UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Software Defined Radio Based Avalanche Beacon Receiver.

Vetle Hofsøy-Woie

Master project in Computer Science - INF-3981 - January 2024

Submitted on January 25, 2024

This page is intentionally left blank

“Once or twice though you should fail, If you would at last prevail, Try again.”
–Edward Hickson, *The Singing Master*, (1836)

Abstract

Annually, avalanches claim an average of 100 lives, and many more are injured. For victims buried by an avalanche, time is of the essence. If not rescued within 15 minutes, the victim only has a 10% probability of survival.

Commonly, the people venturing into avalanche-prone areas equips avalanche beacons. These devices work as small radio transmitters transmitting a signal at 457 kHz once each second. Another avalanche beacon can then receive the signal and give the rescuer an indication of the buried victim's direction and distance.

In this thesis, I present a novel approach to automatically detect and measure the received power of a transmitting avalanche beacon. The approach uses a software-defined radio and a manually created antenna to sample radio signals. The created antenna is tuned to be optimal around 457 kHz.

In addition to the hardware for sampling the radio spectrum, I present three algorithms for analyzing the received radio signal. The first algorithm performs the entire analysis in the time domain, the second algorithm works in the frequency domain, and the third algorithm utilizes both the frequency and time domains of the signal.

The experiments show that the hardware and the algorithms can automatically detect a transmitting avalanche beacon from at least 20 meters away, possibly further.

Acknowledgements

I extend my heartfelt gratitude to Sveinung Olsen from UiT for his exceptional support and expertise in developing the antennas utilized in this project. His invaluable contributions have played an indispensable role in this project.

I would like to express my profound appreciation to Juha Vierinen for his remarkable contributions to this project. Your invaluable assistance in not only conceiving this idea but also guiding me throughout the entire project has been truly invaluable. Juha Vierinen's support and expertise have played a pivotal role in shaping the success of this endeavor, and I am immensely grateful for his invaluable contributions.

Contents

List of Figures

List of Tables

1

Introduction

Each year an average of 100 people are killed by avalanches[8]. In Norway alone, seven people are killed and another 67 injured in avalanche accidents on average each year[18]. Time is of the utmost importance for victims that are not killed by trauma but are buried by the snow. An avalanche victim has around 90% probability of survival if he/she is rescued within 15 minutes. After the first 15 minutes, the survival rate quickly drops to less than 60% at 20 minutes and to around 30% at 30 minutes[21]. Figure 1.1 shows the survival rate of buried victims depending on the duration of the burial. Even lucky victims that manage to survive 20-30 minutes might have lasting brain damage after as little as ten minutes of burial. Therefore, a successful rescue operation is generally accomplished quickly, usually quantified in 15 minutes or less.

There are two main phases in a typical avalanche rescue scenario: localization and excavation of the victim. The duration of the excavation phase is determined by the depth at which the avalanche buried the victim and the texture of the snow in the avalanche. Commonly, it is said that a rescuer needs 5-10 minutes per meter of snow lying on top of the victim. Therefore, if the buried victim lies under 1-2 meters of snow, the localization phase of the rescue should take less than 5 minutes.

The time needed for the localization phase is mainly determined by how far the rescuer is from the victim when the rescue operation starts and the nature of the terrain the rescuer will have to move in during the search.

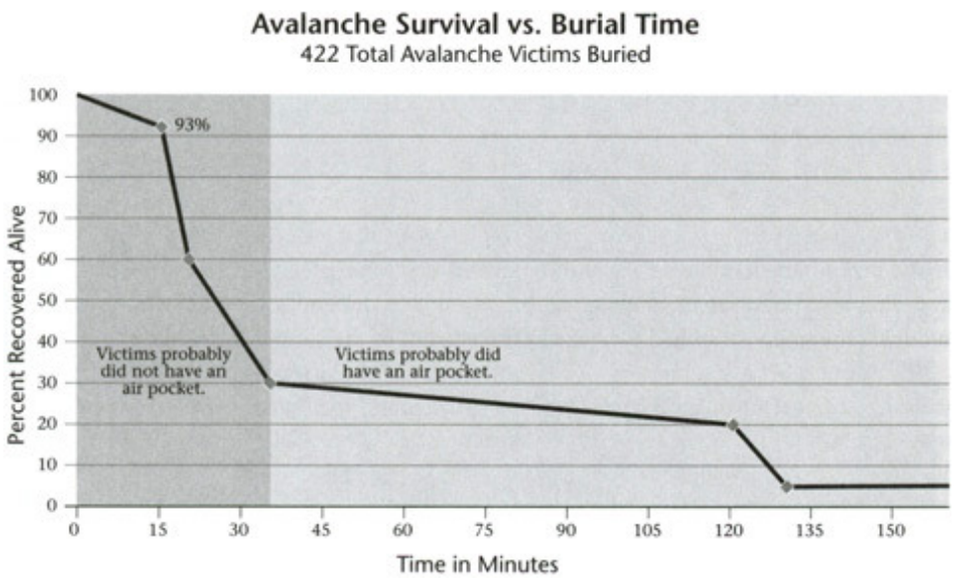


Figure 1.1: Avalanche Survival vs. Burial time, From Ch.1 Tremper[21]. The original caption reads: A generalized graph of European avalanche victims who were completely buried and in total contact with the snow (no people in vehicles or houses). Half of the victims are dead within 25 minutes. This graph does not include victims killed by trauma, which account for about a quarter of avalanche deaths

Currently, the localization phase of the rescue is aided by small wearable radio antennas known as avalanche beacons. Personnel venturing into avalanche areas typically wear these beacons beneath their clothing. These beacons have two use cases; in their default state, they transmit a radio signal each second, and in the case of a rescue, the beacons can be used as radio receivers, showing the direction and distance to another transmitting beacon.

In recent years, methods have been proposed to decrease the localization phase duration. These methods utilize small unmanned drones fitted with avalanche beacons to aid in the search. The main benefit of using drones for the localization phase is that they are not slowed down by the rugged terrain found in an avalanche accident. However, the use of these schemes is currently limited to academic research and is not in practical use.

This thesis introduces a novel digital avalanche beacon receiver that employs software-defined radios and digital signal processing techniques. The study delves into developing and constructing a hardware prototype that can receive and store radio signals. In addition, it proposes three algorithms for detecting whether a transmitting avalanche beacon was nearby at the time of receiving.

The thesis also presents experiments to test the prototype's validity and the three algorithms. The experiments were done in a controlled environment and showed the algorithm's ability to detect the presence of an avalanche beacon from at least 20 meters away.

1.1 Problem statement

This thesis considers the problem of searching for and rescuing victims of avalanche accidents. Such a rescue usually consists of two phases: a localization phase and an excavation phase. The thesis's primary focus is the localization phase of a rescue. More specifically, this thesis limits itself to the digital searching equipment used in such rescue scenarios.

Many factors are of the utmost importance when designing a rescue device before personnel can use the device in real-world scenarios. These factors include, but are not limited to, usability, dependability, and robustness to harsh weather climates. These issues, though essential, fall outside the domain of computer science. Therefore, this thesis limits itself to designing and creating a digital radio to detect nearby avalanche beacons. In this thesis, I attempt to solve this problem by examining the underlying theory of modern avalanche beacons and reviewing related technologies. From my understanding of the theory and the associated technologies, my thesis is that:

> *It is possible to develop a digital avalanche beacon receiver using software-defined radios and digital signal processing techniques, which can detect the presence of other avalanche beacons automatically.*

To support this thesis, I have created a hardware prototype to receive and store radio signals around the specific frequency band of modern avalanche beacons and store these signals for later analysis. In addition, I developed three algorithms that can detect whether an avalanche beacon is transmitting nearby. The prototype allows me to experiment with the system's feasibility to determine its performance and drawbacks. The prototype enables me to make experiments to test the algorithms retrospectively. I also evaluate my system against the range specified by modern avalanche beacons.

1.2 Contributions

The main contributions of this thesis are;

- Algorithms
  - Three algorithms for automatic detection of a transmitting avalanche beacon
    - One algorithm using only time-domain analysis of the signal
    - One algorithm using solely frequency domain analysis of the signal
    - One algorithm using a hybrid of frequency domain and time domain analysis
- Artifacts
  - A System for receiving and storing signal data from avalanche beacons using commercial software-defined radios.
  - Antenna designed for receiving radio signals close to 457kHz, such as those transmitted by avalanche beacons.
- Lessons learned!
  - One can automatically detect the presence of transmitting avalanche beacons using rudimentary digital signal processing techniques.
  - Commercial software-defined radios can receive signals at a high enough quality to detect avalanche beacons transmitting.
  - Improving the quality of antennas is crucial for better signal reception, even with the assistance of digital signal processing techniques.
  - Interference from electrical devices has little impact on a transmitting beacon.
  - Interference from electrical devices can severely impact a receiving beacon. However, this effect quickly degrades as a function of distance.

1.3 Methods and materials

1.3.1 Methodology

According to The Task Force on the Core of Computer Science, there are three main paradigms of computing[5]. These three paradigms are *theory*, *abstraction*, and *design*. Each of these paradigms is characterized by the approach one would take while working within them.

The paradigm of *theory* heavily relies on mathematics and aims to establish a consistent and valid theory. To reach this goal, a mathematician should iterate through the following four steps:

1. Characterize objects of study (definition),
2. Hypothesize possible relationships among them (theorem),
3. Determine whether the relationships are true (proof), and
4. Interpret results.

The next paradigm, *abstraction*, is mainly based upon the scientific method. This paradigm is used when a scientist wants to examine a specific phenomenon. The scientist is then expected to follow these four steps. If the results of the experiment are different than the prediction based on the created model, the scientist should go through the steps again with a new hypothesis and model.

1. Form a hypothesis,
2. Construct a model and make a prediction,
3. Design an experiment and collect data
4. Analyze results.

The last paradigm, *design*, has its roots in engineering. An engineer is expected to iterate through the following four steps when designing a system or device for solving a problem. When the system tests show that the system does not live up to the requirements, the engineer should make necessary changes to the system and start over.

1. State requirements,
2. State specifications,

3. Design and implement the system

4. Test the system.

The paradigm of this project is primarily the third paradigm, *design*, as this project aims to create a proof-of-concept device to aid search and rescue operations during avalanche incidents effectively. However, during the work on this thesis, I also utilized the paradigm *abstraction*, primarily when testing the system. I form hypotheses and design experiments to test these. The requirements and specifications of the device are stated in Table 1.1.

1.3.2 Procedures

This thesis primarily follows the design paradigm from The Task Force on the Core of Computer Science, with minor differences. Firstly, I stated some broad requirements that the final system should be able to comply with. Table 1.1 lists these four requirements. After that, I examined the relevant theory and literature to gain better knowledge about how I could create an artifact that would comply with the requirements.

Once I had a satisfactory understanding of the theory needed, I moved on to the design process. I attempted to design and implement an antenna, a system, and two algorithms that could detect an avalanche beacon. After developing an initial system, I created an experiment to evaluate its strengths and weaknesses concerning the requirements stated at the start of the project. The experiment provided new knowledge and insights, which I used to create a new version of the whole system. Lastly, I designed a new experiment to test the new system concerning the stated requirements.

1.	The beacon should be able to store data for later analysis
2.	The beacon should be small enough to fit on a common unmanned vehicle, such as a drone
3.	The beacon should have a similar search range as today's state-of-the-art avalanche beacons
4.	The algorithms should be able to automatically detect the presence of a transmitting avalanche beacon

Table 1.1: Requirements of the project

1.4 Assumptions and limitations

As mentioned in the problem statement, there are many essential aspects to consider when designing a rescue device. However, the main focus of this thesis is the creation of a software-defined radio-based beacon able to detect the presence of another transmitting beacon automatically. To achieve this goal, I constrained my design and experiments with the following limitations:

- I used only two different commercial avalanche beacons in the experiments. Both are from the same manufacturer.
- I constrained the detection algorithms to detect if there are nearby avalanche beacons, not whether there are several at different locations.
- I conducted the experiments inside a sports arena, where I had no control over the radio interferences that might affect the avalanche beacons.
- Due to place constraints in the sports arena, the experiments only tested a reception range of up to 20 meters.
- I only tested one type of software-defined radio during the receiver's design.
- Although I present the theory of calculating the direction from the receiver to the transmitting beacon based on the relative difference in measured signal strength, this still needs to be implemented and tested.
- As the beacon is supposed to be used during rescue only, there is no need for it to be able to transmit a location signal.

1.5 Structure of thesis

- Chapter 2 Related Theory
  - Chapter 2 presents the theory required to comprehend the solutions presented in this thesis. It will present this thesis's pertinent mathematical formulas and physical concepts.
- Chapter 3 Related technologies
  - Chapter 3 presents the technologies in use today and explains the choice of hardware for this thesis.

- Chapter 4 Related literature
  - Chapter 4 discusses the current advances in the use of technology for avalanche search and rescue and their strengths and weaknesses.
- Chapter 5 Version 1
  - Chapter 5 presents the design and implementation of an initial prototype and the experiments done to understand its shortcomings and areas for improvement.
  - Chapter 5 also presents two algorithms for automatically detecting a transmitting avalanche beacon.
- Chapter 6 Version 2
  - Chapter 6 presents the design and implementation of the second prototype about the shortcomings of Version 1
  - Chapter 6 also presents a new algorithm for detecting a transmitting avalanche beacon and the experiments done to test the new system.
- Chapter 7 Discussion
  - Chapter 7 discusses the project, the shortcomings and limitations of the final product, and the experiments done during the thesis.
- Chapter 8 Conclusion
  - Chapter 8 concludes the project with some final remarks and discusses possible areas for future work.
- Appendix A: Results from experiments 1
  - Appendix A provides the analyzed results from the experiments done in version 1.
- Appendix B: Avalanche dynamics
  - Appendix B gives an overview of avalanche dynamics to provide the reader with a deeper understanding of this thesis's motivation.

2

Related theory

In this section, concise explanations of vital concepts that underlie this master's thesis will be provided. These ideas are crucial in comprehending the research topic and creating a structure for interpreting and analyzing the findings. By understanding these fundamental principles, we can establish a common understanding and conceptual framework that will enhance the understanding of the upcoming chapters. The aim of this section is to provide the reader with a solid foundation in the essential concepts and theoretical foundations, allowing for a meaningful discussion and analysis of the research outcomes.

2.1 Understanding the Functionality of Modern Avalanche Beacons

This project aims to develop a fully digital avalanche beacon receiver. To understand the process better, the reader must have a grasp on the internal workings of existing avalanche beacons. Although there are multiple makes and models of avalanche beacons, most of them share the same basic functions for antenna, signal analysis, and detection.

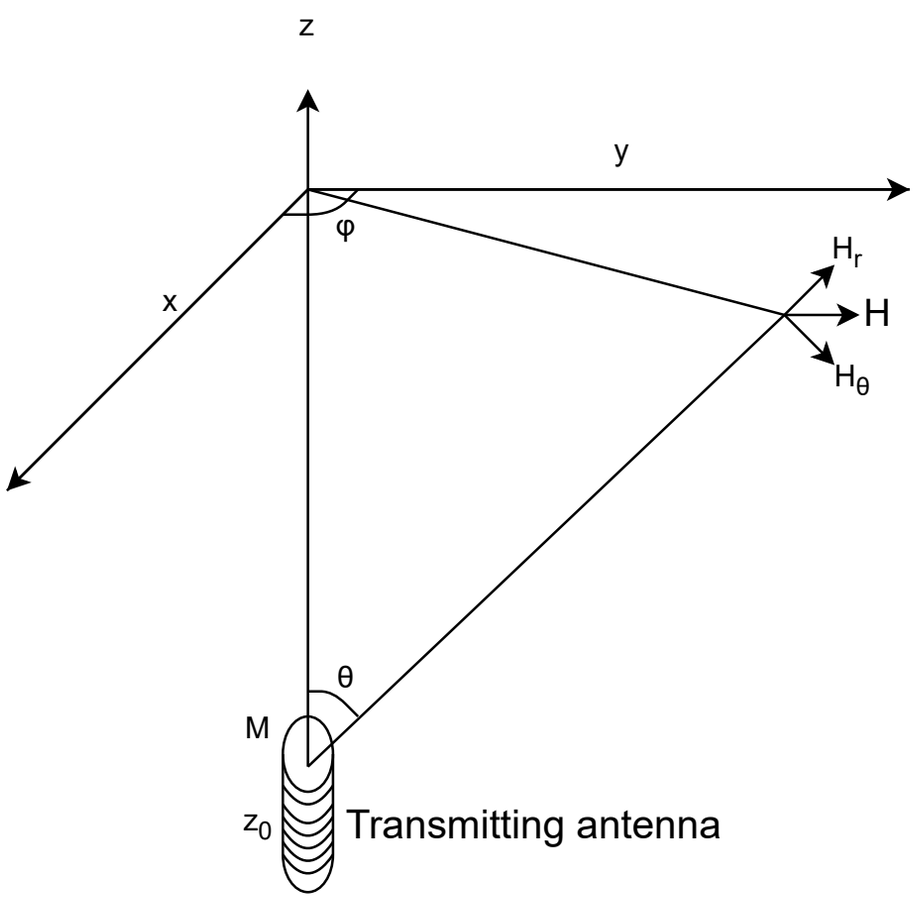


Figure 2.1: This illustration depicts a magnetic dipole antenna, commonly utilized in avalanche beacons, and showcases the magnetic field components surrounding a transmitting antenna. This illustration has been recreated from the illustrations found in Ayuso et al.'s paper on avalanche beacons[2].

2.1.1 Antennas

All avalanche beacons consist of one or more magnetic dipole antennas. Magnetic dipole antennas are compact and efficient devices that radiate electromagnetic waves based on the magnetic field oscillations they generate. These antennas usually consist of a conducting loop around some ferrite material. The length of the antenna is typically much shorter than the wavelength of the signals it receives or transmits. When an alternating current is supplied, the antenna induces a magnetic field H[2]. For modern avalanche beacons, this alternating current has a frequency of $457kHz$. The characteristics of this magnetic field are illustrated by Figure 2.1. When discussing the characteristics of this magnetic field, it is common to consider the surrounding media a vacuum and discard the effects of snow and underlying soil. With these assumptions in mind, the magnetic field intensity can be described in polar coordinates using Equations 2.1 to 2.3, and the magnitude of the field by Equation 2.4.

$$H_r = \frac{M}{2\pi r^3} \cos\theta \tag{2.1}$$

$$H_\theta = \frac{M}{4\pi r^3} \sin\theta \tag{2.2}$$

$$H_\phi = 0 \tag{2.3}$$

$$H = \frac{M}{4\pi r^3} \sqrt{1 + 3\cos^2\theta} \tag{2.4}$$

When the antenna is subjected to a changing magnetic field, a changing current is induced within the conduction loop of the antenna. This current can then be measured and is proportional to the magnitude of the magnetic field around the antenna. However, the sensed magnetic field is also dependent on the orientation of the antenna in relation to the magnetic field, as illustrated in Figure 2.2. The intensity of this sensed magnetic field, $H_{antenna}$, can be described by Equation 2.5.

$$H_{antenna} = H \cdot \cos\alpha \tag{2.5}$$

The older version of avalanche beacons would detect the magnetic field and convert it into an audible tone, with the tone's volume depending on the strength of the magnetic field. The rescuer could then follow the path that resulted in the loudest tone, which would lead them to the victim. Modern avalanche beacons use two or three antennas that are perpendicular to each other during the search process. In cases where there are two antennas, Equation 2.5 describes

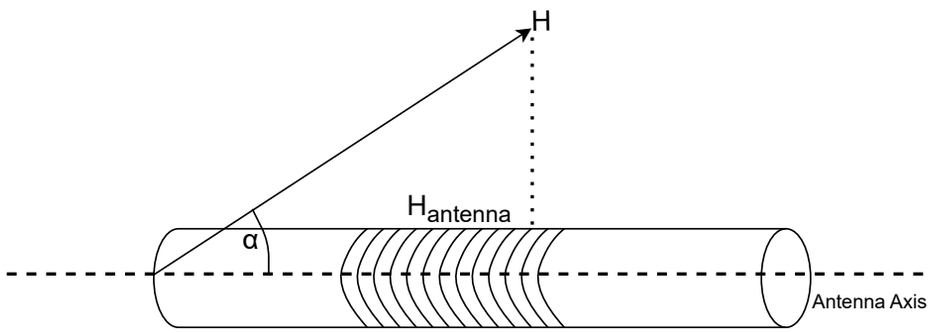


Figure 2.2: This diagram depicts the magnetic field ($H_{antenna}$) detected by a magnetic dipole antenna based on the alignment of the antenna's axis and the magnetic field. This illustration has been recreated from the illustrations found in Ayuso et al.'s paper on avalanche beacons[2].

the magnetic field detected by one antenna, while the second antenna detects a magnetic field that is offset by ninety degrees from the first, as described by Equation 2.6.

$$H_{antenna_2} = H \cdot \cos\left(\alpha + \frac{\pi}{2}\right) = H \cdot \sin\alpha \tag{2.6}$$

To determine the strength of the magnetic field and the orientation of the antenna in relation to it, one can measure the magnetic field in both antennas simultaneously. Equations 2.7 and 2.8 can be used for this purpose. With the help of Equation 2.8, modern avalanche beacons can estimate the direction of the magnetic field from a transmitting beacon. To estimate the distance to the transmitting beacon, modern avalanche beacons utilize a combination of Equation 2.7 and Equation 2.4.

$$H = \sqrt{H_{antenna}^2 + H_{antenna_2}^2} \tag{2.7}$$

$$\alpha = \tan^{-1}\left(\frac{H_{antenna_2}}{H_{antenna}}\right) \tag{2.8}$$

2.1.2 Signals

Per the regulations laid out by the European Telecommunications Standards Institute in ETSI 300718[6], the signals transmitted by avalanche beacons must adhere to specific parameters. The most critical aspects of these parameters for our project are the frequency specifications and the on/off keying of the signal.

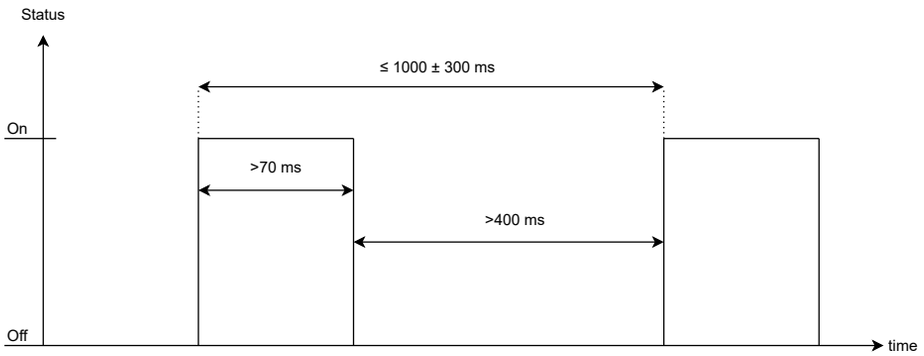


Figure 2.3: Signal characteristics of avalanche beacons as per the European Telecommunications Standards Institute[6].

For instance, ETSI 300718 mandates that the transmitted signal's frequency should be $457kHz$, with an allowable frequency offset of $\pm 80Hz$. This information is crucial to keep in mind while developing methods to detect avalanche beacon signals.

Avalanche beacons do not transmit a continuous signal at all times. Instead, the signal is keyed using a simple on-off keying technique. On/off keying (OOK) is a simple digital modulation technique used in communication systems. It involves turning the carrier signal on and off to transmit a recognizable signal. In the case of avalanche beacons, the carrier signal is the $457kHz$ radio signal transmitted by the beacon. The exact specification of the on/off timing as per the ETSI 300718 is illustrated in Figure 2.3. The signal is supposed to be on for $70ms$ or more and off for $400ms$ or more. The whole on/off period is supposed to be less than $1000ms \pm 300ms$.

Avalanche beacons use on/off keying for two important reasons. Firstly, it helps to conserve battery life, which is critical for anyone caught in an avalanche as their beacon needs to transmit for as long as possible. Secondly, it allows rescuers to detect if there are multiple beacons transmitting in the area. Detecting several beacons at once is possible because if multiple beacons are turned on, they will likely have slight differences in their on/off keying. If the receiver receives signals for more than $70ms$ within a second, then it is highly likely that there are multiple transmitting beacons nearby. Differences in the intensity of signals received, along with the offset in the on/off keying, can be utilized to distinguish the direction and distance of various transmitting beacons.

2.2 Digital signal processing

Digital Signal Processing (DSP) is a branch of signal processing that focuses on the manipulation, analysis, and interpretation of signals using digital techniques. It involves the use of algorithms and mathematical operations to process discrete-time signals represented as sequences of numbers[13].

One of the key concepts in DSP involves converting analog signals to digital format using analog-to-digital conversion (ADC). This process involves sampling a continuous-time signal and transforming it into a discrete-time signal that can be saved in computer memory. An analog signal is a continuous-time signal that can be mathematically represented as a function of time, denoted by x(t). Conversely, a discrete-time signal is a sequence of indexed numbers, x[n], where n indicates the order of values in the sequence. When saving a signal digitally, the index n corresponds to the memory position of the nth value. To obtain a discrete-time signal, x[n], from a continuous-time signal, x(t), we must sample the continuous signal at equally spaced intervals as defined by Equation 2.9.

$$x[n] = x(n\frac{1}{f_s}), -\infty < n < \infty, n \in \mathbb{N} \tag{2.9}$$

In the case of sampling a continuous-time signal to obtain a discrete-time signal, the values in the sequence $x[n]$ are referred to as samples of $x(t)$, and f_s is the sampling rate and is usually specified in samples per second. After sampling a continuous time signal it is possible to do digital analysis and signal processing on the discrete time signal. It is even possible to reconstruct a continuous-time signal from a discrete time signal given that the discrete time signal is sampled at a sufficiently high sample rate. In order to determine what sample rate is sufficent for a given signal one needs to consider the most important theorem of digital signal processing, the Nyquist-Shannon sampling theorem(Theorem 2.2.1).

