspberry Pi.
10. If any issues arise, the GitHub page will remain active, where questions can be asked, as this project will continue with further developments after the conclusion of the course.

Appendix B

Code

Code used for testing and creation of RadarPi: https://github.com/DewanPieterse/AudioRadar

The GitHub page containing all code to set-up and run RadarPi: https://github.com/DewanPieterse/RadarPi

Appendix C

Testing Procedures and Results

C.1 Testing Set-Up and Testing Results

All of the recorded data sets to verify the testing of the microphones and speakers can be found along with all the screenshots and photos taken, here.

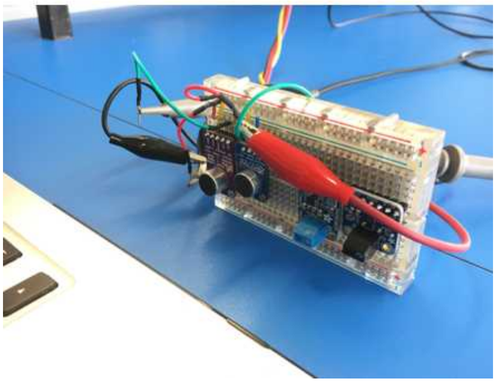

Figure C.1: Microphone Set-Up

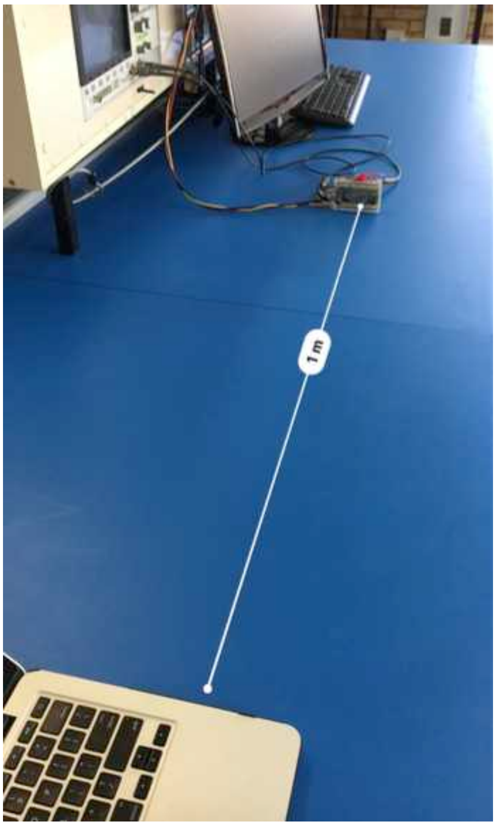

Figure C.2: Distance Measurement

Figure C.3: Microphone Suspended

Figure C.4: Initial gain vs frequencies: Microphones

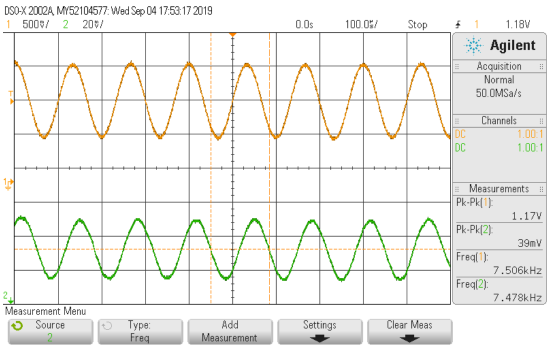

Figure C.5: Sample of Oscilloscope view when testing Speakers

C.2 Experimentation Results

Results from Pulse Doppler Radar mapping a room as from Section 5.3.2

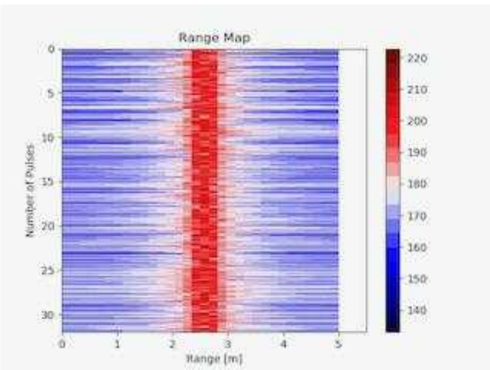

Figure C.6: Measurement to Eastern wall in PD Test 1

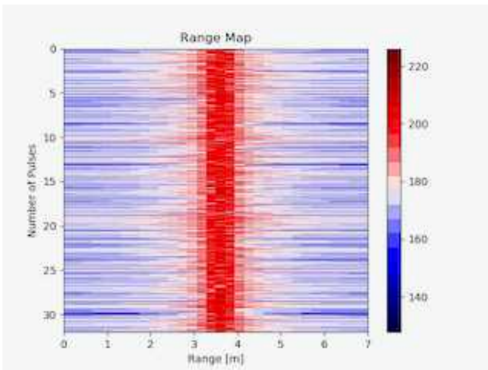

Figure C.7: Measurement to Southern wall in PD Test 1

References

[1] G. L. Charvat, A. J. Fenn, and B. T. Perry, "The MIT IAP radar course: Build a small radar system capable of sensing range, Doppler, and synthetic aperture (SAR) imaging," in *2012 IEEE Radar Conference*. Atlanta, GA: IEEE, May 2012, pp. 0138–0144. [Online]. Available: http://ieeexplore.ieee.org/document/6212126/

[2] "uRad Technology Brochure," 2018. [Online]. Available: https://urad.es/download/uRAD%20-%20Brochure%20EN.pdf

[3] D. Tarchi, M. Vespe, C. Gioia, F. Sermi, V. Kyovtorov, and G. Guglieri, "Low-Cost Mini Radar: Design Prototyping and Tests," *Journal of Sensors*, vol. 2017, pp. 1–15, 2017.