Theorem 2.2.1 (Nyquist–Shannon sampling theorem)**.** *A continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n] = x(nT_s)$, if the samples are taken at a rate $f_s = \frac{1}{T_s}$ that is greater than $2f_{max}$.*

2.2.1 I/Q data

When sampling real-world signals for digital signal processing it is common to store the data as complex values in the form of I/Q data. I/Q data refers to the in-phase (I) and quadrature (Q) components that represent a complex-valued signal. It is a fundamental concept used in many applications, including wireless communication, radar systems, and signal analysis.

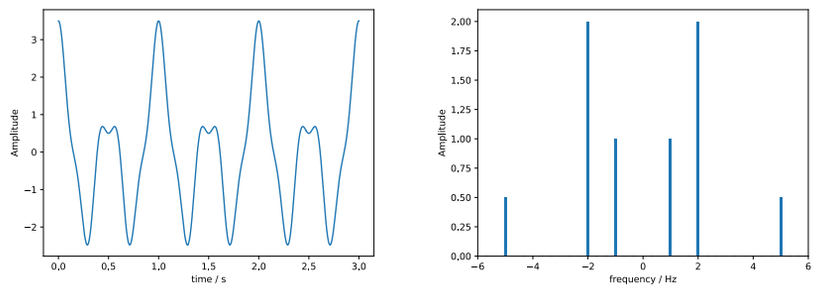
The I/Q data representation is based on the concept of quadrature modulation, where the signal is split into two components with a 90-degree phase difference. The I component represents the amplitude and phase of the signal in phase with a reference carrier, while the Q component represents the amplitude and phase 90 degrees out of phase with the reference carrier.

By using I/Q data, it is possible to fully characterize a complex signal in terms of both amplitude and phase information. This representation enables various operations, such as demodulation, modulation, frequency translation, and signal analysis.

2.2.2 Frequency domain

Frequency domain analysis is another key area of DSP, focusing on the extraction of frequency information from a signal. Techniques like the Fourier transform and its fast implementation, the Fast Fourier Transform (FFT), are widely used to analyze the frequency content of signals, identify specific frequency components, and study their power spectrum.

Understanding the frequency domain is crucial in signal processing as it offers a unique perspective on signals compared to the time domain. It involves breaking down a signal into its frequency components and their magnitudes. Representing a signal in the frequency domain allows us to express a signal as a sum of sinusoidal functions with varying frequencies, magnitudes, and phases. The frequency domain representation of a signal is commonly visualized using a frequency spectrum or power spectrum plot. In such plots, the x-axis represents the frequency values, and the y-axis represents the magnitudes or power levels of the corresponding frequency components. The resulting spectrum provides a graphical representation of how energy is distributed across different frequencies in the signal. For instance, Figure 2.4 presents an example of a continuous-time signal $x(t)$ depicted in both the time and frequency domain. The signal is a combination of three sinusoids with different amplitudes and frequencies.



(a) Time domain representation of the signal $x(t)$

(b) Frequency domain representation of the signal $x(t)$

Figure 2.4: Time domain representation of the continuous-time signal $x(t) = 2\cos(2\pi 2 \cdot t) + \cos(2\pi t) + \frac{1}{2}\cos(2\pi 2 \cdot t)$ and its frequency domain representation

By examining a signal in the frequency domain, we can gain insights into its spectral characteristics, such as the presence of specific frequencies or frequency bands, the distribution of energy across different frequencies, and the relationships between different frequency components.

Fourier transform

The Fourier transform is a mathematical tool used in signal processing to analyze and represent a signal in the frequency domain. It decomposes a time-domain signal into its constituent frequency components, revealing the frequency content and magnitude of each component.

The continuous-time Fourier transform (CTFT) is used for continuous-time signals, while the discrete-time Fourier transform (DTFT) is used for discrete-time signals. However, the most commonly used form of the Fourier transform in practical applications is the discrete Fourier transform (DFT), which is applied to finite-length discrete-time signals.

The Fourier transform takes a time-domain signal as input and produces a frequency-domain representation. The transform operation converts the signal from the time domain to the frequency domain through a series of complex mathematical computations. The resulting frequency-domain representation provides information about the amplitudes and phases of each frequency component contained in the original signal.

The inverse Fourier transform allows us to reconstruct the original time-domain

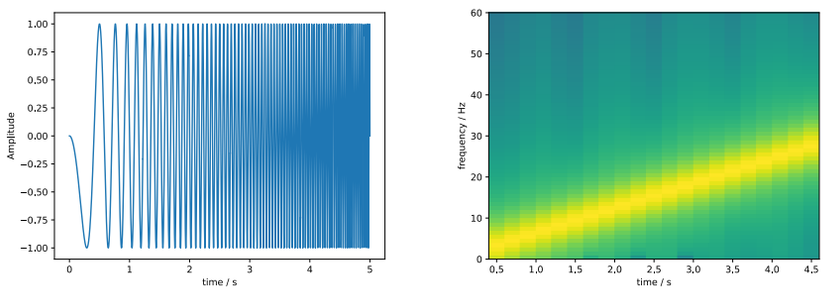
signal from its frequency-domain representation. It converts the frequency-domain signal back to the time domain, enabling us to recover the original waveform.

The Fast Fourier Transform (FFT) is an efficient algorithm for computing the DFT. It significantly reduces the computational complexity of the discrete Fourier transform, making it practical for real-time signal processing and analysis.

2.3 Spectrogram

At times, a signal's frequency components may vary over time, making the Fourier transform insufficient to fully represent the signal. To address this, a spectrogram is used as a visualization tool in signal processing. It displays the frequency content of a time-varying signal over time, providing a detailed and intuitive representation of how the frequency components change over different time intervals.

The process of obtaining a spectrogram involves using a technique called short-time Fourier transform (STFT) on the signal. STFT breaks the signal down into small, overlapping segments and computes the Fourier transform for each of these segments. This enables the analysis of the frequency content of the signal at various points in time. To create a spectrogram, the magnitude of the resulting Fourier transforms is typically plotted against both the time and frequency axes. The intensity or color of the plot represents the magnitude of the frequency components, with brighter colors indicating higher magnitudes and darker colors indicating lower magnitudes. A typical example of when a spectrogram is useful is when dealing with chirp signals. A chirp signal is a signal in which the frequency components are constantly increasing. Figure 2.5, shows both the time domain representation and the spectrogram representation of a chirp.



(a) Time domain representation of the signal $x(t)$

(b) Spectogram representation of the signal $x(t)$

Figure 2.5: Time domain representation of the continuous-time signal $x(t) = \cos(2\pi \cdot 3x^2 + \pi/2)$ and its spectrogram

The resolution of the spectrogram's time is dependent on the length and overlap of the segments used in the STFT. Shorter segments have better time resolution, capturing quick changes in the signal. Longer segments provide better frequency resolution, revealing finer details in the frequency content. In the context of a spectrogram, frequency bins and time bins refer to the segments along the frequency and time axes, respectively. Frequency bins represent the divisions along the frequency axis, defining the discrete frequencies at which the magnitude of the frequency components is measured and displayed. The number of frequency bins is typically determined by the length of the Fourier transform applied to each signal segment during spectrogram computation. Time bins represent the divisions along the time axis and determine the duration of each signal segment or window used for computing the Fourier transform. The length and overlap of the time bins chosen affect the time resolution of the spectrogram.

2.4 Software defined radios

The main objective of this project is to develop a comprehensive digital receiver for avalanche beacons. To accomplish this, the initial step involves capturing and digitizing the analog radio signals emitted by the avalanche beacons. To facilitate this crucial conversion process, the project will leverage the capabilities of software-defined radios (SDRs). SDRs are external devices connected to a computer that enable efficient digital signal processing of radio signals. While SDRs typically integrate essential radio frontend components, such as an analog-to-digital converter, low-noise amplifier, and down conversion, in hardware, they offer the added advantage of enabling further signal processing to

be performed digitally using the power of a computer.

The utilization of SDRs in this project not only enables the initial analog-to-digital conversion but also opens up possibilities for future enhancements and optimizations. Since SDRs rely on software for signal processing, they offer a platform for iterative improvements and experimentation. Researchers and developers can refine and tailor the digital signal processing algorithms and explore novel methods for analyzing and interpreting the received signals. This inherent flexibility of SDRs fosters innovation and provides a solid foundation for the creation of a robust and versatile digital avalanche beacon receiver.

2.4.1 Down Conversion

Down conversion plays a crucial role in Software Defined Radio (SDR) systems, serving as a fundamental process for frequency manipulation. In down-conversion, the objective is to shift the frequency of a received signal from a higher frequency band to a lower frequency range, typically known as the intermediate frequency (IF). This transformation allows software-defined radios to effectively process and reconstruct signals with frequencies higher than the limit imposed by the Shannon-Nyquist sampling theorem(Theorem 2.2.1), which states that a signal must be sampled at least twice its highest frequency component to be accurately reproduced.

By employing down conversion, SDRs enable the use of lower sampling rates while still preserving the integrity and fidelity of high-frequency signals. The received signal, initially situated in a higher frequency band, is mixed or multiplied with a locally generated frequency, known as the local oscillator (LO) signal. The LO signal has a frequency offset that aligns with the desired down conversion frequency. The result of this mixing process is the generation of new signals at the sum and difference of the frequencies present in the received signal and the LO signal.

The crucial advantage of down conversion lies in the ability to shift the received signal to a lower frequency band, which allows for easier and more efficient signal processing in subsequent stages. Once down-converted, the signal can be digitally processed using software-based algorithms, including filtering, demodulation, and decoding. This digitized and down-converted signal can then be readily manipulated, analyzed, and utilized for various applications within the SDR system.

3

Related technologies

3.1 Avalanche beacons

Avalanche beacons, also known as transceivers, were invented to help save lives during an avalanche. Prior to their invention, finding someone buried in the snow was a difficult and time-consuming process that often resulted in fatalities. With the use of avalanche beacons, rescuers are able to quickly locate buried individuals and initiate a rescue, increasing the chances of survival. Avalanche beacons have become an essential tool for backcountry skiers, snowboarders, and mountaineers, allowing them to enjoy the winter landscape with an added layer of safety. Avalanche beacons are portable devices that can be worn under clothing. They work as both radio transmitters and receivers. When used in avalanche terrain, the beacon is set to sending mode, transmitting a low-frequency signal every second. A collection of different avalanche beacons is shown in Figure 3.1. In the unfortunate event of an avalanche burial, the rescuer can switch the beacon to receiving mode, which provides directional and distance indicators to locate the buried victim. Without an avalanche beacon, it is unlikely that a companion rescue will be successful.

3.1.1 History of avalanche beacons

In 1970, the SKADI beacon(Figure 3.1a) was introduced as the first avalanche beacon. Its purpose was to enhance the safety of backcountry professionals, particularly ski patrollers and road maintenance crews. The SKADI operated

(a) The SKADI avalanche transceiver, one antenna broadcasting a signal at 2.275 kHz. This was the first commercial avalanche beacon, which started retailing in 1970.

(b) The Ortovox F1 was the first avalanche beacon to support both 2.275 kHz and 457 kHz. This beacon became available to consumers in 1980

(c) BCA tracker, the first "digital" beacon, came out in 1997. This beacon featured two antennas running at 457 kHz and could indicate the direction and distance of the buried victim.

(d) Mammut Barrywox S is shown to represent state-of-the-art avalanche beacons. These beacons feature three antennas, an LCD screen with instructions to the rescuer, and between beacon communication.

Figure 3.1: Developments in Avalanche Beacons since its invention in 1970

on a low frequency of 2.275 kHz to allow the radiowaves to penetrate the snow surface without significant attenuation. In sending mode the SKADI worked by repeatedly sending out a radio pulse of 2.275 kHz. When turned to receive mode, the SKADI would convert the radio signal directly to an audible tone. The closer the victim was to the rescuer, the higher the volume of this tone would be.

As backcountry travel gained popularity, an increasing number of avalanche beacons became accessible to consumers. There was a division among the producers regarding the frequency to be used. Some opted for the 2.275 kHz frequency of the SKADI, while others preferred a much higher frequency of 457 kHz. This division posed a problem as users with the 2.275 kHz beacon could only rescue other users with the same frequency, and vice versa for the 457 kHz beacons. There were advantages and disadvantages to both frequency levels. The benefit of the 2.275 kHz frequency was that it was within the audible range, so no demodulation was needed to convert it to an audible tone, and the extremely long wavelength made it robust against obstacles that could attenuate the signal. The 457 kHz signal, on the other hand, boasted a longer operational range of 40-70 meters against the 20-30 meter range of the 2.257 kHz signal. In 1980 the outdoor brand Ortovox released the F1 beacon(Figure 3.1b), which supported both frequencies. The division ended in 2001 when a European standard stated that an avalanche beacon should use the frequency 457 kHz[7].

Back in 1997, the outdoor equipment brand BCA (Backcountry Access) launched a groundbreaking beacon named Tracker(Figure 3.1c). This innovative device featured multiple antennas and a unique signal processing method that enabled it to determine the direction and approximate distance of the victim. To be more precise, the tracker had two antennas positioned at right angles to each other. By analyzing the signal strength of both antennas, it was possible to estimate the general direction of the buried individual through triangulation. The term "digital" avalanche beacon is commonly used to refer to these types of beacons, even though the majority of signal processing is accomplished through hardware rather than software.

Today's state-of-the-art avalanche beacons, like the Mammut Barrywox S (as seen in Figure 3.1d), prioritize usability over new technology. LCD screens and clear instructions have been added to enhance usability, along with voice guidance in specific situations during rescue operations. Although the latest avalanche beacons have a third antenna and improved precision, the technology remains similar to that of the BCA tracker.

3.2 Avalanche search and rescue today

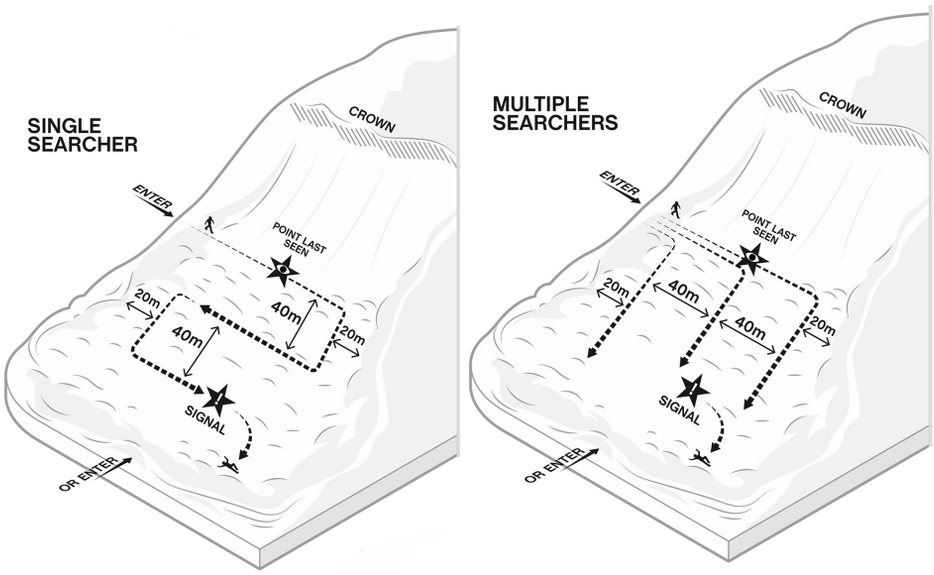
As mentioned and illustrated by Figure 1.1, time is of the essence during an avalanche rescue. In an interview with a rescue professional in the Norwegian air ambulance, it was estimated that the best-case response time to an avalanche incident was 22 minutes. This response time was only estimated as the duration from the rescue services received a distress call until rescue personnel was on the scene, not until the victim was dug up. This response time means that if a victim is to be rescued alive, they should be rescued by their touring companions. Therefore, all who venture into avalanche terrain should have proper rescue gear and training.

When it comes to avalanche search and rescue, it is vital to understand the differences between companion rescue and professional rescue. Companion rescue refers to the immediate rescue efforts carried out by fellow backcountry travelers who are trained in avalanche rescue techniques. In contrast, professional rescue involves a team of trained and certified rescuers who are equipped with specialized gear and equipment to carry out large-scale rescue operations. The techniques and equipment used in each type of rescue vary greatly and must be tailored to the specific situation at hand. Understanding these differences is crucial for rescuers to carry out a successful and safe rescue operation.

3.2.1 Companion rescue

For companion rescue, it is imperative that both the victim and the rescuer are equipped with an avalanche beacon. The workings of an avalanche beacon are discussed in greater detail in section 3.1. A common companion rescue can usually be divided into six phases; *Signal search*,*Course search*,*Fine search*, *Stick search*, *Digging*, *First aid*. Depending on the incident, some phases might vary from this general description, and some might even be omitted during the search. It is, however, vital that a rescuer is familiar with all of these phases and has trained enough in avalanche rescue to know how each phase is carried out in practice.

In the event of an avalanche burial, the rescuer will start by assessing the scene. Usually, this involves looking for signs of the victim that might indicate the burial location. The rescuer will then switch his/hers beacon into receive mode and commence the *signal search*. The *signal search* involves running in a zig-zag pattern across the avalanche, trying to receive a signal on the beacon. The width of each zig-zag pattern should be a little less than the search radius of the beacon, usually between 40-60m depending on the beacon. A signal search with only one rescuer is illustrated in Figure 3.2a. With multiple rescuers, a



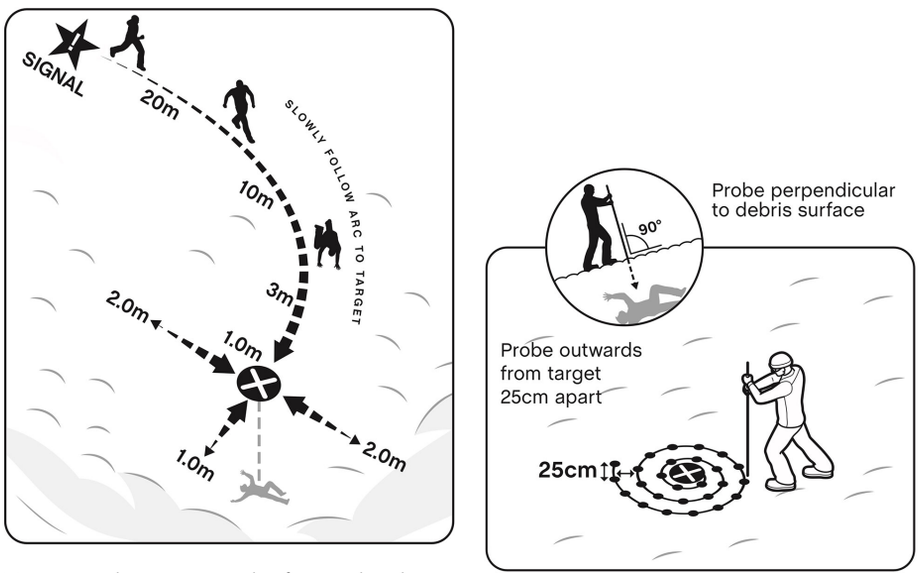
(a) Signal search with only one rescuer **(b)** Signal search with multiple rescuers

Figure 3.2: 3.2a and 3.2b shows how a single rescuer or several rescuers might do a signal search. The Diagrams are taken from the American Institute for Avalanche Research and Education[22].

search can be done in parallel, as illustrated by Figure 3.2b. In the case of multiple rescuers, the distance between each rescuer should, as with a single rescuer, be a little less than the search range of the beacon in use.

Once a signal has been obtained, the rescuer should start a *course search*. During the *course search*, the rescuer should be running in the direction indicated by the beacon. Once the rescuer is within five to ten meters of the buried victim, he/she slows down and moves the beacon closer to the snow surface. When the beacon indicates a distance of fewer than five meters, the rescuer begins a *fine search*. In this phase of the rescue, the rescuer moves the beacon left to right, then forward and backward within a grid, looking for the point that is closest to the victim. Both the course search and the fine search is illustrated by Figure 3.3a. Once the closest point to the victim is discovered, the rescuer moves on to probing the snow to get a confirmation of the victim's depth.

The probing is done in a circular fashion, as displayed in Figure 3.3b, starting at the point nearest the victim. The rescuer probes the snow perpendicular to the snow surface until he/she can feel the victim with the probe. Once the victim's position is confirmed, the rescuer must start digging. Once the victim is out of the snow, it is usually necessary to provide first aid and wait for rescue personnel to arrive.



(a) Fine and course search after avalanche victim

(b) Probing search after victim

Figure 3.3: 3.3a shows how a course and fine search is carried out by a rescuer, and 3.3b shows the subsequent probe search. The Diagrams are taken from the American Institute for Avalanche Research and Education[22].

The method of rescue described in this section considers events with only one victim. In the event of multiple burials, the rescue flow is similar, but the fine and course search phases will be considerably more difficult. Each of these phases will need to be done for each buried victim.

3.2.2 Professional rescue

When it comes to rescue missions, there are two main types: professional and companion rescue. Although they share similarities, there are some key differences in the nature of the missions and the equipment used. Professional rescue involves two types of searches - those using avalanche beacons and those without.

Avalanche incidents typically occur in isolated and difficult-to-reach locations, which can delay the arrival of rescue teams. As a result, companion rescue often becomes crucial for successful outcomes. According to current recommendations for rescue personnel, victims buried for less than an hour should be assumed to be alive[16]. However, if they have been buried for more than an hour, the nature of the incident must be taken into account before deciding whether to consider the victim alive or not.

It is crucial to differentiate between whether the victim is alive or not, as it drastically impacts the time frame for the rescue. In situations where the victim is not equipped with an avalanche beacon, or there is no signal from it, other methods of locating the victim, such as Recco search, dog search, or grid probe search, must be used. Unfortunately, if the victim is not equipped with an avalanche beacon, the chances of rescuing them alive are very slim.

Victims with beacons

Ideally, the victim should be extricated from the avalanche by their companions, enabling the rescue team to prioritize administering critical first aid. If the avalanche victims are not found immediately, the rescue personnel must start searching the avalanche. This search is similar to a companion rescue search. However, first responders, such as rescue personnel, often have the ability to make a signal search from a helicopter, which can greatly increase the chances of finding the victims quickly and safely. It's important to remember that time is of the essence in these situations, so every effort should be made to locate and rescue the victims as soon as possible. During an avalanche rescue operation, a beacon device connected to the helicopter's intercom is used to search for signals. The device emits a beep each time it receives a signal, and the volume of the beep increases with the strength of the signal. This enables the rescue team to assess the situation quickly before sending personnel to the ground. Once on the ground, the team conducts a companion rescue search, as described in section 3.2.1.

Victims without beacons

It can be challenging to locate individuals who are in distress when there are no beacon signals available. This may occur for various reasons, such as not having a beacon, or the beacon losing power or being damaged during an avalanche. In these situations, alternative rescue methods are necessary. However, searching for victims without beacons can be a time-consuming and expensive process that requires a significant amount of personnel and equipment. Apart from beacon search, there are three other common ways to search for avalanche victims: RECCO search, canine search, and grid probe search[19].

The RECCO search method functions similarly to an avalanche beacon search. However, instead of using a transmitter on the victim and a receiver on the rescuer, both the transmitter and receiver are placed on the rescuer. The victim wears a radio reflector instead. The rescuer then sends a RECCO signal in a particular direction. When the victim is positioned in front of the transmitter, the reflector will bounce the signal back to the rescuer's receiver, providing an

indication of the victim's location. Several outdoor clothing brands integrate the RECCO reflector into their apparel and gear for outdoor activities.

Canine rescue teams are highly effective in avalanche search and rescue operations due to their sense of smell and agility, allowing them to locate buried victims faster and more accurately than human rescuers alone. However, despite their effectiveness, canine rescue teams face a number of unique challenges, including harsh weather conditions, difficult terrain, and the risk of injury or death to both the dogs and their handlers. As such, it is crucial that rescue teams have access to high-quality training and equipment, as well as adequate support and resources, to ensure the safety and success of their missions.

In situations where beacon, RECCO, or canine searches fail to yield results, rescuers may resort to a grid probe search. This involves rescuers lining up at arm-length intervals and systematically probing the avalanche in a grid pattern, with each probe point spaced approximately 50cm apart. Searching for victims using this method is effective but requires a significant amount of time and personnel. Therefore, this method of searching is usually only needed when looking for the bodies of avalanche victims.

3.3 Choice of SDR

In order to digitally interpret radio signals, a software-defined radio was required. After careful consideration, the Airspy HF+ discovery was ultimately selected based on its affordable price, compact size, adequate sampling resolution, and low-frequency coverage. Before making the final decision, I evaluated several options for software-defined radios. These included the *Universal Software Radio Peripheral (USRP)* from Ettus Research[1], the *Pluto SDR* from ADALM[2], and three open-source SDRs: the *RTL-SDR*[3], the *Airspy HF+*[4], and the *HackRF One*[5]. A summary of the most relevant features of these is shown in Table 3.1 and a discussion on the pros and cons of each SDR will follow in the next five paragraphs.

1. https://www.ettus.com/products/
2. https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview
3. https://www.rtl-sdr.com/about-rtl-sdr/
4. https://airspy.com/airspy-hf-discovery/
5. https://greatscottgadgets.com/hackrf/one/

Device	Sample resolution	Sample rate	Communication standard	Open source	Frequency range
USRP N200	16 bit	100 Msps	Ethernet	no	DC(0 Hz)-6 GHz
RTL-SDR	8 bit	2.5-3.2 Msps	USB 2.0	yes	500 kHz-1.75 Ghz
Airspy HF+	16 bit	768 ksps	USB 2.0	yes	HF: 0.5 kHz-31 MHz VHF: 60-200 MHz
HackRF One	16 bit	20 MsPs	USB 2.0	yes	1 MHz-6 GHz
Pluto SDR	12 bit	61.44 MsPs	USB 2.0	no	325 Mhz-3.8 GHz

Table 3.1: Summary of the most relevant features of all software-defined radios considered for this project.

3.3.1 Universal Software Radio Peripheral(USRP)

For this project, the most advanced software-defined radio (SDR) taken into account was the Universal Software Radio Peripheral, commonly known as USRP. USRP refers to a collection of SDR models from Ettus Research rather than a single SDR. To be more precise, the model that was considered for this project was the USRP N200. Among the SDRs assessed for this project, this specific one stands out for its exceptional sample rate and broad frequency range. It boasts a sample rate of 100 Msps, a frequency range of 0 Hz-6 GHz, and a high sample resolution of 16 bits. It differs from the rest of the SDRs in that it communicates through an ethernet connection instead of USB 2.0, which allows for a much higher streaming rate of 50 Msps. Although the USRP N200 demonstrates impressive performance when compared to other SDRs, it, unfortunately, does not meet the requirements for this project due to some drawbacks. Firstly, there is limited information available on how to use the USRP SDRs outside of Ettus Research's own documentation, which makes quick development more difficult. Additionally, it is very expensive, with prices starting at 10-30 times higher than other SDRs considered. However, the biggest disadvantage is its size, measuring a footprint of 20x15 cm. Furthermore, it has a separate power supply which makes it unsuitable for meeting the third project requirement of fitting onto a regular unmanned vehicle.

Figure 3.4: The USRP N200 from Ettus research

3.3.2 RTL-SDR

Similar to the USRP, RTL-SDR does not refer to a single SDR model. It actually pertains to a particular type of inexpensive, compact SDR USB dongles that utilize the RTL2832U chipset[1]. The RTL-SDR dongles originated as an open-source endeavor when it was uncovered that the I/Q data on the chipset RTL2832U, which was utilized in a widespread DVB-TV tuner, could be accessed directly. This allowed amateur radio operators to easily create their own cheap SDRs from these TV tuners. Nowadays, several manufacturers offer RTL-SDRs with minor differences in specifications. However, most RTL-SDR brands deliver 8-bit sampling resolution, roughly 2.5 Msps sampling rate, and frequency coverage of 500kHz-1.75GHz. One of the main benefits of using RTL-SDR dongles is their compact size. Although sizes may differ depending on the model, they typically have a footprint of 2x4 cm. Finally, it's worth noting that the RTL-SDR lacks official documentation. However, due to its widespread popularity among amateur radio enthusiasts, a wealth of information can be readily found on numerous forums dedicated to the hobby. After careful consideration, it was decided that an RTL-SDR dongle would not be suitable for this project due to

inadequate documentation, insufficient sampling resolution, and a frequency range that is above that of avalanche beacons.

Figure 3.5: This is a small assortment of various RTL-SDRs. Despite their distinct appearances, they all employ the same chipset internally.

3.3.3 Pluto SDR

The ADALM Pluto SDR is an educational SDR that boasts impressive features like a sampling resolution of 12 bits and a high sampling rate of 61.44 Msps. Being a tool designed for educational purposes, it is extensively documented and provides libraries for various programming languages like Python, C, C++, and C#. The Pluto SDR not only receives signals but also has a digital-to-analog converter allowing for radio signal transmission. However, as mentioned in section 1.1, this project does not require signal transmission, making this feature unnecessary for the project. For this project, the Pluto SDR seemed like a great option due to its user-friendly interface, support for popular programming languages, and thorough documentation. Unfortunately, the frequency coverage of the Pluto SDR is far beyond the necessary 457 kHz for avalanche beacons, rendering it unsuitable for this particular task.

Figure 3.6: The Pluto SDR from ADALM

3.3.4 HackRF One

The HackRF One SDR is intended for the purpose of developing and testing contemporary and future radio technology. This device boasts a 16-bit sampling resolution and a sampling rate of 20 Msps. Its frequency coverage ranges from 1 Mhz to 6 Ghz, surpassing the frequency of avalanche beacons. It should be noted, however, that the Hack RF one can operate below 1 Mhz, albeit with a decrease in performance[6]. The Hack RF also has a built-in variable LNA which can be tuned from 0 dB to 40 dB in 8 dB steps. The HackRF One is a popular SDR device among radio enthusiasts, similar to the RTL-SDR. As a result, there is a wealth of information available online on how to use it. However, unlike the RTL-SDR, HackRF One has a comprehensive documentation page maintained by the community. Like the Pluto SDR, the HackRF One is capable of transmitting radio signals, but for the purposes of this project, this feature is not relevant. Unfortunately, I had to exclude the Hack RF from this project as it is unable to function optimally at the targeted frequency of 457 kHz.

6. https://greatscottgadgets.com/2015/05-15-hackrf-one-at-1-mhz/

Figure 3.7: The HackRF One SDR

3.3.5 Airspy HF+ Discovery

The Airspy HF+ Discovery is an affordable SDR that boasts a high sample resolution of 16 bits and covers frequencies ranging from 0.5 kHz to 31 MHz, as well as 60 MHz to 200 MHz. One thing to note is that its sampling rate is significantly lower compared to the other SDRs being considered for this project. Although there is limited documentation available, the Airspy HF+ Discovery is a widely used SDR that benefits from strong community support and open-source software[7]. By using the Airspy HF+ Discovery's open-source host software, users can effortlessly stream I/Q data directly to a file or file stream. This feature is significant since one of the project's necessities is to save signal data for future analysis.

7. https://github.com/airspy/airspyhf

Figure 3.8: The Airspy HF+ Discovery

3.3.6 Final choice of SDR

After careful evaluation, I decided to select the Airspy HF+ discovery as the SDR. This was due to its default frequency coverage being able to detect avalanche beacons, compact size of 4.5x6 cm, and affordable price point. Additionally, I appreciate the fact that there is already software available that can stream signal data to a file. This saves me time and effort in the development process.

4

Related literature

The area of research pertaining to avalanche beacons and autonomous drone rescue is a relatively niche topic. Some research has been conducted in this area, but the number of publications is limited. Despite this, the potential benefits of using autonomous drones for avalanche rescue are significant. In this section, I will explore the existing research on avalanche beacons and autonomous drone rescue, focusing on identifying gaps in the current knowledge and potential areas for future research.

4.1 The AVERLA project.

The AVERLA project was a part of the "Innovative Student Projects" program at the University of Padova. Its primary objective was to create a small drone to locate transmitting avalanche beacons nearby quickly. The project was divided into three phases: antenna design and prototyping, autonomous flight, and research on patterns for avalanche rescue.[20, 10, 4]

This project's most relevant result was the antenna developed. The project created a magnetic dipole antenna using a ferrite core with a spool of copper wire around it, similar to the antennas in commercial avalanche beacons. They connected three of these antennas to an antenna switch and a control logic, allowing them to combine the received signal from all three antennas or listen to the specific components of one antenna. They created two antennas, a small

one at which they achieved a range of 13-14 meters and a larger one with a higher inductance. The larger antenna achieved a maximum range of 40 meters. However, it was so large that it made it challenging to integrate on a small drone. Ultimately, they proved the feasibility of manually constructing a ferrite core antenna tuned to the frequency of avalanche beacons. Still, an algorithm to detect the signal has yet to be developed.

The other works of the AVERLA project focused on implementing a drone capable of autonomous flight in avalanche-prone environments and finding an optimal flight pattern for detecting avalanche beacons. The topics of these two papers are outside the scope of this thesis, and their results will, therefore, not be presented in greater detail. However, the reader is encouraged to examine the original papers if the topics are of interest.

4.2 Non-autonomous drone

Janovec et al. designed a prototype of a crewless aerial vehicle for use in avalanche search and rescue operations[11]. The prototype utilized a camera pointed toward the screen of a commercial avalanche beacon connected to a drone. The whole system had to be controlled by two operators, one controlling the drone controller and navigating using another camera pointing forward on the drone, the other looking at the camera feed of the commercial avalanche beacon and giving instructions to the operator controlling the drone. Although the solution presented in the article is relatively basic, the authors demonstrated that detecting a transmitting avalanche beacon from a drone is achievable. However, they experienced significant radio interference from the engines. Through experimentation, they found that they had to move the avalanche beacon 50cm away from the drone body to remove the effects of interference.

4.3 Using other methods than avalanche beacons for search and rescue

There has been an exploration of alternative techniques for avalanche rescue, aside from using avalanche beacons. These techniques use optical or thermal images, radio protocols like cellular or satellite positioning systems, or radar[3, 14, 15]. However, these methods have yet to reach a level of accuracy that could make them relevant in real-world situations.

The primary limitation of the methods involving optical or thermal imaging is that they only work for victims not entirely buried in the snow.

Research on the use of cellular or satellite positioning systems for avalanche rescue has found that the snow attenuates the signal too much, resulting in a high positional error that makes it challenging to locate victims.

Moodroo et al demonstrated that ground penetrating radar can detect buried objects in snow-masses[14]. However, differentiating between these objects is very difficult, which makes it unsuitable for avalanche rescue where quick localization is essential.

4.4 Software-defined radio-based receiver.

In his master's thesis from the University of Upsala, Hedlund describes the design and creation of an avalanche beacon receiver using software-defined radios[9]. The master thesis was part of a larger project at ÅF Digital Solutions AB.

Hedlund uses two software-defined radios connected to two ferrite core antennas stationed perpendicular to each other. Experts within the larger project provided the ferrite core antennas.

Hedlund proposes a detection scheme that processes the received signal in three steps. The signal is first filtered through a low pass filter and decimated to a lower sampling rate. After that, Hedlund applies a matched filter to the filtered signal. Lastly, the finished signal is thresholded against a predetermined power level; if a power level is above the threshold, Hedlund concludes that there must be a transmitting avalanche beacon nearby.

Hedlund's experiments show a detection range of 15 meters, which is far from the range of commercial avalanche beacons. However, he carried out experiments in an office location with much radio interference, and he hypothesizes that the range would be drastically higher in a mountain environment. However, proof of this hypothesis needs to be presented.

Ultimately, Hedlund manages to create a working prototype utilizing software-defined radios to detect avalanche beacons automatically, and in the process, shows that commercial software-defined radios are capable of providing high-quality signals for detecting avalanche beacons.

4.5 Conclusion

Although limited, there have been some published works on the use of technology in avalanche rescue operations. Most of the research discusses the use of crewless vehicles in rescue operations and focuses on the vehicles themselves or the search patterns they use. Other studies propose rescue techniques that do not rely on avalanche beacons. However, these have yet to be as accurate. More research is needed to optimize the digitalization of avalanche beacons.

5

Version 1

This chapter will describe the first attempt at designing an avalanche beacon using software-defined radios. The chapter will begin by discussing the design and implementation of the beacon. Thereafter, a description of two of the algorithms for detecting avalanche beacons presented in this thesis. Next, I will delve into the experiments conducted with the beacon and their outcomes. Finally, I will conclude this chapter with a discussion of the advantages and disadvantages of this initial design. As an additional remark to this chapter, I will present a hypothesis regarding the enhancement of the initial design, followed by an implementation attempt that effectively invalidated this hypothesis.

5.1 Implementation and design

5.1.1 Passive antenna

The first and most important part to get working was an antenna that could pick up the low-frequency signals of an avalanche beacon. In general, antennas can be classified into two main categories: electric and magnetic. This classification is based on the antenna's reactivity to either the electric field or the magnetic field. The most common electric antenna is a resonant-length dipole antenna. To achieve resonance in an electric antenna, its length should be a quarter of the wavelength of the desired radio frequency or a multiple of that length. The

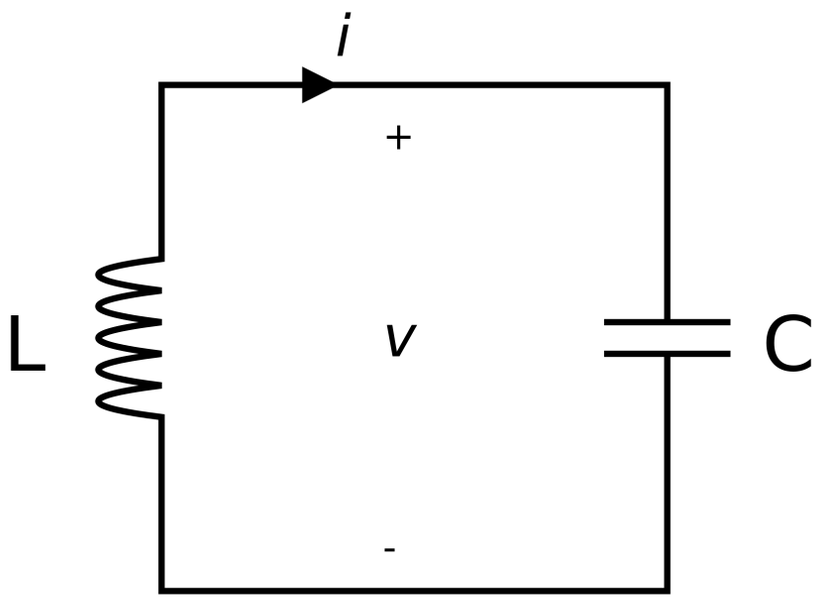


Figure 5.1: Typical LC circuit used for magnetic dipole antennas.

wavelength λ of a radio wave with frequency f is given by Equation 5.1.

$$\lambda = \frac{c}{f} \tag{5.1}$$

For avalanche beacon signals operating at $457kHz$, Equation 5.1 results in a wavelength of roughly $656m$. Even an antenna with a length of a quarter of this wavelength is infeasible in this project. On the other hand, magnetic antennas achieve resonance by matching the inductance of an inductor coil with a capacitor. As a result, they can be much smaller than their electric counterparts. Therefore, due to their smaller size compared to their electric counterparts, I opted to pursue the creation of a magnetic dipole antenna.

A magnetic dipole antenna is made up of an inductor coil and a capacitor, forming an LC circuit. This circuit is illustrated in Figure 4.1. To improve the reception of the LC circuit, a ferrite rod is typically inserted into the inductor coil, creating a ferrite rod antenna.

For optimal antenna performance, it's crucial to match the resonant frequency of the antenna with the frequency of interest. With LC circuits, the resonant

frequency can be tuned by varying the inductance of the inductor and the capacitance of the capacitor used in the circuit. The inductance, capacitance, and resonant frequency are related through Equation 5.2.

$$f_r = \frac{1}{2\pi\sqrt{LC}} \tag{5.2}$$

It is easy to control the capacitance in Equation 5.2. However, the inductance of a coil can be affected by various factors, including the number of windings, the size of the coil, the type of core material used, and the length of the coil. To obtain an approximate value of the inductance, one can utilize Equation 5.3[12].

$$L = \mu_{rod} F_L \mu_0 \frac{N^2 A}{l_c} \tag{5.3}$$

Equation 5.3 involves various parameters. These parameters include the permeability of the ferrite rod represented by μ_{rod}, the number of windings in the coil represented by N, the crossectional area of the coil represented by A, and the length of the coil represented by l_c. Finally, it should be noted that the value of F_L is a factor dependent on the relationship between the length of the ferrite rod and the length of the coil, l_c. Typically, F_L is determined through empirical methods.

The creation of the antenna was done by manually winding turns of a thin, enameled copper wire around a ferrite rod. For the antenna core, a ferrite rod with a permeability of 25 was chosen, suitable for frequencies between 0.2 and 5 MHz. This is ideal for avalanche beacons, which transmit signals at a frequency of 0.457 MHz. To evaluate the accuracy of Equation 5.3, I built three antennas using the same ferrite material but with varying numbers of windings: twenty, fifty, and one hundred. Afterward, I measured the inductance of all three antennas by employing an LCR meter. In Table 5.1, both the measured inductance and the estimated inductance calculated using Equation 5.3 is displayed.

Number of windings	Estimated inductance	Measured inductance
20	$14.03\mu H$	$\sim 23.6\mu H$
50	$87.69\mu H$	$\sim 106\mu H$
100	$350\mu H$	$\sim 355\mu H$

Table 5.1: Comparison of the estimated inductance with the measured inductance of wound antennas. All the antennas were coiled around a 45x8mm ferrite rod that has a permeability of 25. When estimating the inductance, the empirical factor F_L was set to 1

The results shown in Table 5.1 show a significant difference in the estimated inductance from Equation 5.3 and the real-world inductance of the antennas. Hence, Equation 5.3 is only suitable for estimating values and should not be relied upon when designing an antenna for a particular resonance frequency.

For the first try at making an antenna, I opted for a one-hundred-turn antenna. I made two antennas and gauged their inductance using an LCR meter. The inductance was measured as $355\mu H$ for the first antenna and $358\mu H$ for the second antenna. I utilized Equation 5.2 to calculate the required capacitance for the antenna to operate at a resonance frequency of $457kHz$. To achieve a resonance frequency of 457 kHz with an inductance of $355\mu H$, Equation 5.2 indicates that $341pF$ of capacitance is necessary. With an inductance of $358\mu H$, the ideal capacitance would be $338pF$.

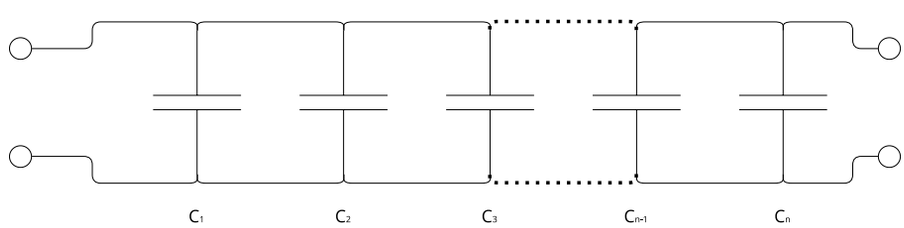


Figure 5.2: Capacitors connected in parallel have an additive effect on the total capacitance of the circuit. The total capacitance of the circuit in this figure is given by Equation 5.4.

$$C_{tot} = C_1 + C_2 + C_3 + \ldots C_{n-1} + C_n \tag{5.4}$$

No capacitors with the precise capacitances needed were available at the time of construction. The closest one available had a capacitance of $300pF$. However, when capacitors are connected in parallel, as shown in Figure 5.2, the overall capacitance of the circuit increases by an additive amount, as described by Equation 5.4. In order to get closer to the ideal capacitance, I utilized two capacitors - one had a capacitance of $300pF$, and the other had a capacitance of $30pF$, giving a total capacitance of $330pF$ as per Equation 5.4. With the capacitors available, $330pF$ was as close to the desired capacitance of $338pF$ and $341pF$ as I could get.

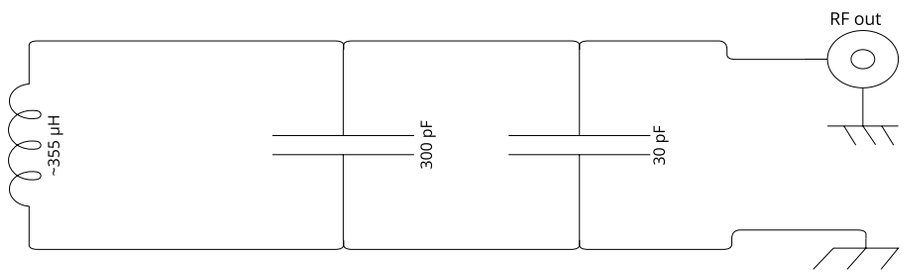
In Figure 5.3a, the completed circuit design for the passive antenna is displayed. An SMA connector was incorporated into the circuit to enable connection to a software-defined radio, such as the Airspy HF+ Discovery. I got help to solder

the antenna and capacitors onto a prototype PCB board to create the entire circuit. Quick-connect sockets were added to the capacitor locations to allow for easier swapping of capacitors.

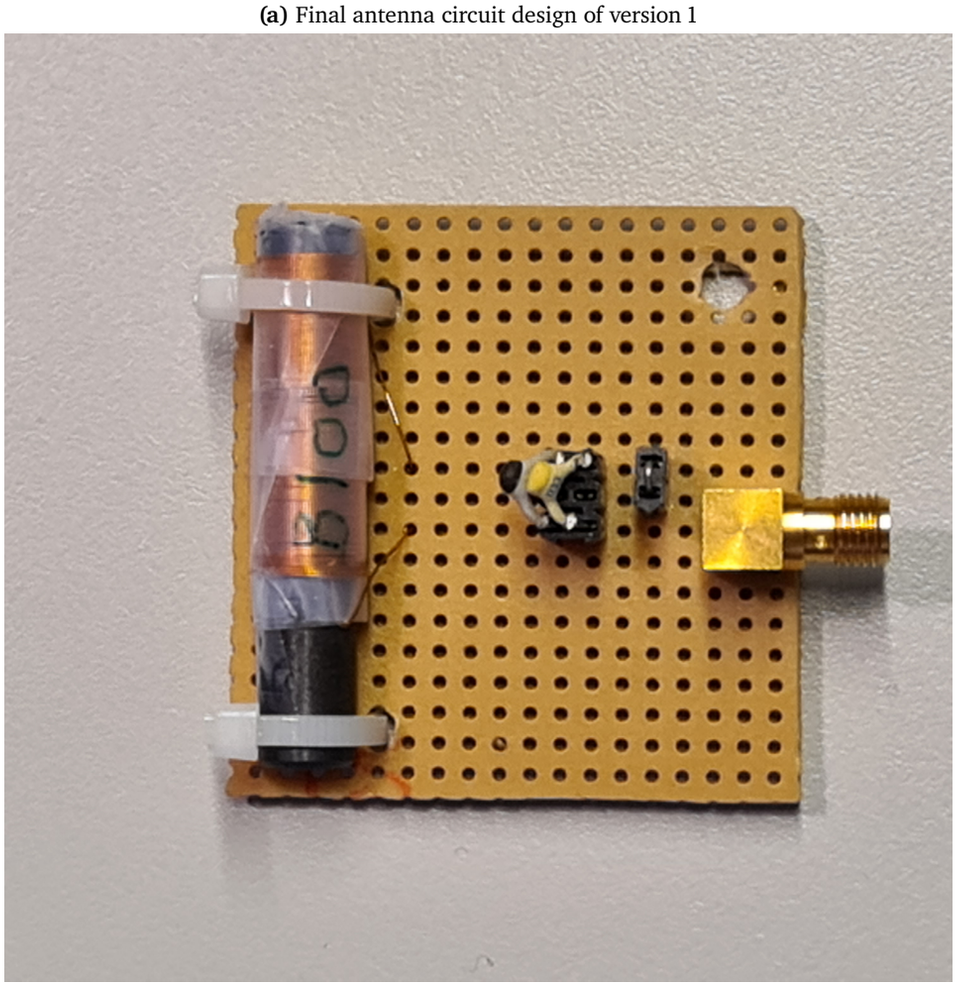
According to Equation 5.2, a capacitance of $330pF$ and an inductance of approximately $355\mu H$ will result in an antenna with a resonance frequency of roughly $464kHz$. Due to limited capacitors during the implementation of the antenna, the only option to adjust its resonance frequency was by modifying the inductance. This involved adding or removing windings from the inductor. However, the inductance formula (Equation 5.3) was found to be unreliable, so adjustments would have to be made through trial and error. In my preliminary experiments, I decided that relying on trial and error was too tedious. So, I formulated a hypothesis that the antenna, with a resonance frequency of $464kHz$, would have a broad enough bandwidth to function effectively at $457kHz$ too. To verify this hypothesis, I utilized a network analyzer to assess the return loss of the antenna.

In antenna design, return loss is a crucial parameter that measures the power reflected back from an antenna caused by impedance mismatches. The measurement involves connecting the antenna to a signal generator and transmitting a known signal (the incident wave) through the antenna. The antenna then reflects the signal back. Return loss is expressed in decibels (dB) and measures the ratio between the power of the incident wave and the power of the reflected wave. A lower return loss indicates better matching between the antenna and the transmission line or system it is connected to, resulting in more efficient power transfer. High return loss implies a significant amount of power being reflected back, leading to decreased antenna performance, signal degradation, and potential interference. Measuring the return loss of an antenna at different frequencies gives a good indication of what frequencies the antenna is suited for[17, ch. 2].

The return loss value itself is unitless, and this means that it only makes sense in relation to something else. For instance, to evaluate the efficiency of two antennas operating at a particular frequency, one can compare their return loss at that frequency. Assessing an antenna's effectiveness across different frequencies is another way to make use of return loss. This is particularly helpful for those who depend on the antenna to eliminate certain frequencies, as frequencies with high return loss will have less influence on the received signal. Lastly, measuring the return loss over several frequencies can give indications of the bandwidth of the antenna. If the antenna has a high return loss for most frequencies and a low return loss for very specific frequencies, then the antenna will express a very narrow bandwidth around those frequencies. Whereas, if the antenna has low return loss for several frequencies around a center frequency, the antenna will exhibit a wide bandwidth around that center frequency. Fig-



(a) Final antenna circuit design of version 1



(b) Circuit from Figure 5.3a, implemented on PCB.