[4] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of modern radar*. Scitech, 2010.

[5] "Radar Basics." [Online]. Available: http://www.radartutorial.eu/11.coherent/co06.en.html

[6] "Matched Filtering - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/help/phased/ug/matched-filtering.html

[7] "Software Engineering | SDLC V-Model," Jul. 2018. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

[8] "MAX98357a Datasheet," 2019. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3006/MAX98357A-MAX98357B.pdf

[9] "MCP3008 - 8-Channel 10-Bit ADC With SPI Interface." [Online]. Available: https://thepihut.com/products/adafruit-mcp3008-8-channel-10-bit-adc-with-spi-interface

[10] R. C. Watson, *Radar origins worldwide: history of its evolution in 13 nations through World War II*. S.l.: Trafford Publ, 2009, oCLC: 951267177.

[11] "uRad Technology White paper," 2018. [Online]. Available: https://urad.es/download/uRAD%20-%20White%20paper%20EN.pdf

[12] "Raymarine Pack." [Online]. Available: https://bit.ly/2omCSSm

[13] G. L. Charvat, "MIT IAP 2011 Laptop Based Radar: Block Diagram, Schematics, Bill of Material, and Fabrication Instructions*," MIT Lincoln Laboratory, Jan. 2011. [Online]. Available: https://bit.ly/2nJ9xS2

[14] "525.648—Introduction to Radar Systems Course Homepage." [Online]. Available: https://apps.ep.jhu.edu/course-homepages/2928-525.648-introduction-to-radar-systems-farthing

[15] C.-S. Lin, "Design of Audio Radar for hand-gesture recognition," Final Year Project, University of Cape Town, Cape Town, 2018.

[16] R. Ali Yasin, "Design and Implementation of an Ultrasonic Radar for Hand Gesture Recognition," Final Year Project, University of Cape Town, Cape Town, 2018.

[17] "Microwave Journal - uRad," 2019. [Online]. Available: https://www.microwavejournal.com/articles/31762-ghz-radar-works-with-arduino-and-raspberry-pi

[18] "Radar Basics - Maximum Unambiguous Range." [Online]. Available: http://www.radartutorial.eu/01.basics/Maximum%20Unambiguous%20Range.en.html

[19] "Radar Basics - Range Resolution." [Online]. Available: http://www.radartutorial.eu/01.basics/Range%20Resolution.en.html

[20] M. Viswanathan, "Chirp Signal – Frequency Sweeping – FFT and power spectral density – GaussianWaves," 2019. [Online]. Available: https://www.gaussianwaves.com/2014/07/chirp-signal-frequency-sweeping-fft-and-power-spectral-density/

[21] "Continuous Wave Radar," 2019. [Online]. Available: https://fas.org/man/dod-101/navy/docs/es310/cwradar/cwradar.htm

[22] G. Heinzel, A. Rüdiger, and R. Schilling, "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows," *Max Plank Inst*, vol. 12, 2002.

[23] Chip, "Open Audio: Benchmarking - Teensy 3.6 is Fast!" Oct. 2016. [Online]. Available: https://openaudio.blogspot.com/2016/10/benchmarking-teensy-36-is-fast.html

[24] "Teensy LC (Low Cost)." [Online]. Available: https://www.pjrc.com/teensy/techspecs.html

[25] "Buy a Raspberry Pi 3 Model B – Raspberry Pi." [Online]. Available: https://www.raspberrypi.org

[26] "What is the power consumption of the Teensy 3.6?" [Online]. Available: https://forum.pjrc.com/threads/47256-What-is-the-power-consumption-of-the-Teensy-3-6

[27] "Power Consumption Benchmarks | Raspberry Pi Dramble." [Online]. Available: https://www.pidramble.com/wiki/benchmarks/power-consumption

[28] A. Industries, "Electret Microphone Amplifier - MAX4466 with Adjustable Gain." [Online]. Available: https://www.adafruit.com/product/1063

[29] "Adafruit MAX98357 I2s Class-D Mono Amp." [Online]. Available: https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview

[30] H. Landau, "Sampling, data transmission, and the Nyquist rate," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967. [Online]. Available: http://ieeexplore.ieee.org/document/1447892/

[31] G. Dwyer, "Flask vs. Django: Why Flask Might Be Better | Codementor." [Online]. Available: https://www.codementor.io/garethdwyer/flask-vs-django-why-flask-might-be-better-4xs7mdf8v