Figure 5.3: Final antenna circuit design and its real-world implementation

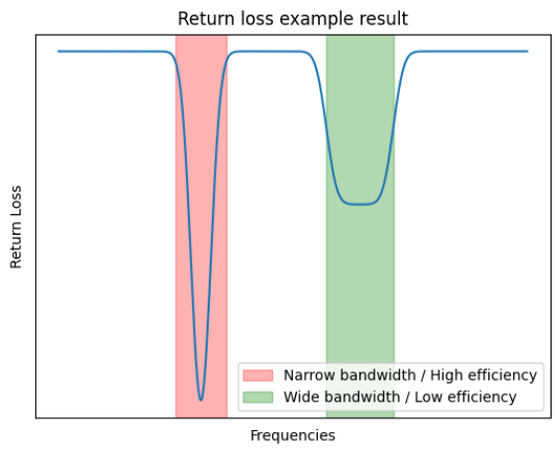


Figure 5.4: Fabricated example of return loss at various frequencies for an antenna. The red-shaded area highlights that the antenna has high efficiency for a specific frequency but limited bandwidth around it. On the other hand, the green-shaded area shows a broader bandwidth, albeit with lower efficiency. The example is left unitless as it is the relative return loss between frequencies that are important in this example.

ure 5.4 provides an example of a fabricated return loss result across several frequencies to illustrate this concept.

To measure the return loss of the antenna that was constructed, a handheld network analyzer was utilized. The analyzer is capable of conducting a sweep within a specific frequency range and measuring the return loss for each frequency. This allows the user to visually assess the return loss across multiple frequencies at once. Figure 5.5 shows a photo of the outcome of such a measurement. The handheld network analyzer showed the results of the return loss measurements in real-time and the data was not stored for later inspection. Instead, the return loss was inspected visually at the time, and the result showed a prominent dip in return loss near the desired frequency of $457khz$. Because of the project's objective of utilizing digital signal processing, I decided to utilize the antenna for initial experiments despite the fact that the lowest return loss dip was not at the exact frequency of $457kHz$.

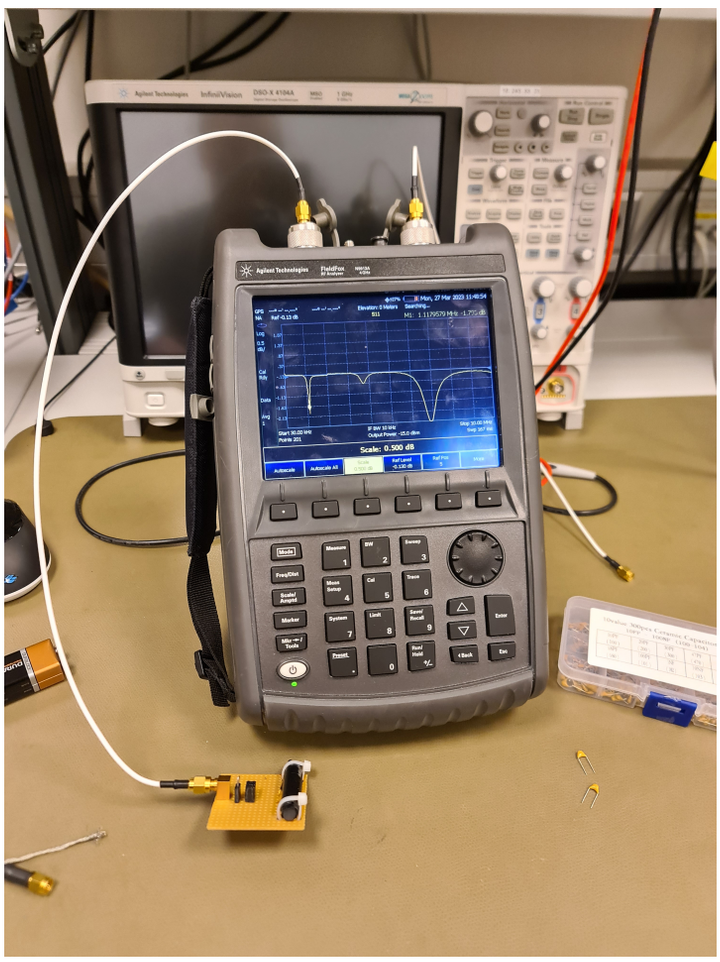


Figure 5.5: To measure the return loss of an antenna, a handheld spectrum analyzer is used. The analyzer sweeps across a designated frequency range and records the return loss at each point. Dips in the yellow line indicate low return loss. The width of these dips corresponds to the antenna's bandwidth at that frequency range.

5.1.2 System design

Once the two antennas were made, the subsequent task in the project involved assembling a system that could receive and analyze unprocessed radio signals to detect the transmission from an avalanche beacon. The task was ultimately completed by connecting the two antennas to two separate Airspy HF+ Discovery SDRs, as illustrated in Figure 5.6. These SDRs can be linked to a standard computer or laptop via a USB interface. The signal could then be measured from both antennas simultaneously, which is important for being able to estimate the magnetic field direction and the distance from a transmitting avalanche

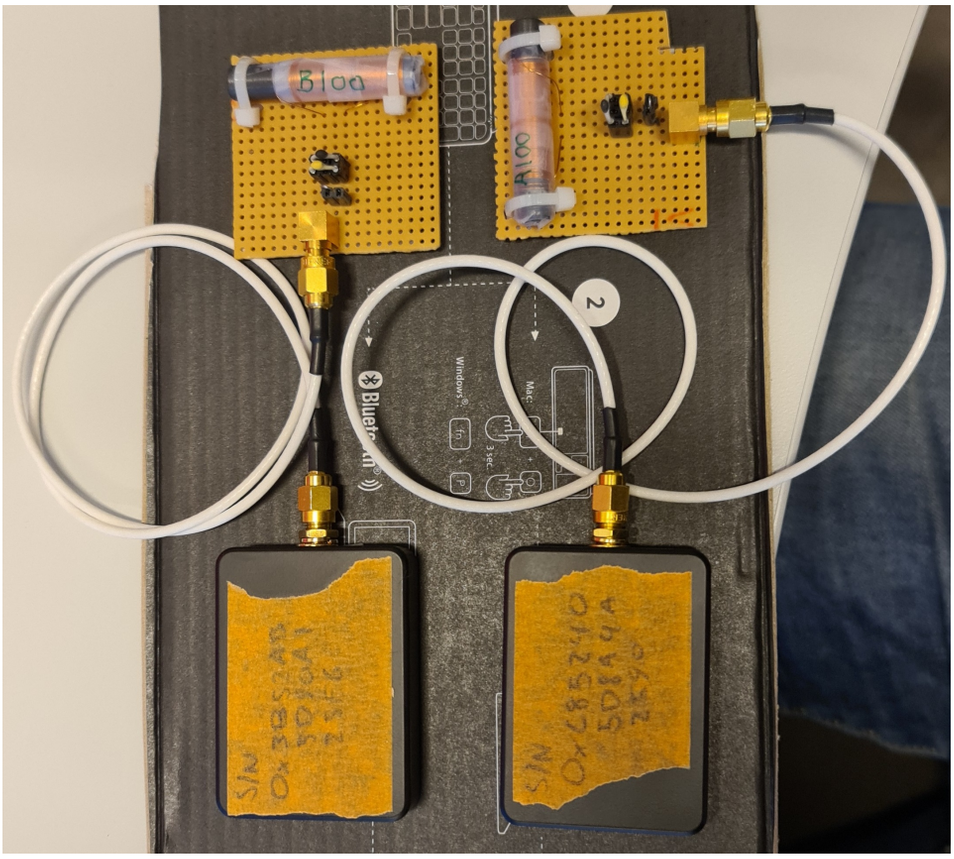


Figure 5.6: The initial design of the system: two antennas are connected to two separate Airspy HF+ Discoverys, which can be connected to a computer with a USB for conducting measurements.

beacon. Measuring the signal from both antennas simultaneously is crucial because the relationship between the signal power received by both antennas at a specific time is utilized to calculate the direction and distance of a transmitting beacon, as discussed in section 2.1.

To retrieve the signal from the SDRs, Airspys' own software was used[1]. With this software, users can customize the way the SDR receives data. They can set the center frequency for receiving data and choose whether to save the received data in a file, among other options. All the receiving software options for Airspys are specified in listing 5.1. When working on this project, the key parameters to consider are the center frequency, sample rate, usage of automatic gain control, and the option to incorporate a low-noise amplifier before signal capture.

```
1 airspyhf_rx
```

1. https://github.com/airspy/airspyhf/tree/master

```
Usage:
  -r <filename>   Receive data into the file;
        stdout emits values on standard output
  -s <serial number>  Open device with specified 64bits serial
    number
  -f <frequency>    Set frequency in MHz between 9 kHz - 31 MHz
     or 60 - 260 MHz
  -a <sample_rate>  Set sample rate, at the moment only 768 kS/
    s supported
  -n <#samples>   Number of samples to transfer (default is
    unlimited)
  -d      Verbose mode
  -w      Receive data into file with WAV header and automatic
    name
        This is for SDR# compatibility and may not work with
    other software
        It works even with HDSDR
  -g on|off   HF AGC on / off
  -l high|low   HF AGC threshold high / low (when AGC On)
  -t <value>    HF attenuator value 0..8 (each step increases 6
     dB the attenuation)
  -m on|off   on to activate LNA (preamplifier): +6 dB gain -
    compensated in digital.
  -z      Do not attempt to use manual AGC/LNA commands
        (useful in order to avoid errors with old firmware)
```

Listing 5.1: The usage page for airspyhf_rx

The sample rate is important as it dictates the bandwidth that the SDR will receive. The size of the bandwidth is dictated by the Nyquist-Shannon sampling theorem (Theorem 2.2.1), which states that when reconstructing a signal from discrete sample points, the maximum frequency that can be fully reconstructed is limited. More specifically, the maximum frequency of the signal that can be fully reconstructed is double the sampling rate used.

However, as discussed in section 2.4, an SDR can reconstruct a band-limited signal of a frequency higher than twice the sample rate through a process known as down conversion. How down conversion works is discussed in detail in section 2.4. However, the result of down conversion is shifting a high-frequency signal to a lower-frequency range. Therefore, the sample rate dictates the bandwidth of frequencies one can pick up before experiencing aliasing, and the center frequency dictates at what frequency the center of this bandwidth will be.

Currently, The Airspy HF+ discovery has a fixed sample rate of $768ksps$, which by Theorem 2.2.1 gives a bandwidth of $384kHz$, but the center frequency can be any frequency in the range $9kHz$ to $31MHz$ or 60 to $260MHz$. This bandwidth is sufficient for this project as avalanche beacons only transmit on one specific frequency.

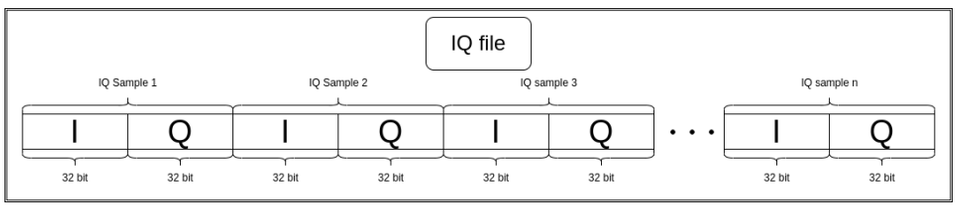


Figure 5.7: Layout of a binary I/Q file

The Airspy HF+ discovery has the ability to do automatic gain control. Automatic Gain Control (AGC) is a signal processing technique that adjusts the gain or amplification of a signal in real-time to maintain a consistent output level despite variations in input signal strength. AGC plays a crucial role in a wide range of applications, including audio systems, radio receivers, and communication devices. By automatically adjusting the gain, AGC ensures that the signal remains within an optimal operating range, preventing distortion due to under-amplification or saturation caused by over-amplification. This is especially useful in situations where information is encoded in the amplification of a signal, such as many digital modulation schemes.

However, automatic gain control impacts the dynamic range of the signal. AGC adjusts the gain to maintain a constant output level, which means that high-amplitude signals are attenuated while low-amplitude signals are amplified. As it is the measured power of the transmitted signal from the beacon that allows for distance and direction estimation, the AGC will be turned off for this project.

The last parameter of the Airspy software to consider is the ability to turn on a Low-Noise Amplifier(LNA). A low-noise amplifier is an electronic component that has the ability to amplify a low-power signal without much degradation of the signal-to-noise ratio of the signal. When using an electronic amplifier, the signal and noise power are both increased. However, the amplifier will also introduce further noise. An LNA is designed specifically to leave this additional noise at a minimum. With the LNA turned on when receiving signals, it should be possible to detect the beacon from further away.

When writing data to a file, the Airspy software represents complex signal data using I/Q data. Section 2.2.1 delves into the detailed benefits of utilizing I/Q data for signal processing. In essence, the utilization of I/Q data enables the separation of amplitude and phase information, thereby enhancing signal processing capabilities. The output file generated by the Airspy software is binary, containing consecutive I/Q samples. The I/Q data is displayed using two 32-bit floating-point numbers - the first number signifies the real part (I) of the I/Q data, while the second number signifies the imaginary part (Q). The

data is stored one after the other in the file, resulting in a consecutive sequence of I/Q samples. An outline of the resulting file is illustrated in Figure 5.7.

5.2 Signal detection algorithm

Once I had a working antenna and a system for receiving and storing radio signals, I needed to implement an algorithm that could analyze the received signals and detect whether or not an avalanche beacon was transmitting a signal at the time of measurement.

I devised two such algorithms: one focused on time domain analysis of the signal and one focused on frequency domain analysis. However, although the algorithms differ in which domain the algorithms do the analysis, both rely upon the user deciding a threshold value for detecting the signal.

5.2.1 Time domain signal detection

The time domain algorithm defines noise as the received signal power between pings from the avalanche beacon and signal power as the received power during the pings from the avalanche beacon.

The assumption from the previous paragraph makes it trivial to calculate the signal-to-power ratio if the periodicity of the transmitting beacon is known; however, in real-world scenarios, the period will not be known. Therefore, the algorithm needs some method of detecting the pings of the beacon.

The algorithm consists primarily of three steps: Frequency shifting the received signal to DC, passing the frequency-shifted signal through a lowpass filter, and separating the noise and the beacon ping from the received signal.

In the first step of the algorithm, the received signal is shifted in frequency so that the desired frequency, which is 457kHz in the case of avalanche beacons, is brought to 0 Hz (DC). This shift is necessary because the lowpass filter in the next step degrades the signal strength in all frequencies except 0 Hz.

Frequency-shifting is a process that involves moving a signal from one location in the frequency domain to another. People often use frequency-shifting to bring a specific frequency range closer to DC or zero frequency. This process is simple in the frequency domain, as it only requires translating the received signal. However, in the time domain, frequency-shifting involves multiplying the signal by a complex sinusoid. Depending on the reception settings

of the software-defined radio, frequency shifting may not be required since the software-defined radio typically shifts the signal by a preconfigured amount during the receiving process.

The second step of the time-domain algorithm is to convolve the received signal with a narrow lowpass filter. The algorithm uses the lowpass filter to lessen the effect on received signal power from frequencies outside the frequency area of interest.

As the avalanche beacon only transmits its signal at 457khz and no other frequencies, it is beneficial to use a lowpass filter that is as narrowband as possible.

The European standard for avalanche beacons states that all approved beacons should transmit at 457KHz with a possible error of ±80Hz. This regulation suggests that a lowpass filter with a bandwidth of 160Hz is a good filter for this algorithm.

After the time domain algorithm has been applied to the signal, the final step is to check whether the power of the filtered signal is higher than a predefined threshold value. If the signal's power exceeds this threshold, the algorithm concludes that a beacon must be transmitting nearby.

The time domain signal algorithm implemented in python is shown in listing 5.2.

```
def detect_signal(signal,N,sample_rate,signal_threshold,
    cutoff_freq,frequency_shift = None, decimation_rate=1):

    #
    #Frequency shift signal to DC
    #
    t = np.arange(N)
    signal = signal * exp(2j * pi * t * sample_rate)

    #
    #Pass signal through lowpass-filter
    #
    #Create filter
    filter = firwin(numtaps=101, cutoff=cutoff_freq, fs=
    sample_rate)

    #Convolve filter with signal
    signal = convolve(signal, filter)

    #
    # Detect beacon signal
    #

```

```
22      #Calculate signal power
23      signal_power = signal**2
24
25      #Separate signal and noise based on signal threshold
26      beacon_signal_power = where(signal_power > signal_threshold
        )
27      noise_power = where(signal_power < signal_threshold)
28
29      #If either beacon_signal_power or noise_power is empty no
        beacon signal was detected
30      if len(beacon_signal_power) > 0 and len(noise_power > 0):
31          return True
32      else:
33          return False
```

Listing 5.2: The time domain signal detection algorithm

5.2.2 Frequency domain analysis

The second algorithm I developed for the first version of the beacon focused exclusively on the frequency domain for signal analysis. The frequency domain algorithm is quite similar to the time domain algorithm. The only difference is that instead of estimating the noise power based on the signal strength between beacon signals, the algorithm estimates the noise as the maximum power of the frequency spectrum close to the frequency area of interest. And the signal power is the frequency spectrum's power in the frequency area of interest.

With the definitions above for signal and noise power, the estimated value will vary depending on two parameters. The first is the width of the frequency area of interest, and the second is the width of the area outside the area of interest used for noise estimation. The width of the frequency area of interest, the first parameter, should be set to $2 \cdot 80$ Hz because it is the allowed frequency error of avalanche beacons according to the European standard. For the second parameter, the width of the noise area, you need to choose it interactively. However, it is recommended to keep it as narrow as possible.

Once both parameters are decided, the steps of the frequency domain algorithm are as follows.

First, the algorithm calculates the frequency spectrum of the received signal using the fast Fourier transform. Because the analysis happens in the frequency domain, applying a frequency shift or filter to the signal is unnecessary.

The second step is to find the maximum power of the noise by examining the frequencies closely surrounding the frequency area of interest and find the

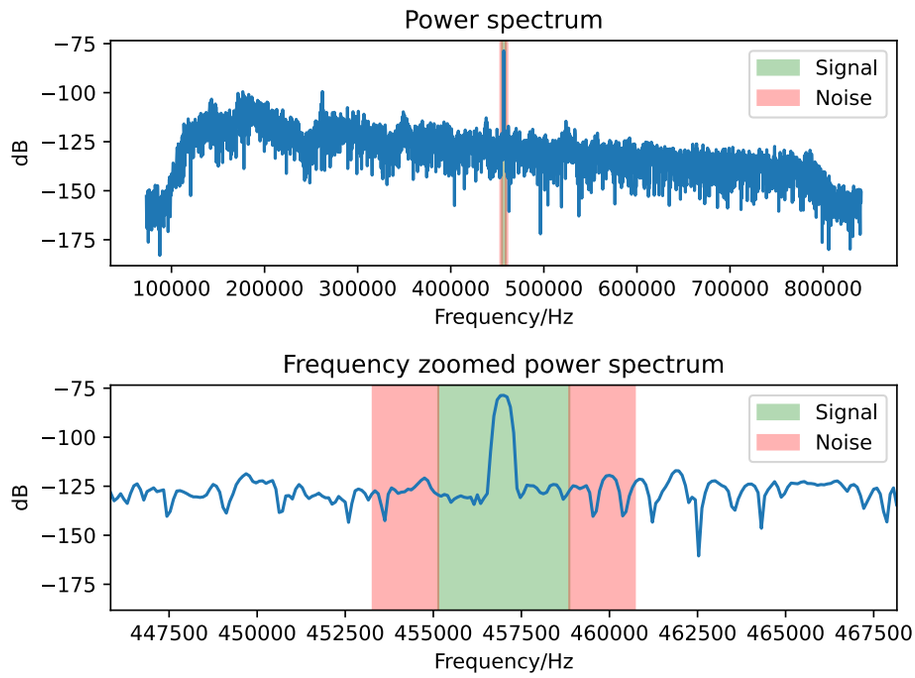


Figure 5.8: This image showcases both the power spectrum of a signal and a zoomed-in version of the power spectrum. This represents a cross-section of a dot found in the spectrogram shown in Figure 5.11. To estimate the noise at the frequency of interest, I use the red-shaded area. Meanwhile, the green-shaded area is utilized to estimate the signal.

signal's maximum power by examining the frequency area of interest. Figure 5.8 depicts how this estimation is done by looking at the power spectrum at a timestep where an avalanche beacon was transmitting.

As with the time domain algorithm, the frequency domain algorithm detects avalanche beacons by thresholding the difference between the noise and signal power. If the difference between the signal and the noise is beyond an interactively chosen threshold the algorithms concludes a avalanche beacon is nearby.

Listing 5.3 shows the frequency domain algorithm implemented in Python.

```
1 def frequencydomain(signal,signal_center, signal_width,
    noise_width, threshold):
2     #
3     # Calculate spectrogram
4     #
5     spec = spectrogram(signal)
6
```

```
    #
    # Find frequency area of interest
    #
    signal_min = signal_center-signal_width
    signal_max = signal_center+signal_width

    #
    # Find maximum signal strength measured
    #
    signal_amp = np.max(spec[signal_min:signal_max])

    #
    # Find noise estimation area and maximum noise level
    #
    noise_amp_left = np.max(spec[signal_min-noise_width:
    signal_min])
    noise_amp_right = np.max(spec[signal_max:signal_max+
    noise_width])
    noise_amp = noise_amp_left if noise_amp_left >
    noise_amp_right else noise_amp_right

    #
    # If the Signal-to-noise ratio is above a certain threshold
     then assume a beacon is transmitting
    #
    if signal_amp / noise_amp > threshold:
        return True
    else:
        return False
```

Listing 5.3: The frequency domain signal detection algorithm

5.3 Evaluation

When undertaking a project like this, it's advantageous to have a swift representation of the frequencies received by an antenna. These visualizations enable the user to promptly identify and correct minor mistakes without requiring extensive analysis.

For this purpose, I opted to utilize the GQRX tool[2]. GQRX is a popular open-source software-defined radio application used for exploring, analyzing, and visualizing radio frequencies. It provides a user-friendly interface and supports a wide range of SDR hardware. GQRX offers real-time spectrum analysis, demodulation, and visualization of various modulation schemes, including AM, FM, SSB. With its flexible and customizable settings, GQRX allows users to

2. https://gqrx.dk

tune into different frequencies, adjust bandwidth, apply filters, and visualize the received signals in a spectrum display.

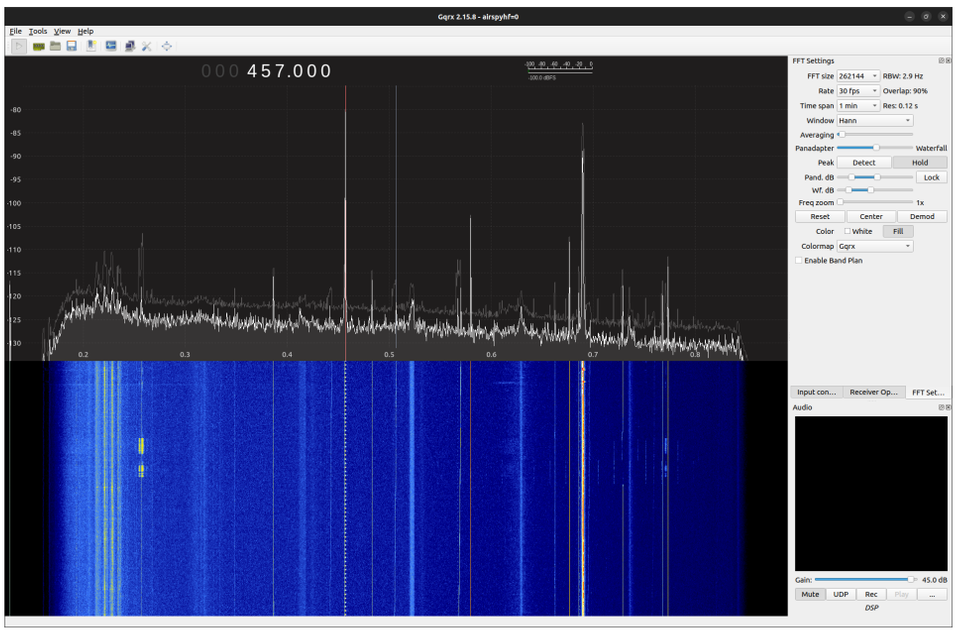
Thanks to GQRX, I could effectively assess an antenna's performance, gauge the radio interference in a specific setting, and asses the maximum distance for receiving an avalanche beacon signal, all in real-time. Figure 5.9 displays two snapshots from the GQRX app. GQRX was a valuable tool to have before running experiments, as it allowed me to inspect the system for errors visually. As previously stated, GQRX can also demodulate various signals. This feature proved helpful as it enabled me to convert the radio signal from a transmitting beacon into an audible tone. Consequently, I was able to inspect the system both visually and audibly.

In order to estimate the feasible range of my system's beacon detection, I would utilize a typical wireless headset connected to my laptop and leverage the demodulation capability of GQRX. By maintaining a fixed position for my laptop and the antenna, I could carry an avalanche beacon and listen for its signal through the wireless headset while moving away. The distance at which the audible tone fades into white noise would serve as an indicator of the range. This proved to be helpful in designing the initial experiments, as it provided insight into the meaningful ranges to conduct experiments on.

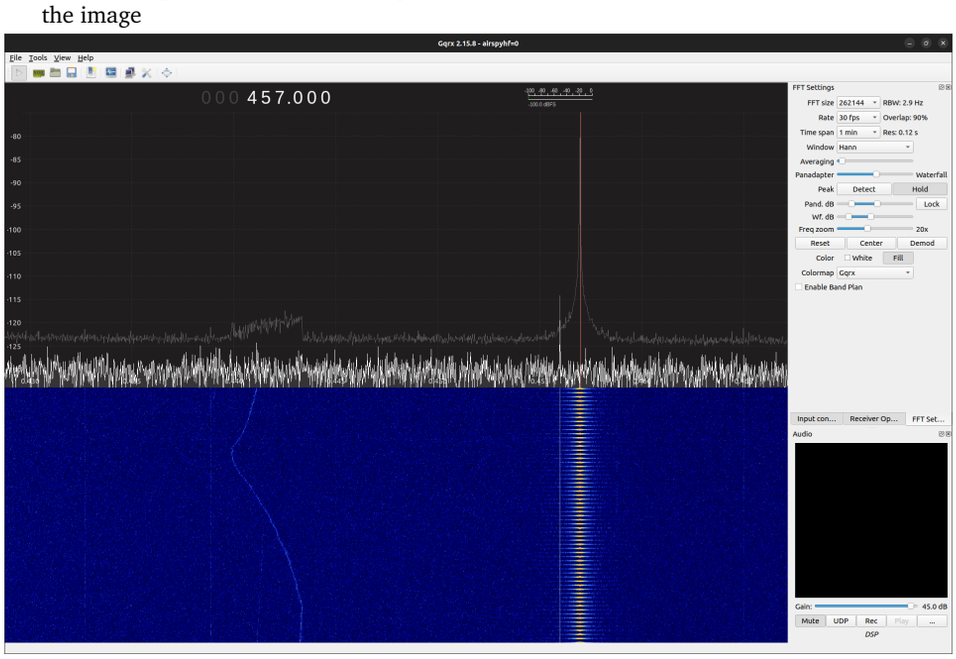
5.3.1 Initial experiments

Initial experiments were conducted within an office setting to verify the feasibility of digital techniques in detecting avalanche beacon signals and to identify any issues with the current system. Two separate experiments were conducted: one aimed at identifying the signal degradation with respect to distance and the other at observing the impact of rotating the transmitting beacon in relation to the receiver.

I conducted experiments to analyze the degradation of the signal with respect to distance. The transmitter beacon was placed at various distances from the receiver, with the maximum distance limited to five meters due to office space constraints. The measurements were taken at one-meter intervals, giving five sample points. Originally, I planned to measure each distance for ten seconds, which would have resulted in $768 \cdot 10^4$ samples for each distance. However, as I conducted my experiments, I discovered that the Airspy HF+ rounds up the number of samples to the nearest power of ten, resulting in 10^5 samples being taken and a measurement period of thirteen seconds. The avalanche beacon emits a pulse that lasts at least 70 milliseconds every second. Consequently, thirteen pulses from the transmitter should be detected after the thirteen-second measurement period. The angle of the avalanche beacon to



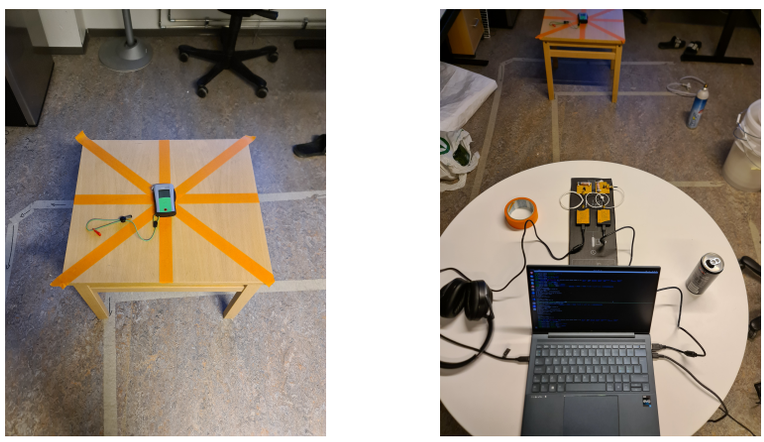
(a) GQRX displaying the full bandwidth of frequencies received from an Airspy HF+ Discovery SDR. The signal from a transmitting avalanche beacon can be seen beneath the red line in the image



(b) GQRX zoomed in to the frequency of interest. The signal from a transmitting avalanche beacon can be seen beneath the red line.

Figure 5.9: Two screen captures from the GQRX software. Both images were captured whilst listening for an avalanche beacon. In this case, the signal is strong enough to be easily observed visually.

the receiver remained consistent for every distance. This angle positioned the beacon parallel to one of the receiver's antennas and perpendicular to the other. Based on Equation 2.5, I anticipated that the signal strength received by the parallel antenna would be significantly higher compared to the perpendicular antenna.



(a) I placed a Mammut Element avalanche beacon on a table. I then gradually moved the table away from the receiver, one meter at a time.

(b) The receiver consisted of two antennas at right angles from each other connected to two different SDRs connected to a laptop for measurements.

Figure 5.10: Photos from the initial experiment. Figure 5.10a is a photo of the avalanche beacon that was used, and Figure 5.10b is a photo of the receiver setup.

Another experiment was conducted using a fixed distance but with a rotating beacon. The measurements were taken for each rotation, with a similar recording time of about 13 seconds. The beacon initially started parallel to one antenna, then rotated by 45-degree intervals for each measurement. This process was repeated for a half rotation of the antenna, providing sample points at 0, 45, 90, 135, and 180 degrees.

In the field of signal processing, the energy of a discrete signal is defined by the sum of the squared samples from the signal. Equation 5.5 demonstrates this concept. The signal, denoted as $x[n]$, represents the measured magnetic field around the receiving antenna. As explained in section 2.1, this value is determined by the equation $H_{antenna} = H \cos \alpha$ (Equation 2.5), where H represents the value of the magnetic field created by the transmitting beacon. This value can be calculated using the equation $H = \frac{M}{4\pi r^3}\sqrt{1 + 3\cos^2 \theta}$ (Equation 2.4). Based on this information, it is reasonable to expect that the estimated signal energy would be dependent on H^2 and would degrade by r^6.

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 \tag{5.5}$$

5.3.2 Analysis

The initial process for analyzing the data obtained from the experiments discussed in the previous section involves computing a spectrogram of the data. A spectrogram is a visual representation of the frequencies present in a signal over time. It is created by analyzing the spectrum of a signal and displaying the intensity of each frequency as a function of time. They provide valuable insights into a signal's frequency content and temporal characteristics. The spectrogram of the signal with the beacon at a one-meter distance is displayed in Figure 5.11. The spectrogram displays small yellow dots representing the signal occurring every second, while the remaining parts indicate noise occurring at different frequencies. When detecting a signal automatically, the crucial factor is the correlation between the signal's strength and the accompanying noise's strength. This correlation is referred to as the Signal-to-Noise Ratio (SNR), which is determined by Equation 5.6.

$$SNR = \frac{P_{Signal}}{P_{Noise}} \tag{5.6}$$

To calculate the SNR using Equation 5.6, one must estimate signal and noise power, as they are rarely known beforehand. To estimate the signal and noise power, I changed the output of the signal detection algorithms so that they returned their estimated signal and noise power instead of whether they detected an avalanche or not. However, during the preliminary tests, I discovered that the time domain algorithm was inconsistent in the measured signal or noise power over distance and did not produce reproducible results. Therefore, I did not use the time domain algorithm in the experiments of this thesis.

5.3.3 Results

The following section will present the outcomes of the experiments and analyze how they align with the hypotheses formulated prior to conducting the experiments. The signal-to-noise ratio degradation in relation to the distance from the receiver is illustrated in Figure 5.12. Additionally, Figure 5.13 showcases the measured signal-to-noise ratio concerning the rotation of the transmitting beacon. For a detailed view of individual plots from each distance and rotation of this study, refer to Appendix A.

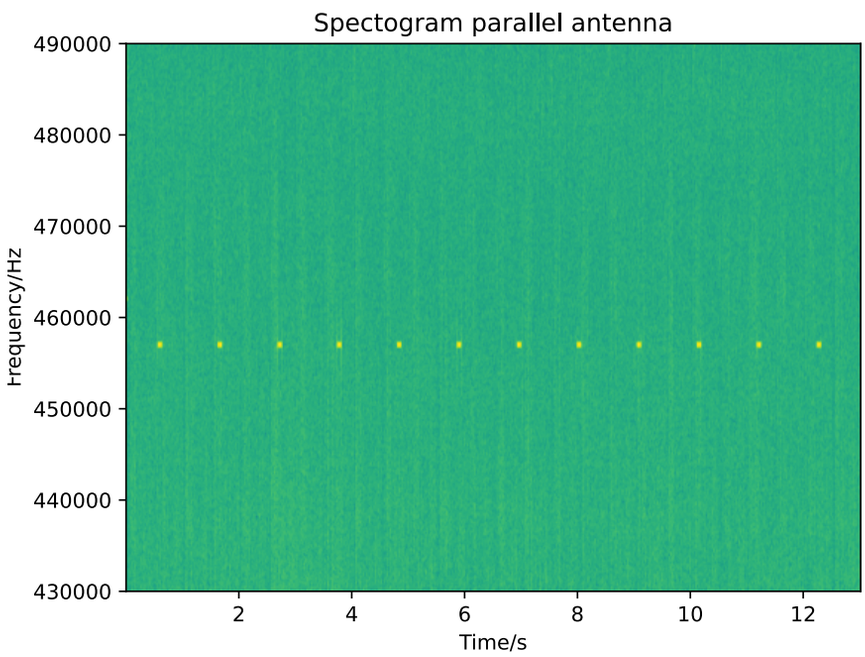


Figure 5.11: The spectrogram shows radio signals received when the beacon is placed one meter away. The yellow dots on the spectrogram indicate the signal from the beacon, occurring once every second.

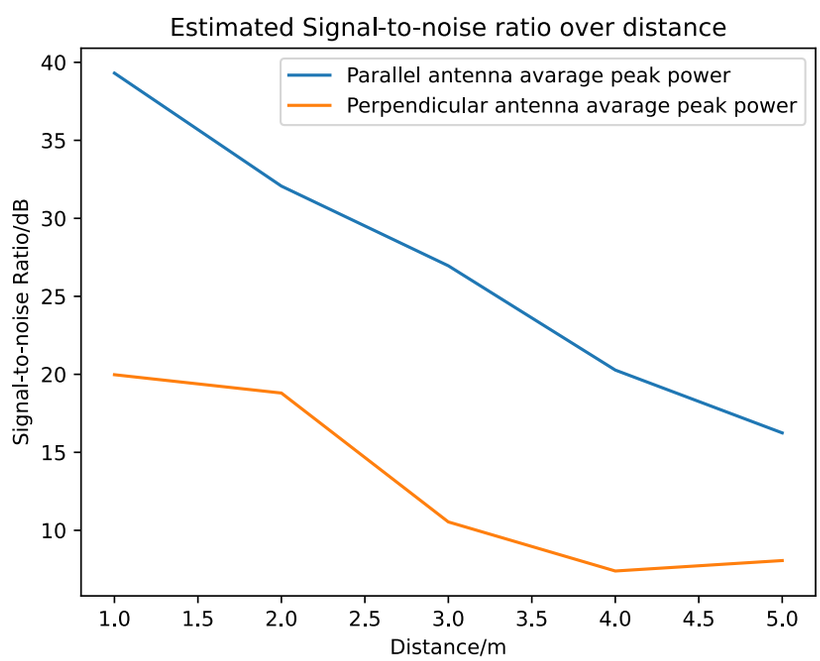


Figure 5.12: The degradation of estimated Signal-to-noise Ratio as a function of distance from the transmitting beacon.

Prior to conducting the experiments, I had two expectations. Firstly, I anticipated that the signal strength would deteriorate rapidly as the distance increased. This is due to the fact that signal power is linked to the magnetic field, which degrades at a rate of the cube of the distance. Secondly, I anticipated that the antenna positioned perpendicular to the transmitting beacon would receive a considerably weaker signal compared to the parallel antenna. The experiment supported both of these expectations. Figure 5.12 displays the resulting signal-to-noise ratio for both antennas over the distance. It appears that there is a significant discrepancy of approximately twenty decibels between the two antennas. This suggests that the signal received by the parallel antenna is approximately one hundred times stronger than the signal received by the perpendicular antenna. The first three distances show this pattern accurately. Beyond that, the graph of the perpendicular antenna levels off. Nevertheless, examining the distance plots in Figures A.4 and A.5 separately shows that the signal is no longer detectable in these plots, resulting in a flawed signal-to-noise ratio mostly defined by noise. Based on the experiments conducted with the parallel antenna, it can be concluded that the signal weakens by around six decibels per meter. This means there is a four-fold decrease in signal strength for every meter. The findings support the hypothesis that the signal degrades at a rapid pace. However, the hypothesis that the signal weakens proportionally to the sixth power of the distance is not validated by the consistent degradation observed in the experiments.

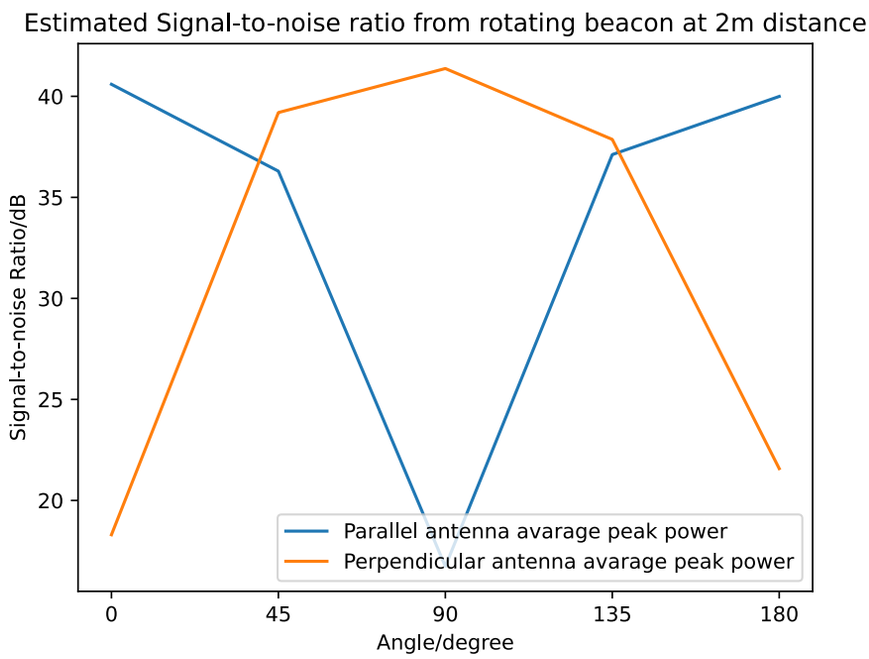


Figure 5.13: The degradation of estimated Signal-to-noise ratio as a function of beacon rotation.

In the second experiment, I examined how the received signal strength changes based on the position of the transmitting beacon in relation to the receiver's orientation. I conducted this experiment to verify the hypothesis that the strength of the received signal is dependent on the cosine of the angle between the receiving antenna and the magnetic field, as explained by Equation 2.5. The measured signal-to-noise ratio for both antennas is displayed in Figure 5.13. Based on the ninety-degree offset in the measured signal-to-noise ratio between the two antennas, the hypothesis seems plausible. However, the plots do not show a distinct sinusoidal pattern, and a more precise experiment with a higher resolution between each measurement may be required to draw a definitive conclusion.

5.4 Lessons learned

After conducting experiments with the first prototype, I have gained valuable insights that confirm the feasibility of using software-defined radios to receive signals from avalanche beacons. Some of my hypotheses were confirmed dur-

ing the experiments, although some confirmations were more conclusive than others. This section will highlight the significant findings from the initial prototype.

5.4.1 Range issues

Although the initial system was able to store data for future analysis, it failed to meet several requirements outlined in Section 1.1. Despite being able to detect a beacon at a distance of five meters during experiments, it was clear that it would be incapable of detecting signals from greater distances. This range is notably low compared to contemporary avalanche beacons, which can detect other beacons from a range of forty to seventy meters.

5.4.2 Hardware setup

Although the hardware setup used in this prototype version was initially meant to be a temporary solution, it's important to note the issues that were uncovered with this configuration. Initially, the cardboard plate serving as the attachment base for the antennas and software-defined radios posed a challenge in maintaining the perpendicularity of the antennas. Additionally, the requirement for a complete laptop connected to the system via USB cables impeded the mobility of the setup. These issues made conducting experiments with the system difficult and must be corrected in future versions.

5.4.3 Low-resolution experiments

The experiments yielded valuable insights, but their resolution was limited, making it difficult to draw definitive conclusions about the outcome. Furthermore, the experiments were conducted in a university office, which differs significantly from the typical environment where avalanche beacons are utilized. This discrepancy may pose a problem as the university houses a multitude of electronics that emit radio noise, particularly at low frequencies, which is the frequency range employed by avalanche beacons. From a visual inspection of the radio spectrum using GQRX, I found that computer screens generated lots of noise at the relevant frequency range. To ensure accurate results, it is advisable to conduct future experiments in a location with minimal radio interference. Moreover, it is crucial to increase the number of sample points to enhance the resolution of the experiment and arrive at valid conclusions.

5.5 Version 1.5

Before concluding this chapter, I would like to briefly discuss an attempt at a potential solution to the limited range of the antenna. According to Faraday's law, the current induced in an inductor is directly proportional to the number of windings in the inductor and the rate of change of magnetic flux over time (as shown in Equation 5.7).

$$\epsilon = -N\frac{\Delta\Phi_B}{\Delta t} \tag{5.7}$$

$$U = RI \tag{5.8}$$

To gauge radio signals, software-defined radios evaluate the variation in voltage across the antenna. Following Ohm's law (Equation 5.8), if the resistance of the circuit remains constant, a surge in current leads to a proportional boost in voltage. I hypothesized that by adding more windings to the antenna, I could improve its sensitivity by increasing the induced current and, consequently, amplifying the induced voltage.

In order to examine this hypothesis, I fashioned an antenna consisting of 300 windings. I utilized the identical ferrite core and copper wire I had previously employed for the 100-turn antennas. As the induced current is proportional to the number of windings, I gathered that this could improve reception by a factor of three. With 300 windings, Equation 5.3 estimates an inductance of $3.16mH$. The measured inductance was within the same order of magnitude. According to Equation 5.2, if the inductance is $3.16mH$ and the desired resonance frequency is $457kHz$, then the required capacitance is $38pF$.

After creating the coil and matching it with the appropriate capacitance, I utilized a network analyzer to examine the return loss of the antenna. Unfortunately, the results were not useful. Following discussions with a field expert, it was determined that when capacitance levels are this low, the parasitic capacitance between antenna components may become prominent, ultimately impacting the antenna's resonance frequency. Unfortunately, I was unable to create an antenna with 300 turns and a resonance frequency of 457 kHz due to the issue of parasitic capacitance. Therefore, I was unable to investigate whether increasing the number of turns would expand the antenna's receptive range. Eventually, I abandoned attempting to boost the range passively and opted to use the 100-turn antenna while amplifying the signal using alternative methods.

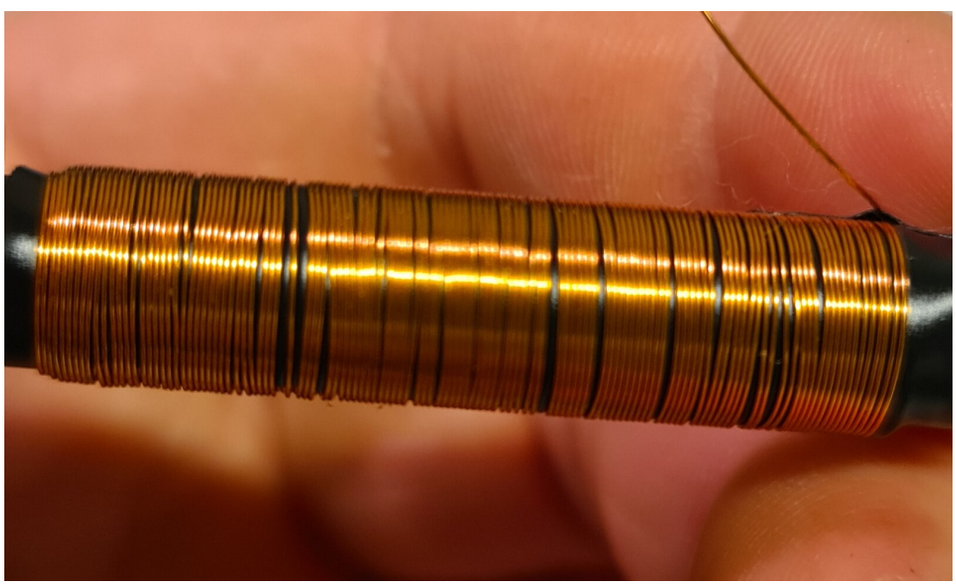


Figure 5.14: The 300-turn antenna utilized to investigate whether increasing the number of turns can enhance the antenna's receptive range.

6

Version 2

In this chapter, I will discuss the second attempt at developing a prototype for a software-defined radio avalanche beacon. While the first prototype discussed in the previous chapter proved the feasibility of the project, it had some limitations, particularly with regard to the antenna's receptive range.

To begin the chapter, I discuss a new algorithm for signal detection. Following that, the development of a new system with a new antenna will be described with the aim of making the receiver more portable and with a longer range. Subsequently, new experiments conducted with the new system will be outlined. Finally, the results from the experiments will be presented, and the new prototype will be discussed.

6.1 New signal detection algorithm

In the previous chapter, I presented two algorithms for detecting the presence of an avalanche beacon, one time-domain algorithm and one frequency-domain algorithm. The time-domain algorithm did not produce reproducible results, so I did not utilize it in my experiments. The frequency domain algorithm worked satisfactorily for strong signals, but this method needed to be improved in low signal-to-noise environments.

To improve on both the algorithms from version one, I combined them into an

algorithm that considered both the frequency and time domains when detecting a signal. Analysis in the frequency domain was particularly useful since avalanche beacons can have a frequency discrepancy of up to $80Hz$ from the required frequency of $457kHz$ [7]. On the other hand, the time domain analysis allowed for noise estimations at each specific frequency. I will outline each step of the new signal processing algorithm in the upcoming seven paragraphs. Figure 6.1 will illustrate each step using plots from the intermediate steps of the algorithm while listing 6.1 will present an implementation of the algorithm in Python.

To analyze the properties of the received signal, I calculate its spectrogram, which provides a visual representation of the signal's frequency content over time. I discard frequencies outside the designated frequency range to focus my analysis on specific frequencies of interest. For example, in this case, the frequency area of interest is $457kHz \pm 80Hz$. I narrow the examination to the relevant frequency area by excluding frequencies outside this range, enabling a more targeted analysis.

The first step of the algorithm is to estimate the noise level in each frequency bin. I calculate the median value within each bin to estimate these noise levels. This approach provides a measure of the central tendency of the noise within the frequency bin. By computing each bin's median, I estimate the noise levels associated with different frequencies, assisting in subsequent analysis and noise removal. One could also use the mean power value of each frequency bin to estimate the noise. However, I used the median because it is less affected by significant outliers that may occur due to the presence of a strong signal.

The next step is subtracting the median power level from the power levels of the whole frequency domain. This subtraction serves two purposes: it centers the noise around zero and removes constant tones. This step helps isolate the desired signal components from the background noise, enhancing the accuracy of subsequent processing and interpretation of the data. The subtraction of the noise median has a significant visual effect on the spectrogram, which can be seen in Figure 6.1d.

For the rest of the algorithm to work, it is essential to know how many time-samples of the spectrogram one can expect to find the transmitted signal of an avalanche beacon. One can calculate the number of time-samples following this reasoning:

Let N be the number of time samples that contain a signal. To calculate N, you need to determine the duration of each signal in seconds, denoted as T_{signal}, and the duration of one time-sample in the spectrogram, represented as ΔT. The count of time samples that include a signal within a measurement period

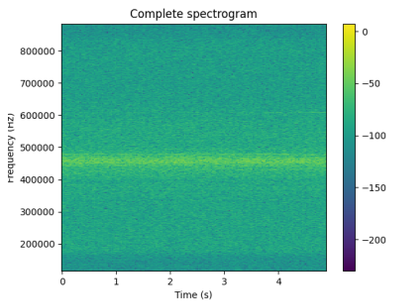
of $T_{measurement}$ can be calculated using the formula $N = T_{measurments}\frac{T_{signal}}{\Delta T}$. The European standard gives the duration of the blip for avalanche beacons as $70ms$[7].

To find the maximum signal power in each frequency bin, I sort the frequency bins based on their max power and average the N largest values in each frequency bin. This sorting enables me to identify and quantify the maximum power associated with each frequency bin.

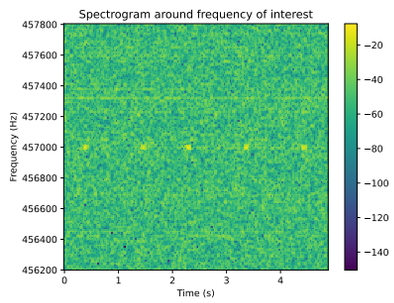
The next step is to estimate the standard deviation of the noise. Again, I use the median value to account for large outliers. By employing the median, I obtain a reliable standard deviation estimate. The standard deviation can be estimated using the median as per Equation 6.1.

$$\sigma_X \approx Median(X - Median(X)) \tag{6.1}$$

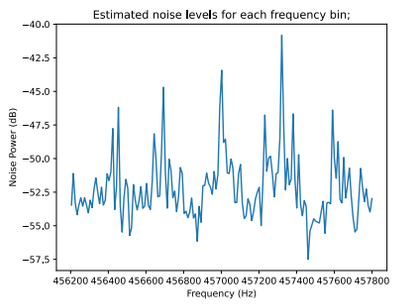
The whole frequency spectrum can then be normalized using this standard deviation estimate, resulting in a unit defined by the standard deviation. With the power spectrum defined through this unit of standard deviations, it is easier to decide upon a proper threshold for detection. The end result is a more accurate and reliable detection of the beacon signal than the analysis done during the previous experiment.



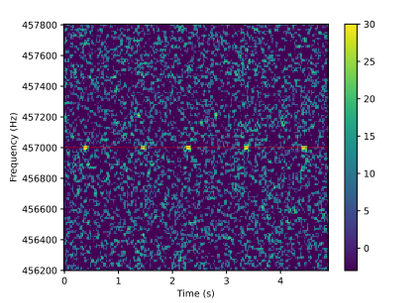
(a) Calculate the spectrogram of the received signal.



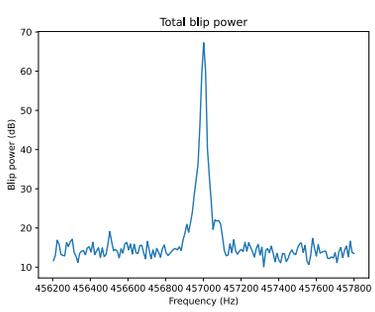
(b) Discard frequencies outside frequencies of interest, in this case, the frequency area of interest is $457kHz \pm 800Hz$.



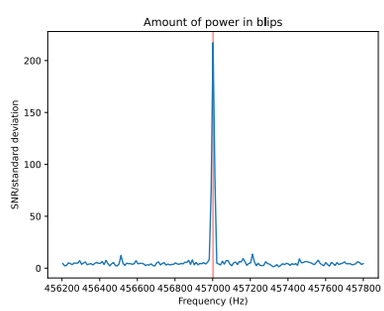
(c) Estimate the time domain noise levels for each frequency bin by calculating the median of each frequency bin.



(d) Remove the time domain noise levels from each frequency bin to facilitate further analysis.



(e) Let N be the number of time samples that contain a blip, sort the frequency bins on max power, and average the N largest values in each frequency bin. To find the maximum blip power on each frequency bin.



(f) Estimate the standard deviation using the median to account for large outliers. We can now express the beacon blip power in standard deviation units. This value can be used for signal detection by thresholding.

Figure 6.1: Illustrations for each step of the signal processing algorithm used for the analysis of the second experiment.

This analysis provides two valuable metrics for future comparison. Firstly, the maximum signal power detected can indicate the degradation of the signal over distance or angles. Secondly, the signal power expressed in standard deviation units can be used to compare the signal's detectability as it degrades.

A Python implementation of this new detection algorithm is provided in listing 6.1.

```
def hybridDomain(
    x,
    fs, # Sample rate of signal
    fc, # Frequency center of signal
    f, # Frequency of interest
    max_freq_dev=1000,
    fft_step=20000,
    nfft = 77000,
    plot = False
):
    fo = f - fc
    t_vec,f_vec,Sxx=spectrogram(x,sr=fs,nfft=nfft,step=fft_step
    ,wf=signal.hann(77000))


    # Extract only frequencies indices of interest
    fidx=np.where( np.abs(f_vec - fo) < max_freq_dev)[0]
    freq_vec2=f_vec[fidx]+500e3
    pwr=np.abs(Sxx[fidx,:])**2.0

    # # Estimate noise using median of signal

    # #Calculate SNR
    # # remove median power to remove constant tones
    for fi in range(pwr.shape[0]):
        # signal to noise ratio for each frequency bin
        pwr[fi,:]=(pwr[fi,:]-np.median(pwr[fi,:]))/np.median(
    pwr[fi,:])

    # detect center frequency of beacon signal, might be up to
    80Hz offset from 457 kHz
    # we know that the blips are within 70 ms
    duration=np.max(t_vec)-np.min(t_vec)
    # this is how many seconds there are in one time sample of
    the spectrogram
    dt=(fft_step/fs)
    # this is how many time samples contain a blip
    n_blip_samples=int((duration*70e-3)/dt)

    # go through each frequency bin
    # and count how much power is in blips
    total_blip_pwr=np.zeros(pwr.shape[0])
    for fi in range(pwr.shape[0]):
        # todo: could be improved by integrating power in one
    second segments
```

```
41          # sum the peak power of each one second segment,
    because each time
42          # sample is approximately the length of a blip (770
    samples)
43          idx=np.argsort(pwr[fi,:])
44          total_blip_pwr[fi]=np.mean(pwr[fi,idx[(pwr.shape[1]-
    n_blip_samples):(pwr.shape[1])]])
45
46      max_idx=np.argmax(total_blip_pwr)
47      # estimate standard deviation with median
48      std_estimate = np.median(np.abs(total_blip_pwr - np.median(
    total_blip_pwr)))
49
50
51      # amount of power from beacon divided by one standard
    deviation
52      beacon_blip_power_for_detection = total_blip_pwr[max_idx]/
    std_estimate
53
54      # unnormalized peak detected power (for mapping power as a
    function of position)
55      beacon_blip_power_for_mapping = total_blip_pwr[max_idx]
```

Listing 6.1: The new signal detection algorithm for version 2

Active antenna

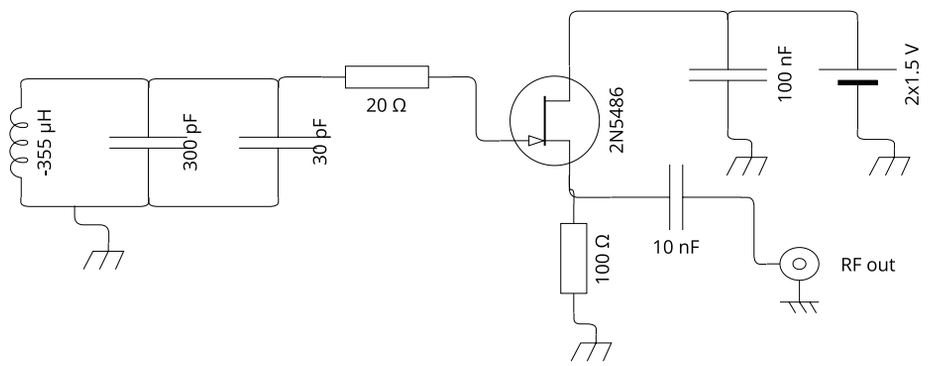
The biggest problem with the first prototype was the short receptive range it had. An attempt was made to expand the antenna's receptive range by adding more windings, but it was promptly abandoned during the implementation. In order for this project to be feasible, it's necessary for the receiver to have a receptive range comparable to that of modern avalanche beacons. Therefore, the primary goal of the new version is to improve upon the previous version's receptive range.

Upon consulting with a field expert, it was concluded that utilizing an active antenna would be the most effective course of action over a passive antenna. Passive and active antennas differ in their ability to receive and transmit wireless signals. A passive antenna acts solely as a receiver, converting electromagnetic waves into electrical signals without any additional amplification or signal processing. It relies on the received signal's strength to operate effectively. On the other hand, an active antenna not only receives signals but also incorporates amplification and signal processing capabilities within the antenna itself. Active antennas require a power source to operate and offer greater flexibility and performance in challenging signal environments.

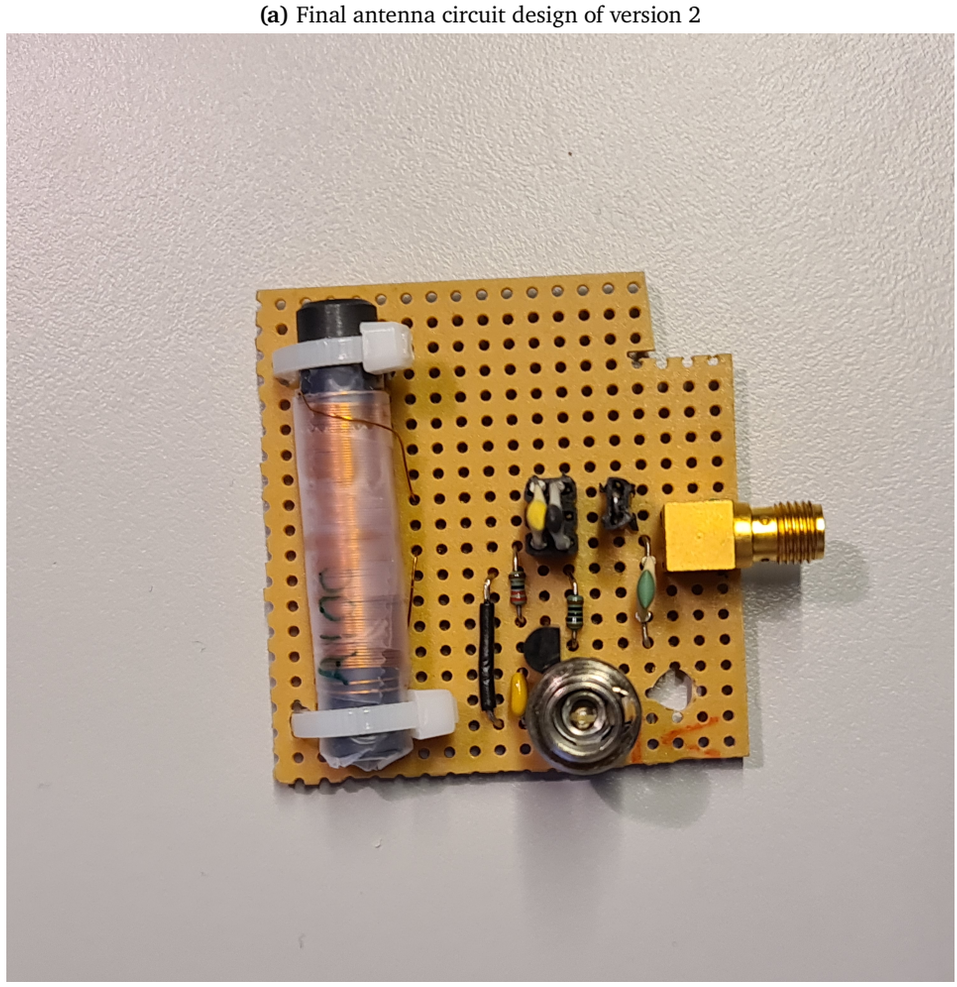
The simplest active antenna is a passive antenna with connected to a low-noise

amplifier. As the goal of this project is to do most signal processing digitally, no other components than an amplifier were added to the circuit. An expert in the field designed the amplifier circuit, which utilizes a three-volt battery pack to boost the signal from the antenna. Figure 6.2 illustrates the active antenna circuit that was ultimately utilized, as well as its implementation in real-life.

I implemented the amplifier circuit for both antennas in the system. However, I decided to power each antenna with separate battery packs to prevent any additional noise caused by connecting both antennas.



(a) Final antenna circuit design of version 2



(b) Circuit from Figure 6.2a, implemented on PCB.

Figure 6.2: The final design and real-world implementation of version two's antenna circuit.

6.2 System design

One of the concerns that required attention in the initial prototype was its lack of portability. It was fragile and challenging to transport, which made conducting precise experiments difficult. In order to address the issues, steps were taken to improve the system's portability and rigidity. Two key enhancements were implemented: developing a sturdy frame for the system and eliminating the requirement of connecting a laptop directly to it.

6.2.1 Raspberry Pi for data collection

Initially, the first version encountered portability issues as the software-defined radios required a laptop connection to commence measurements. Consequently, a wireless control solution was necessary to enable users to operate the receiver remotely. However, the primary objective of this project is to explore the potential of software-defined radios for detecting avalanche beacons. As a result, the development of a user-friendly application for receiver control was not the top priority. It is reasonable to assume that the prototype user possesses expert knowledge regarding its use. Therefore, I ultimately ended up utilizing a small one-card computer of the model Raspberry PI 3B+, which exposes a terminal interface over SSH.

One-card computers, exemplified by devices such as the Raspberry Pi, are compact and versatile computing platforms that have gained immense popularity in recent years. These miniature computers are designed to provide a low-cost yet powerful solution for a wide range of applications. Despite their small form factor, one-card computers possess the essential components of a traditional computer, including a processor, memory, storage, and input/output interfaces. These devices often run on open-source operating systems, such as Linux, and can be easily programmed and customized to suit specific needs. The Raspberry Pi, in particular, has become a prominent player in this domain, offering a credit card-sized board with abundant connectivity options, including USB ports, HDMI, and networking capabilities. Their affordability, accessibility, and vast community support made them an appealing choice for this project. A significant advantage of one-card computers, particularly the Raspberry Pi, is their capability to be powered by portable battery banks.

When communicating wirelessly with the system, it's crucial to avoid introducing additional radio interference near the frequency of avalanche beacons. Fortunately, modern communication protocols like Bluetooth and Wifi operate at a much higher frequency range, which minimizes this risk. Bluetooth operates on the 2.4GHz ISM spectrum band, which utilizes frequencies in the

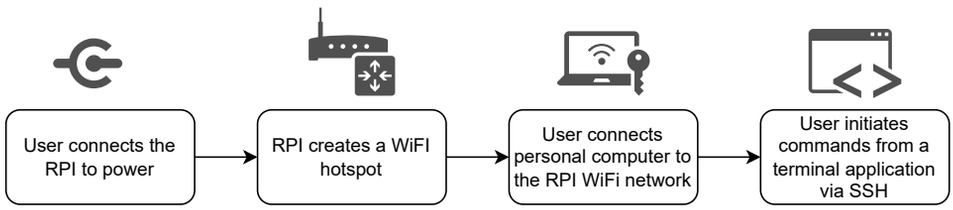


Figure 6.3: Flowchart that illustrates how to control the system wirelessly.

range of 2400 to 2483.5MHz[1]. And WiFi either operates on the same 2.4GHz band or on 5GHz. All of these frequencies are well above the frequency of an avalanche beacon, which operates at 457kHz. WiFi generally has a longer range than Bluetooth. Therefore, I opted to use WiFi for wireless communication with the system. The Raspberry Pi can create a WiFi hotspot, allowing other computers to connect and enabling wireless communication without the need for a pre-existing wireless network.

Since this is just a prototype, it's beneficial to have full control over the system and operating system. This enables me to make quick fixes on the spot. To achieve this I utilize the SSH protocol. SSH, or Secure Shell, is a cryptographic network protocol that provides a secure and encrypted means of connecting to remote computers or servers over an untrusted network. It enables secure remote access, remote command execution, and secure file transfers, making it a widely used protocol for secure remote administration and secure data transfer. In addition to offering a secure means of communication, SSH comes preinstalled on numerous operating systems, enabling swift setup of new systems.

To control the system wirelessly, the final pipeline is illustrated in Figure 6.3. The Raspberry Pi is powered by the user using a wall socket or a portable power bank, which automatically turns it on. After being powered on, the Raspberry Pi will automatically open a WiFi hotspot. The user can connect their personal computer to the Raspberry Pi's WiFi network and then directly control it through SSH. The Airspy HF+ Discovery with the antennas can now be connected to the Raspberry Pi instead of the user's laptop. Once the user has access, they can easily instruct the software-defined radios to collect data, execute analysis scripts, or transfer files to and from the Raspberry Pi.

1. `https://www.bluetooth.com/learn-about-bluetooth/key-attributes/`

6.2.2 3D printed frame

During the frame-building process, I carefully considered three materials: wood, steel, and plastic. Wood boasts benefits such as affordability, lightness, and a strong structure. Steel, on the other hand, offers more stability and water resistance but comes at a higher cost and is significantly heavier. Both wood and steel require construction knowledge and tools to use effectively. However, the emergence of 3D printing has made it possible to quickly create lightweight, durable, and cost-effective prototypes with minimal construction knowledge. As a result, I decided to use plastic and 3D printing to create the frame.

3D printing, also known as additive manufacturing, is a groundbreaking technology that has revolutionized traditional manufacturing processes. It enables the creation of three-dimensional objects by layering materials based on a digital design. The process begins with the generation of a digital model using computer-aided design (CAD) software or through 3D scanning of an existing object. This digital design is then sliced into multiple thin cross-sectional layers, which serve as the foundation for the additive manufacturing process.

Computer-aided design (CAD) is a powerful tool used by designers to create detailed digital models of products and structures. With CAD software, designers can create 2D and 3D models, make modifications, and test their designs for performance and functionality. The introduction of CAD has transformed the design process, resulting in increased precision and productivity while also minimizing the time and resources needed to create a product. Whether designing a complex machine or a simple product, CAD is an essential tool for modern design professionals. I designed the frame for this system using the online CAD application Onshape[2]. A screenshot from this design process can be seen in Figure 6.4.

There are different types of 3D printers that use various techniques, such as extrusion, material deposition, and solidification, to create objects. During the process, the printer deposits or cures materials like polymers, metals, and ceramics, building the object layer by layer. This additive approach provides a high level of design flexibility, making it possible to produce intricate details, complex geometries, and prototypes with minimal material waste. For this project, I utilized a Prusa MK3S[3] 3D printer, which belongs to the Fused Deposition Modeling (FDM) category. This type of printer operates by layering thermoplastic filaments to construct 3D objects.

The final frame had a cross shape and grooves that securely held the receiver's

2. `https://www.onshape.com/en/`
3. https://www.prusa3d.com/product/original-prusa-i3-mk3s-kit-3/

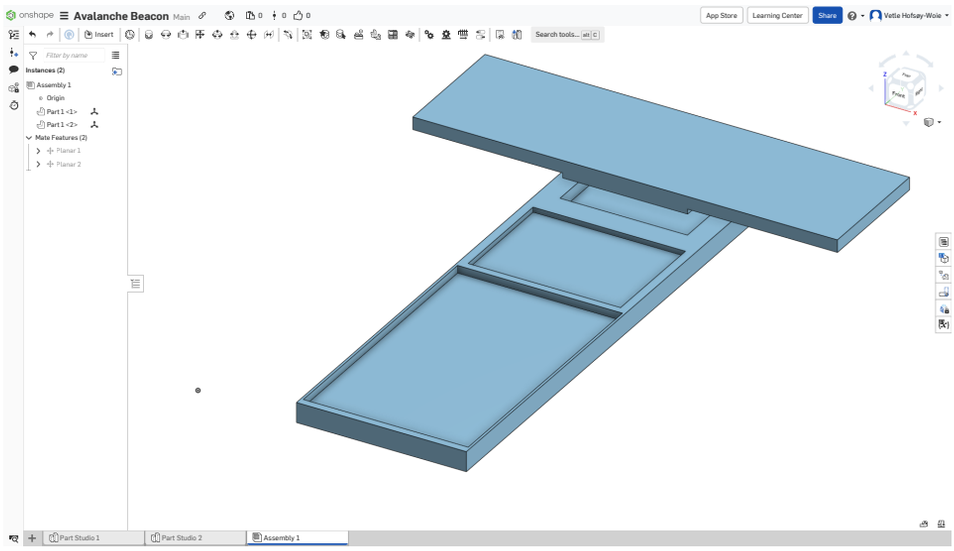


Figure 6.4: I utilized the Onshape CAD application to design the 3D frame model.

different components. The final 3D printed frame with components can be seen in Figure 6.5. Zip ties or rubber bands were used to fasten each component, making it easy to remove and modify. To minimize the impact of antenna coupling, the two antennas were positioned at a minor distance from each other. This is because the electromagnetic fields of one antenna can interfere with the performance of another antenna situated nearby. Additionally, a small gap was maintained between the antennas and the Raspberry Pi to prevent the antennas from being affected by any radio noise emitted by the Raspberry Pi's electrical components.

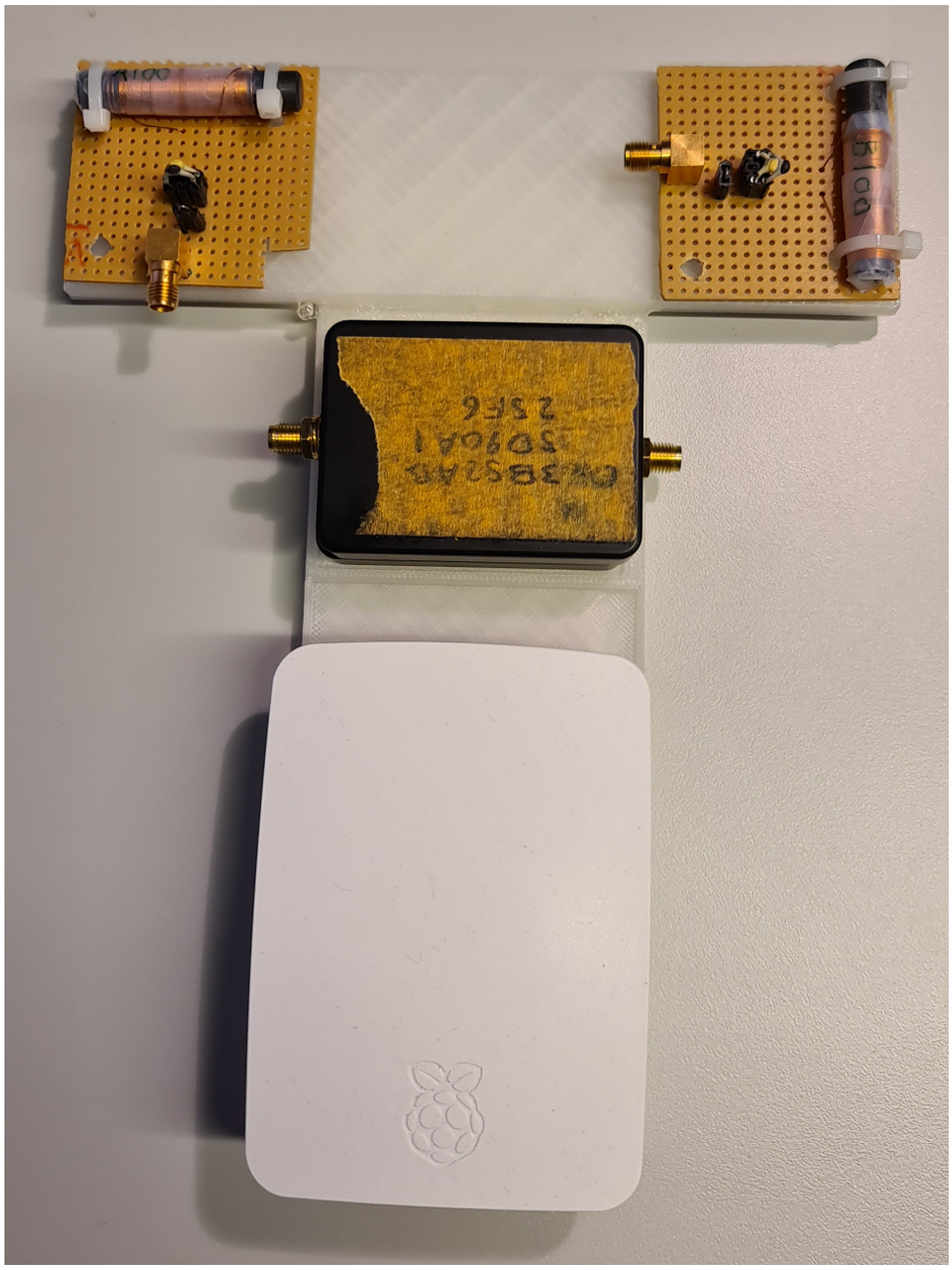


Figure 6.5: The completed frame design featuring all the components, including two antennas, Airspy HF+ Discovery SDR devices, and a Raspberry Pi, all arranged on top of the 3D printed frame.

6.3 Evaluation

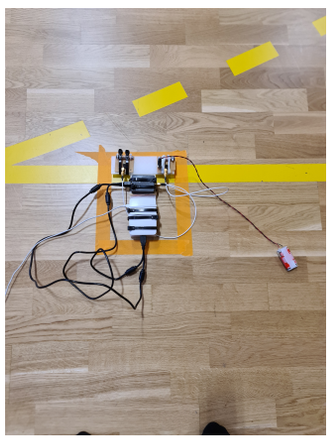
The previous experiments had shortcomings in various areas, including low resolution, inadequate sample points, and high radio interference. In this section, I will discuss the new experiments conducted with the latest version. My objective for conducting these experiments was to assess the coverage area of the previous version's passive antenna in comparison to the active antenna of the current version.

6.3.1 Experiment

Avalanche beacons are primarily used in mountainous regions where there are minimal electronic devices that could cause radio interference. Therefore, to test the prototype, it would be best to conduct experiments outdoors away from any potential sources of interference. However, due to the inclement weather conditions at the time of the project, it was not feasible to perform the tests outside, as the prototype was not waterproof. Instead, I conducted the experiments inside a sports hall on a handball court, as it was readily available and seemed to possess minimal radio interference.

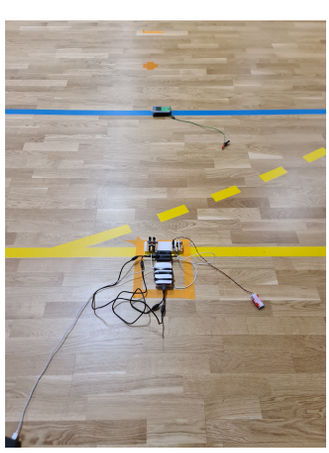
The tests followed the same methodology as the previous range tests, with the exception of an increase in both the number of sample points and the total distance covered. To determine whether the new active antenna performed better than the old passive antenna, one of each was used for this test. According to the previous experiments, the strongest signal is detected when the receiving antenna is parallel to the transmitting antenna. This aligns with the principles of avalanche beacons, as explained in Section 2.1. Therefore, in order to determine the farthest distance at which the transmitter can be detected, the two antennas are positioned parallel to the transmitting beacon.

During the experimentation, the sports hall was being used by other individuals, thus limiting the maximum range available for the experiments to twenty meters. The measurements were taken at one-meter intervals up to the twenty-meter mark. A five-second recording was made for each measurement, resulting in five on/off periods of the transmitting beacon. Pictures of the experiment environment are shown in Figure 6.6.



(a) Close up on the receiver, both antennas are parallel to the transmitter.

(b) The Mammut Element avalanche beacon was one of the beacons used for the experiment.



(c) The receiver and the Mammut element avalanche beacon during one of the experiments.

(d) The Mammut Barryvox S was the other beacon used for the experiments.

Figure 6.6: Pictures from the second experiment.

Earlier, it was mentioned that electronic parts have the potential to create radio interference that can have a detrimental effect on the signal-to-noise ratio of the measured signal. As a result, the Raspberry Pi could potentially impact the system's performance in a negative manner. To determine the extent of this impact, the same range experiment was conducted with the Raspberry Pi situated as far from the antennas as possible using the USB cables. A noticeable difference in the measured signal-to-noise ratio would indicate that the Raspberry Pi has a significant effect on radio interference when it is positioned near the antenna compared to when it is placed further away.

Figure 6.7: To test the effect the Raspberry Pi had on radio interference, tests were done with the Raspberry Pi as far away as the USB cables would allow.

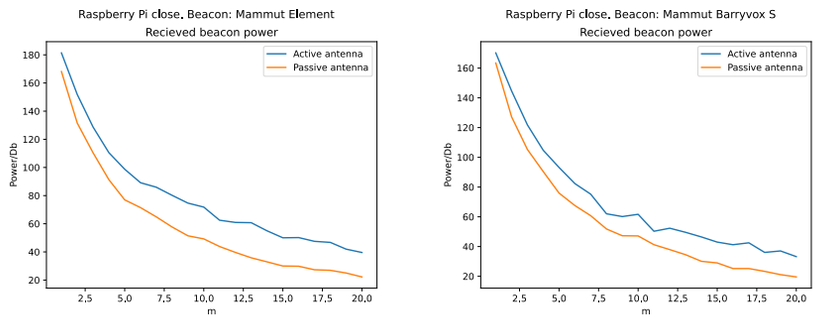
In order to evaluate if the brand of the transmitting beacon affects the received signal, two different beacons were utilized for all tests. The beacons used were the Mammut Barryvox S and the Mammut Element avalanche beacon, which were chosen due to their availability and representing two different eras of avalanche beacons. The Mammut Barryvox S is considered the latest technology, while the Mammut Element is an older but still commonly used model for avalanche beacons.

In the previous experiments, when receiving signals from the Airspy HF+ Discovery, the center frequency of the down conversion was set to the same frequency as the avalanche beacons. However, when using software-defined radios, it is common to observe a frequency spike at the center frequency, even though there is no signal at the time. This frequency spike is a result of inherent DC offset and other factors in the SDR's receiver chain and can introduce distortions and inaccuracies in the received signal. When attempting to receive signals from avalanche beacons or any other desired frequency, the presence of the DC spike can interfere with the accurate detection and analysis of the intended signal. Therefore, to ensure the accuracy of my experiment, I shifted the center frequency to $500kHz$ to prevent any interference with the beacon frequency at $457kHz$.

6.3.2 Results

This section displays the outcomes of the experiments conducted to compare the effectiveness of the active and passive antennas. The analysis in the last section was utilized to ascertain the decline in measured beacon blip power and signal detectability as distance increases.

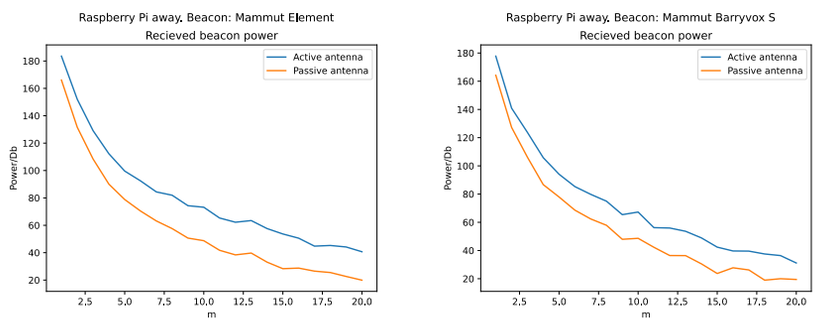
Figures 6.8 and 6.9 illustrate how the measured signal power declines as the distance between the transmitting beacon and the receiver increases. In both of these figures, there are plots of signal degradation for both beacons used in the experiments. Figure 6.8 plots the measured signal power over distance with the Raspberry Pi mounted to the frame discussed in Section 6.2.2, and Figure 6.9 illustrates the result measured with the Raspberry Pi as far away from the antennas as the USB cables would allow.



(a) Received signal power from Mammut Element avalanche beacon

(b) Received signal power from Mammut Barryvox S avalanche beacon

Figure 6.8: Measured signal power over distance with the Raspberry Pi close to the antennas.

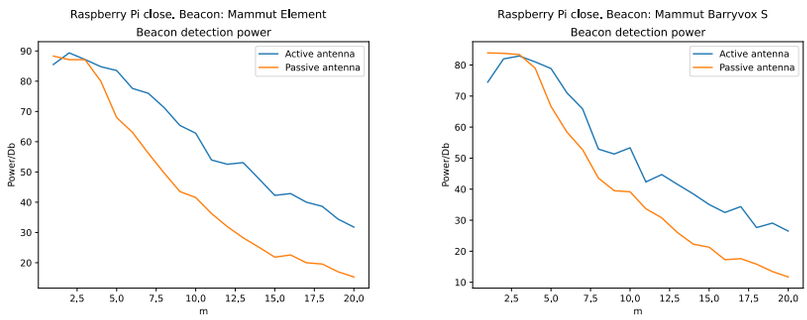


(a) Received signal power from Mammut Element avalanche beacon

(b) Received signal power from Mammut Barryvox S avalanche beacon

Figure 6.9: Measured signal power over distance with the Raspberry Pi away from the antennas.

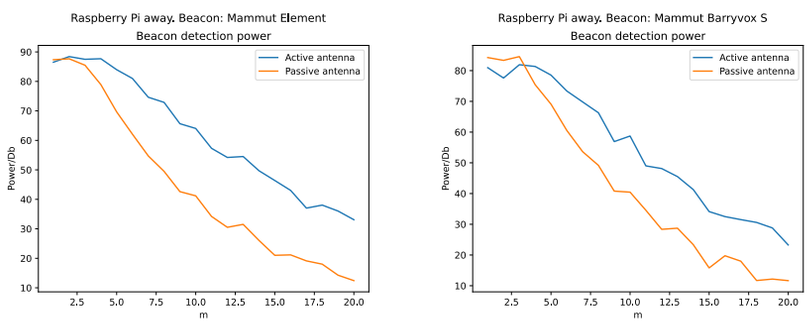
Figures 6.10 and 6.11 demonstrate how the detectability of the beacon signal decreases as the distance between the transmitting beacon and the receiver increases. Both graphs show signal degradation plots for the Mammut Barryvox and Mammut element beacons used in the experiments. Figure 6.10 represents the standard deviation normalized blip power over distance with the Raspberry Pi mounted to the frame discussed in Section 6.2.2, while Figure 6.11 illustrates the detectability measured with the Raspberry Pi positioned as far away from the antennas as the USB cables would allow.



(a) Measured signal-to-noise ratio from Mammut Element avalanche beacon

(b) Measured signal-to-noise ratio from Mammut Barryvox S avalanche beacon

Figure 6.10: Beacon detection power, or signal-to-noise ratio, with the Raspberry Pi close to the antenna.



(a) Measured signal-to-noise ratio from Mammut Element avalanche beacon

(b) Measured signal-to-noise ratio from Mammut Barryvox S avalanche beacon

Figure 6.11: Beacon detection power, or signal-to-noise ratio, with the Raspberry Pi away from the antenna.

6.3.3 Discussion

In the previous sections, I presented experiments were conducted with several objectives in mind. The primary goal was to determine if the active antenna could provide a better range compared to the passive antenna in the previous chapter. Another objective was to assess the extent to which the Raspberry Pi influenced the measured radio noise. Lastly, the experiments aimed to provide insights into the potential signal variations between avalanche beacon models.

The results presented in Figures 6.8 and 6.9 clearly demonstrate that the active antenna outperforms the passive antenna in terms of signal power. When the transmitting beacon is within five meters of the receiver, there is a ten-decibel

difference between the two antennas. As the distance increases, the difference also increases to between twenty and thirty decibels, indicating that the signal measured at the active antenna is one hundred to one thousand times more powerful than the signal measured at the passive antenna. However, it is important to note that the amplifier in the active antenna also amplifies radio noise around the signal, which may explain the less smooth signal detection power depicted in Figures 6.10 and 6.11. Nonetheless, the active antenna still exhibits significantly higher signal detection power compared to the passive antenna.

From visual inspection of the standard deviation normalized plots, as exemplified in Figure 6.1f, at each distance, a good threshold for signal detection is a power level of twenty standard deviation units or more. At a distance of twenty meters, the active antenna displays a signal power that exceeds two hundred standard deviations. However, the passive antenna falls below the threshold of twenty standard deviations at 18 meters.

The experiment had a second objective, which was to investigate the impact of the Raspberry Pi's location relative to the antennas on the detectability of beacon signals. After analyzing the plots in Figure 6.10 and Figure 6.11, I found that there was minimal difference. This suggests that relocating the Raspberry Pi away from the frame had little effect on improving signal detectability.

I conducted an investigation to determine if there were any distinguishable differences in the signals produced by various avalanche beacons. The results revealed that the two beacons that were tested did not display any noticeable discrepancies. However, for a more conclusive outcome, it would have been ideal to use more beacons from different manufacturers. Unfortunately, this was not possible due to the unavailability of additional beacons at the time of the study.

6.4 Final thoughts on version 2

The second version of the software-defined radio beacon surpasses the first version in most aspects, particularly in terms of portability and range, which were the key objectives for the new design. Because the maximum range for the experiments was restricted, the limit for detection with the active antenna was not determined. However, based on the rapid decline of the signal power plots in Figures 6.8 and 6.9, it is evident that it would not be able to compete with contemporary state-of-the-art avalanche beacons in terms of detection range. One potential solution for extending the detection range is to raise the amplification on the active antenna. To achieve this, the input voltage of the

amplifier circuit from Figure 6.2a can be increased. However, due to project time constraints and uncertainty about whether the Airspy HF+ Discoverys can handle such an increase in input voltage, this option was not further investigated.

7

Discussion

7.1 Requirements

This section will discuss the project's success regarding the requirements stated in Table 1. There were four initial requirements;

- The beacon should be able to store data for later analysis
- The beacon should be small enough to fit on a common crewless vehicle
- The beacon should have a search range similar to today's state-of-the-art avalanche beacons.
- The algorithms should be able to detect the presence of an avalanche beacon automatically.

7.1.1 Signal data storage

Although work still needs to be done to evaluate different methods of storing signal data, the system can store data in the form of I/Q data. I/Q data uses 64 bytes for each sample; even with a modest sampling frequency of 1 Mhz, this storage method can quickly fill up the persistent memory of small devices. To store the total signal of a 10-minute rescue operation, one would need approximately 38 GB of storage. However, storing plain I/Q data stores information

about all frequencies, but only one frequency is of interest when discussing avalanche beacons. Therefore, it might be beneficial to implement some real-time compression algorithm, possibly filtering out frequencies outside the range of interest.

7.1.2 Size

The size of the beacon ended up at 20x17cm with a weight of 150 grams, which is small enough to fit many crewless vehicles. The beacon itself could be made smaller through a more thoughtful design; however, the goal of this project was only to provide proof of concept for this sort of avalanche beacon. Suppose an attempt is made to decrease the size. In that case, it is essential to remember the effect of electronic interference and that the antennas should be positioned away from any other electronic sources.

7.1.3 Range

Today's state-of-the-art avalanche beacons commonly have an effective search range of 40-70 meters, and the project aimed to develop an artifact with similar search ranges. However, due to the limitations of the experiments carried out in this thesis, we only know that the search range of the final artifact is greater than 20 meters. Based on the projections of the findings, it is likely that the artifact could achieve a search range of 30 meters. Ultimately, I did not reach the goal of having the same search range as commercial avalanche beacons.

7.1.4 Automatic detection of signal

The last goal of creating an algorithm that could automatically detect the presence of a transmitting avalanche beacon was reached. Indeed, I designed three algorithms, and all three could automatically detect the presence of other beacons. However, both the time domain and the frequency domain algorithm were very limited by the surrounding noise levels, and both struggled with low signal-to-noise ratios. The third hybrid algorithm was a definite improvement on the first two. It was more robust to low signal-to-noise ratios, and one could choose all necessary parameters based on the specifications of avalanche beacons or statistics. However, although the experiments did not show it, the hybrid algorithm can be fooled by significant outliers in the noise. One could possibly improve upon this weakness by considering the periodic nature of the avalanche beacon signal and integrating the signal over time. If the noise is evenly distributed around zero, then an integration over time should zero out the noise, while the signal will grow indefinitely. However, this thesis did not

examine this approach.

7.2 Detecting multiple transmitting beacons

The solutions presented in this thesis gloss over an essential feature of avalanche beacons. The ability to detect and differentiate multiple transmitting avalanche beacons. An avalanche beacon that cannot differentiate between multiple transmitting beacons will be useless in the event of multiple burials, as the detected signal can come from several directions. The algorithms explained in this thesis make a big assumption: they expect only one transmitting beacon. One would have to develop them further to be helpful in a real-world scenario. Detecting and differentiating between multiple burials could potentially be achieved by examining the duration between pings. If the algorithm detects more than one ping each second, it can be sure that more than one beacon is transmitting. The beacon can then estimate directions and distances for each beacon each second.

7.3 Distance and direction estimation

A significant limitation of this thesis is distance and direction estimation once a beacon signal is detected. Even though I present the theory on how one could do these estimations in Chapter 2, I neither implement nor test these techniques, as with the problem of detecting multiple burials, a beacon that cannot give direction and distance is useless in a real-world situation.

The distance and direction equations described in Chapter 2 only depend on the relative power difference of each antenna in the beacon. However, in practice, many other parameters affect the accuracy of the estimations. There may be many such parameters, but the two that probably have the most considerable effect are sample resolution and sample rate.

The possible error introduced to the estimations by sample rate should be null if the sample rate is kept above the Nyquist frequency of two times the frequency of interest. In the case of avalanche beacons, this is right below 1Mhz.

The sample resolution will affect the estimations by introducing uncertainty in each sample. This uncertainty grows larger with lesser sampling resolutions and will increase the error of the estimations. This thesis has only examined one software-defined radio with one sampling resolution. Although it was shown that the radio with the given sampling resolution could detect transmitting

avalanche beacons, it is unknown whether it would provide sufficient distance and direction estimations.

7.4 Experiment limitations

The main experiment of this thesis took place inside an indoor sports arena on a handball court. In this arena, I was limited to a maximum range of 20 meters, far from the range of commercial avalanche beacons. These experiments should have been performed outside in an area with little radio interference and open space with the possibility of testing a range of 100 meters. However, due to the arctic climate, the need for a computer during the experiments, and a prototype that needed to be water resistant, this was not deemed possible.

Another limitation of the experiments was that I utilized only one type of commercial avalanche beacon and one type of software-defined radio.

Although using different avalanche beacons should not significantly affect the results due to the rigorous demands of the European standard, it still would have been preferable to do the experiments across several models and brands.

The choice of software-defined radio could significantly impact the experiment's results. As mentioned earlier, different sampling rates and resolutions could impact the ability of the algorithm to detect beacons, but also hardware considerations such as the quality of the analog to digital converter, the quality of the linear amplifier used, or the device's sensitivity. Although not directly visible in the received data, these hardware considerations can severely impact the quality of the data. However, the experiments presented in this thesis do not examine the magnitude of these effects.

7.5 Lessons learned

During this project, I have worked within several domains, such as engineering, software development, antenna design, and physics, and from this, I have learned several lessons. Most prevalent of these is that commercial software-defined radios can deliver a signal quality in which one can detect a transmitting avalanche beacon using rudimentary digital signal processing techniques. Furthermore, when carrying out the experiments, I observed that electronic interference has little effect on a transmitting beacon but can drastically affect a receiving beacon if the source of the interference is very close to the beacon.

8 Conclusion

In my thesis, I have developed a system capable of utilizing software-defined radios to receive and store radio signals within a frequency range similar to the one used by avalanche beacons. Additionally, I have created three algorithms that can analyze the received signals and automatically identify if there was a transmitting avalanche beacon nearby during the signal reading.

Due to limited space, the experiments were constrained to a range of 20 meters; however, the system and the algorithms were able to detect a transmitting signal from this distance, and the results showed that the system should be able to detect the beacon from even further away.

Based on the findings of this thesis, commercial software-defined radios are good enough to be used for avalanche beacon detection. It is also clear that digital signal processing techniques can be beneficial in detecting avalanche beacons.

8.1 Future work

During the conclusion of the work of this thesis, I have identified areas that need further development and research before one can use the system and algorithms presented in real-world scenarios. I will now present some of these areas in the following paragraphs.

8.1.1 Direction finding and distance estimation

Firstly, estimating the direction and distance to the buried victim is essential. Although I present the theory on how to do this from each antenna's relative power measurements, I have yet to implement and test this method.

As an afterthought, If the intended use is for a drone to carry the system, one might not need to estimate the direction or the distance. Indeed, one could utilize the drone's ability to move quickly across an area and instead use methods like gradient ascent. By measuring the power and then moving in a direction, the drone could follow the path of the most significant increase in signal strength. In this case, the only estimation needed is the actual power level of the signal, which one can estimate using the Pythagorean theorem on the received signal from both antennas.

8.1.2 Differentiating between several transmitting avalanche beacons

In a rescue scenario involving multiple victims, it will be crucial for the system to differentiate between various transmitters and indicate the locations of each victim's burial. For this task, one could utilize the periodicity of the transmitted signal. More than one detected signal in one second means more than one nearby beacon is transmitting. After detecting multiple beacons, each beacon can be differentiated by the time since the last signal was received.

Suppose a gradient ascent technique is used for localizing. In that case, it is essential to note that the method will most likely find only one buried victim as there will be a local maximum of the signal power above that victim.

8.1.3 Integrating signal over time

If one assumes that the radio interference behaves like Gaussian noise, one can translate the noise so that its expected value is 0, as I do in the hybrid algorithm of version 2. Then, integrating the signal over time should lead to an unbounded growth of the signal power while the noise should average out to 0.

If this hypothesis is true, it would be interesting to examine whether one could increase the signal-to-noise ratio during a search by utilizing this technique. However, it will be hard to differentiate between multiple transmitting beacons because we're integrating the signal over time. Therefore, this might be most beneficial when the signal-to-noise ratio is meager and the main objective is to

find a transmitted signal. In such situations, one could again utilize gradient ascent and switch to regular search when the signal is strong enough.

8.1.4 Better antenna

I constructed the antennas used in this thesis using regular copper wire despite having limited knowledge of antenna design. In contrast, commercial avalanche beacons use litz wire and are created by professionals in the field. Through my work on this thesis, I have come to appreciate the importance of proper antenna design and its impact on the signal-to-noise ratio of the received signal. Therefore, there is a need to develop an antenna that is better suited for this task.

A

Appendix: Results from experiment 1

In this appendix, you will find plots for each distance recorded in the first experiment. The signal from the avalanche beacon is easily identifiable in most of the plots, appearing as yellow spots in the spectrogram or spikes on the estimated signal-to-noise ratio plots.

A.1 Distance plots:

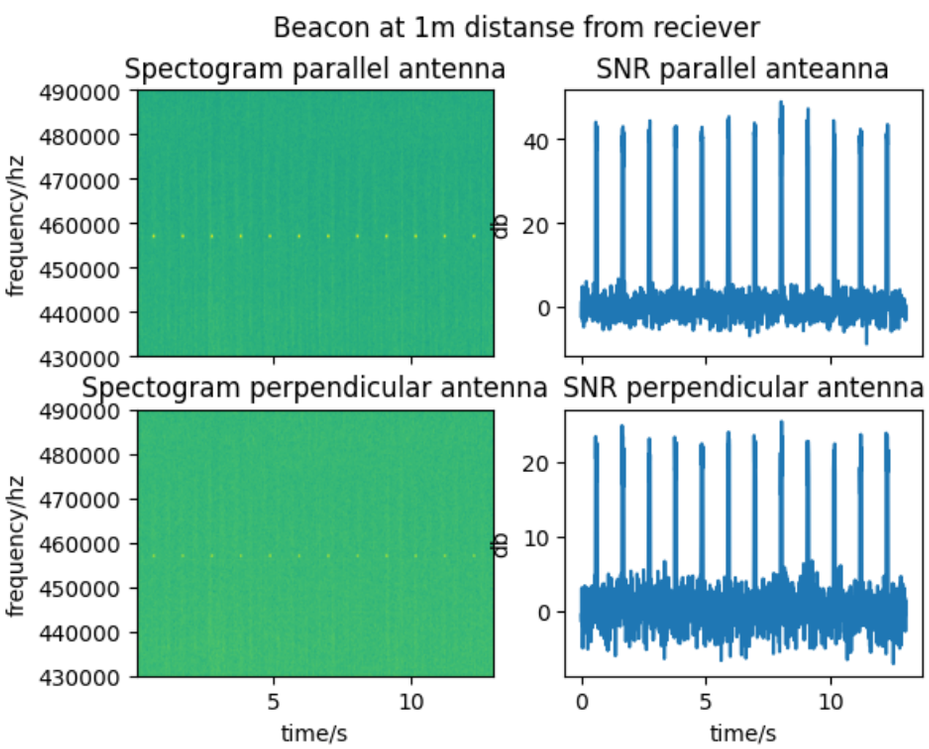


Figure A.1: Spectogram and estimated signal-to-noise ratio with beacon one meter away from the receiver. The signal can be seen from the yellow dots in the spectrogram or the spikes in SNR.

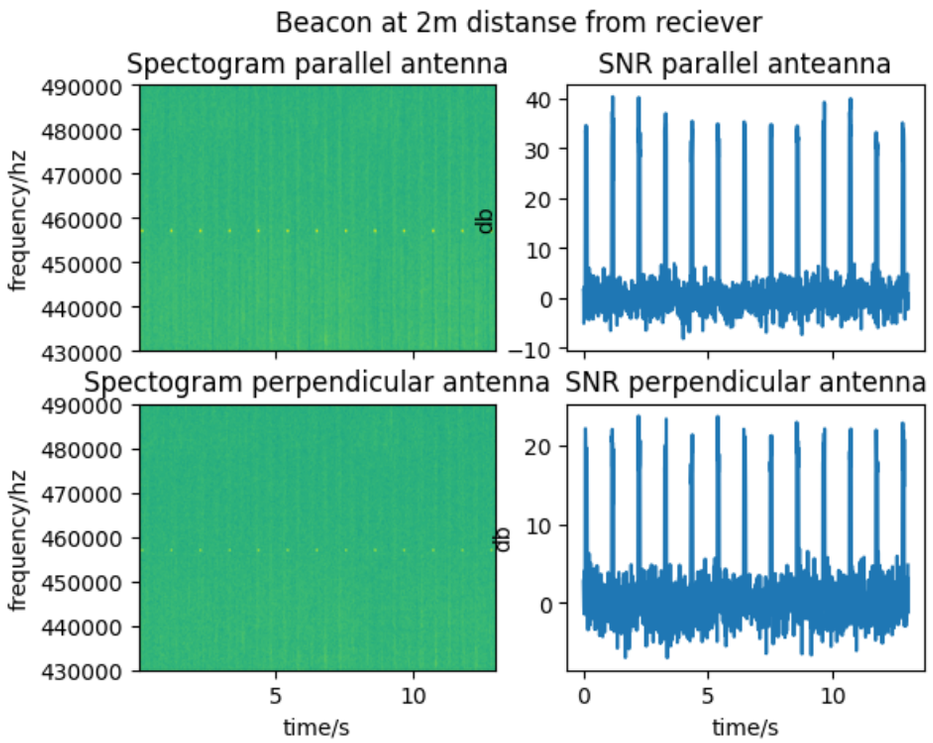


Figure A.2: Spectrogram and estimated signal-to-noise ratio with beacon two meters from the receiver. The signal is still quite strong and identifiable.

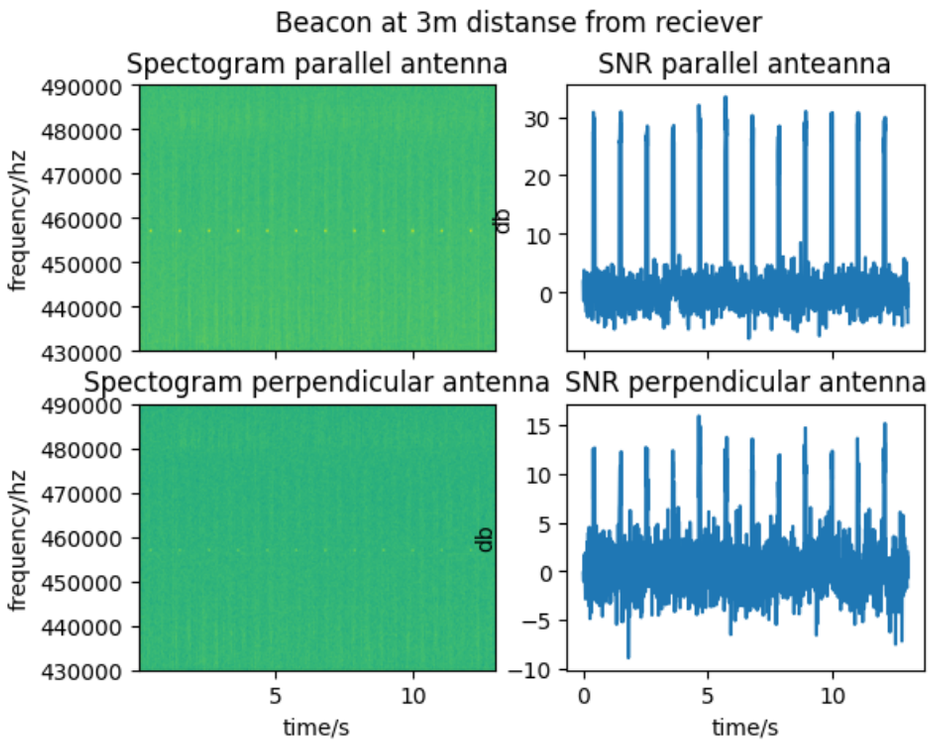


Figure A.3: Spectrogram and estimated signal-to-noise ratio with beacon three meters from the receiver. A decrease in signal strength is now clearly visible.

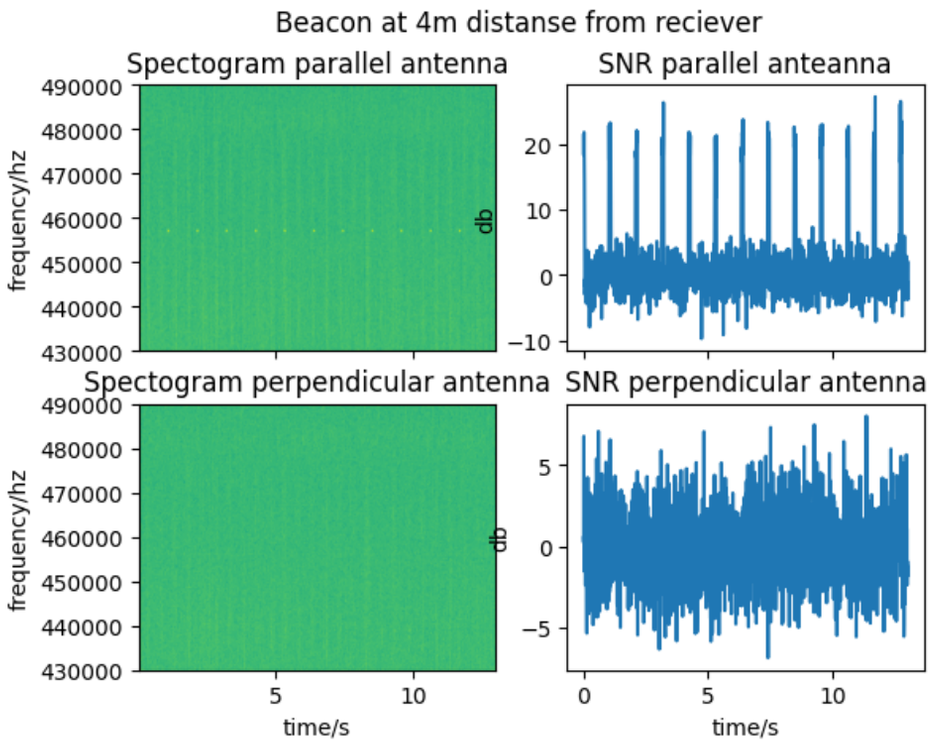


Figure A.4: These plots display the spectrogram and estimated signal-to-noise ratio with the beacon placed four meters away from the receiver. However, the signal is almost undetectable on the perpendicular antenna, while it is still clearly distinguishable on the parallel antenna.

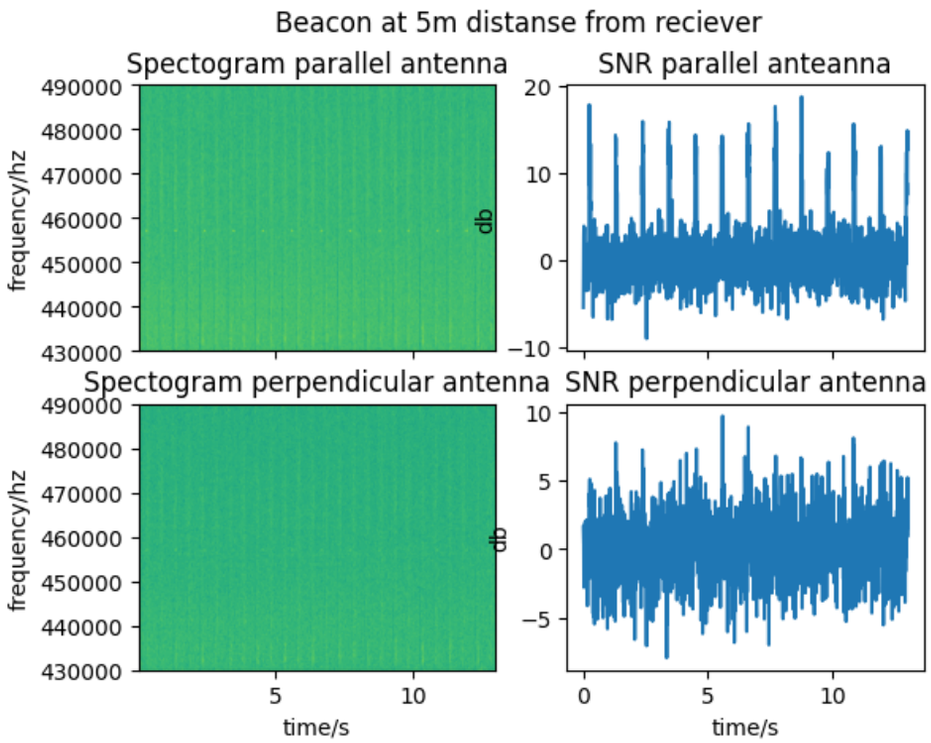


Figure A.5: In these plots, both the spectrogram and estimated signal-to-noise ratio are displayed with the beacon located five meters away from the receiver. On the perpendicular antenna, the signal is almost completely lost, but on the parallel antenna, it is still easily distinguishable.

A.2 Rotation plots:

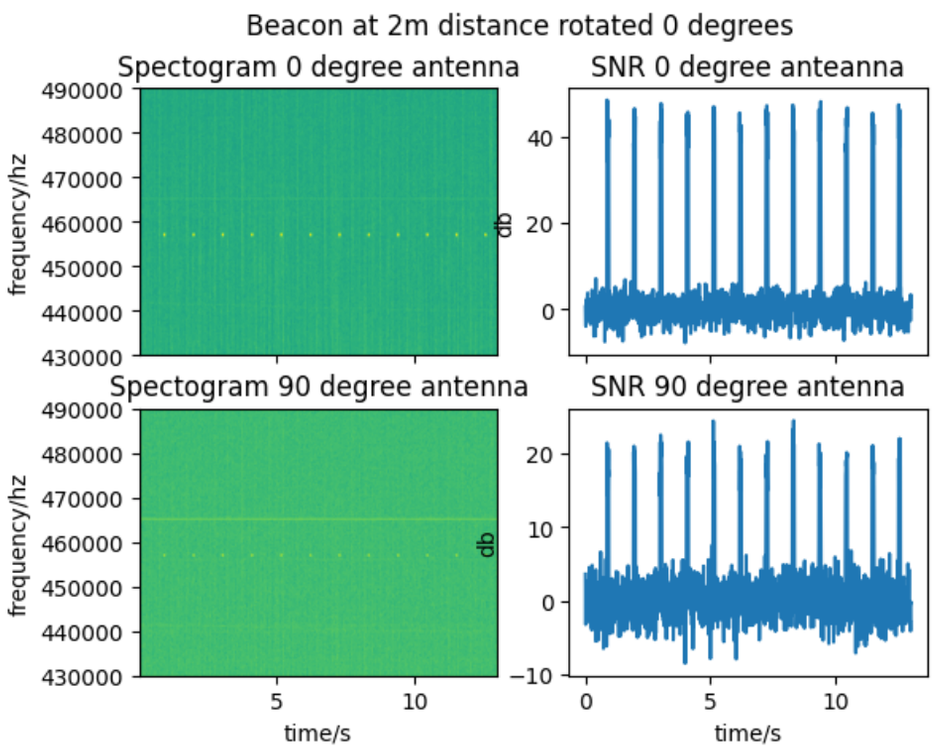


Figure A.6: Spectogram and estimated signal-to-noise ratio with beacon two meters from the receiver and rotated 0 degrees from the first antenna. There is a clear difference between the two antennas in the received signal strength.

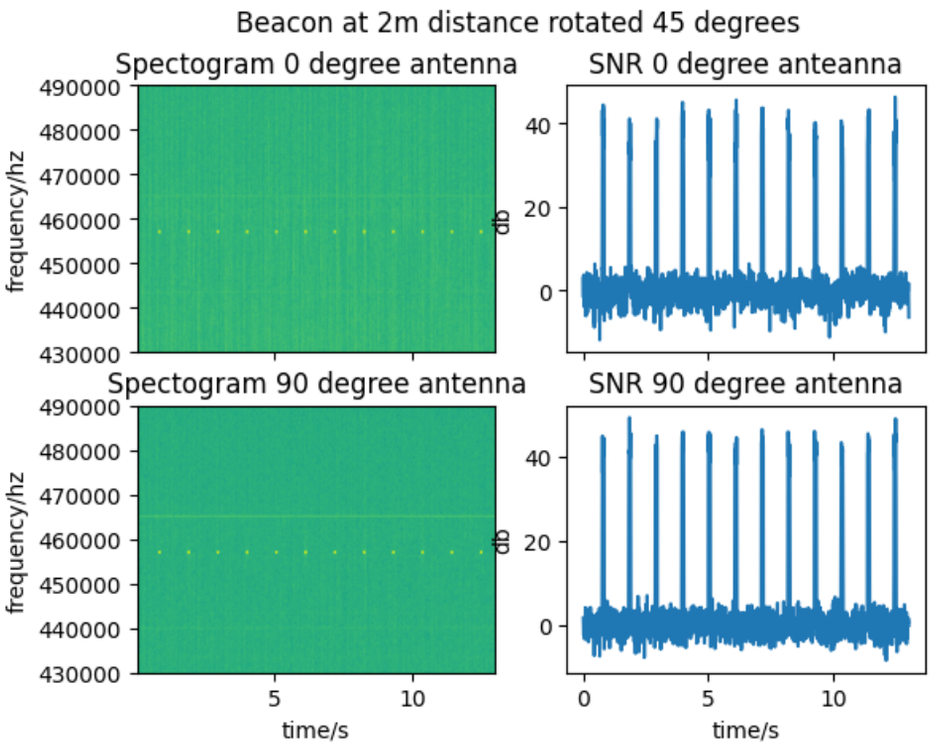


Figure A.7: Spectogram and estimated signal-to-noise ratio with beacon two meters from the receiver and rotated 45 degrees from the first antenna. Both antennas now receive roughly the same signal strength.

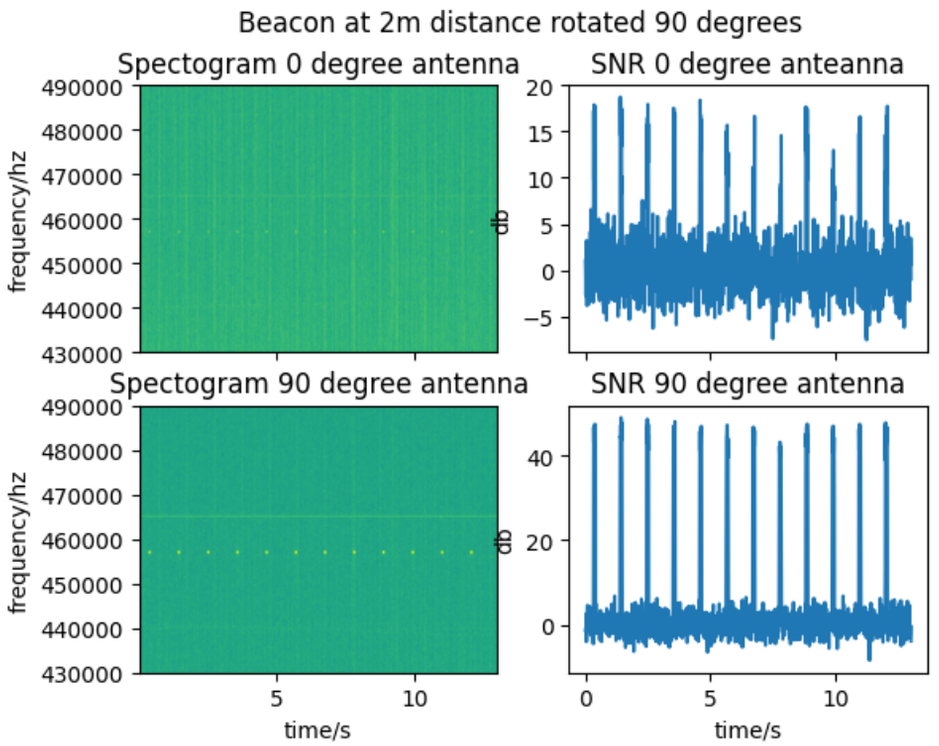


Figure A.8: Spectogram and estimated signal-to-noise ratio with beacon two meters from the receiver and rotated 90 degrees from the first antenna, making it parallel to the second antenna. Antenna number two now receives a significantly higher signal strength than antenna one.

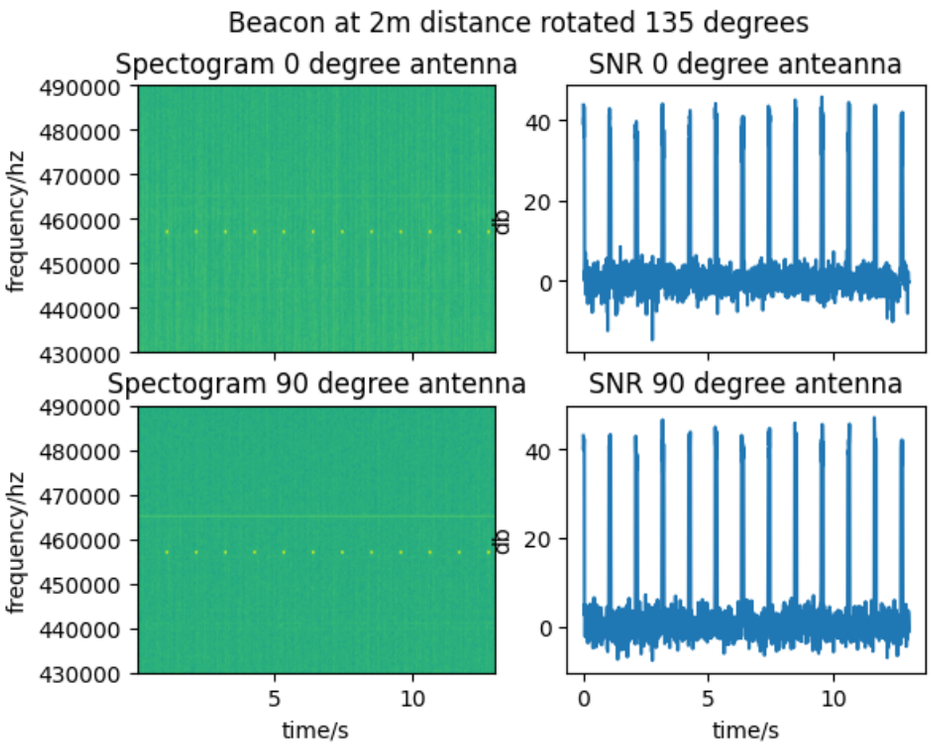


Figure A.9: Spectogram and estimated signal-to-noise ratio with beacon two meters from the receiver and rotated 135 degrees from the first antenna. Both antennas now receive roughly the same signal strength.

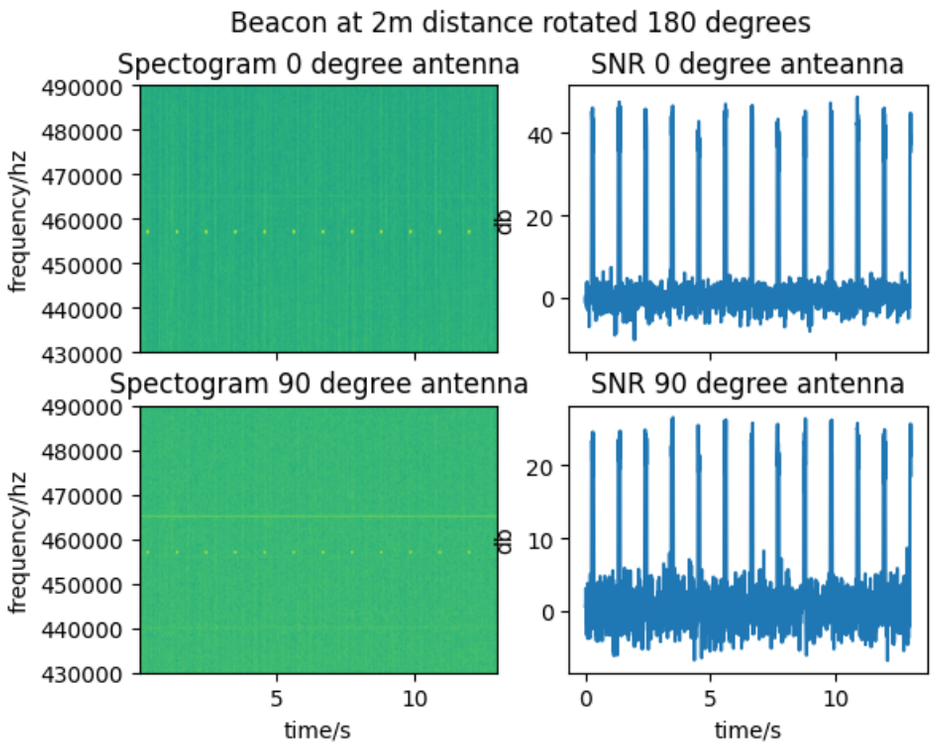


Figure A.10: Spectogram and estimated signal-to-noise ratio with beacon two meters from the receiver and rotated 180 degrees from the first antenna, making it again parallel to the first antenna. The first antenna picks up a much stronger signal.

/B

Avalanche dynamics

There are two main types of avalanches: loose snow avalanches and slab avalanches. Each of these categories can further be divided into dry and wet avalanches. Other categories of avalanches exist, but they have less impact on humans and will not be discussed here.

Unlike all other earthly surficial materials, snow is usually found in temperatures very close to its melting point. This means that there are processes that occur in snow that do not apply to any other material. This fact means snow can deform and compress rapidly under its weight. Another property of existing so close to its melting point is that snow and water can exist simultaneously under many circumstances. Adding water to a snowpack has a trifold effect. Firstly adding water to the snowpack increases the weight of the snowpack, which in turn quickens the deformation. Second, water melts the snowpack, weakening the bounds between the ice crystals. Third, water can lubricate potential gliding surfaces within the snow. Snow is also an excellent insulator, which means it can contain heat from the ground. This fact makes it so that the temperature at the bottom of the snowpack is always close to zero, regardless of the temperature above it. Because of this, the snowpack is constantly melting and freezing, which in turn, changes the properties of the snowpack.

An avalanche occurs when the gravitational pull on the snowpack exceeds the binding forces within the snowpack and the external friction force. From empirical and statistical methods, it has been found that this occurs when the steepness of the slope is greater than 30 degrees[21].

(a) Image of naturally released loose snow avalanche ©Børge Bårdsen

(b) Image of skier released loose snow avalanche. ©Brage Bårdsen

Figure B.1: B.1a and B.1b show two different wet snow avalanches, B.1a shows a naturally triggered avalanche with its characteristic teardrop shape. B.1b Shows a skier-triggered avalanche, where the snow slid beneath the skier. Both of these avalanches were caused by the sun melting the snowpack.

B.1 Loose snow avalanches

Loose snow avalanches occur when the binding force between each ice particle within the snowpack is low. This is the case if much new snow has recently fallen or the snowpack is weakened by sun or rain. If newly fallen snow breaks into an avalanche, it is usually a dry loose snow avalanche. On the other hand, if it is melting or rain that creates the avalanche, it is called a wet loose snow avalanche. Loose snow avalanches often start as minor points and fan outward on their way down the slope, leading to a tear-shaped avalanche. Loose snow avalanches rarely result in fatalities. The reason is that it is usually small in size and is often triggered beneath a skier. Many backcountry enthusiasts call loose snow avalanches for *harmless* sluff. However, given the right circumstances, loose snow avalanches can also prove deadly. This is especially the case if an avalanche happens near cliffs or trees. A loose snow avalanche can also move heavy loads of snow onto weaker snowpacks; the result is a larger, more deadly slab avalanche.

Figure B.1 shows two examples of wet loose snow avalanches. Both avalanches were caused by the heating of the snowpack by the sun. In Figure B.1a, the sun loosened the bindings enough so that the avalanche started by itself, resulting in the classical teardrop shape. Figure B.1b shows a skier-triggered loose snow avalanche. The avalanche started beneath the skier, and the skier was unharmed.

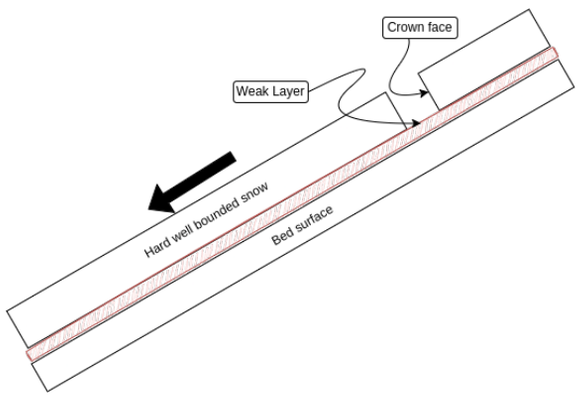


Figure B.2: Diagram showing the building blocks needed for a slab avalanche to occur

B.2 Slab avalanches

Slab avalanches occur when the layers in a snowpack follow specific configurations. A slab avalanche needs three specific layers. First, a relatively harder or more cohesive layer that can slide and act as an avalanche. Second, a relatively weaker or less cohesive layer which can fracture and initiate the sliding of the top layer. Lastly, the bed surface is another harder layer beneath the weak layer where the avalanche can slide. Figure B.2 shows this configuration of layers in a simplified diagram. Sometimes, the bed surface does not need to be snow or ice but can be the ground itself.

When a slab avalanche occurs, it leaves an evident fracture on the top of the slope. This fracture is always orthogonal to the slope where the avalanche occurred. The avalanche will then slide on the bed surface until it reaches the bottom of the slope, depositing all of the snow it has carried; this part of the avalanche is called the run-out. Figure B.3 shows the fracture line and the run-out of a naturally triggered slab avalanche. When this avalanche occurred, there was much wind. The wind packed much snow on top of a weak layer of loose new snow that had fallen earlier that day. When the weight of the wind-packed snow exceeded the strength of the weak layer, the weak layer collapsed, and the wind-packed layer slid down the slope. In this case, the bed surface was a hard ice crust created the week before.

Slab avalanches account for 95% of avalanche fatalities in North America[21], and similar numbers are seen elsewhere.

(a) Image of the crown and starting zone ©Vetle Hofsøy-Woie

(b) Image of the runout of the same avalanche. ©Vetle Hofsøy-Woie

Figure B.3: B.3a and B.3b show the same large slab-avalanche from different directions. This avalanche was caused by wind depositing large amounts of snow on top of a weak layer of soft new snow.

Bibliography

[1] *About RTL-SDR*. en-US. Apr. 2013. URL: https://www.rtl-sdr.com/about-rtl-sdr/ (visited on 05/09/2023).

[2] N. Ayuso et al. "A deep insight into avalanche transceivers for optimizing rescue." In: *Cold Regions Science and Technology* 111 (Dec. 2014). DOI: 10.1016/j.coldregions.2014.12.005.

[3] Mesay Belete Bejiga, Abdallah Zeggada, and Farid Melgani. "Convolutional neural networks for near real-time object detection from UAV imagery in avalanche search and rescue operations." In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. ISSN: 2153-7003. July 2016, pp. 693–696. DOI: 10.1109/IGARSS.2016.7729174. URL: https://ieeexplore.ieee.org/abstract/document/7729174?casa_token=DeWoVOyTaOYAAAAA:uPDMfwARjyrVn4fw3WPC7qwqq5EAlFaaODoCtoIALIudz_oVzGuxmimNfSmxOlwnZJA4kvb8 (visited on 01/15/2024).

[4] Francesco De Giudici et al. "Design and testing of an autonomous ARTVA detector for small drones." In: *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. 2021, pp. 104–108. DOI: 10.1109/MetroAeroSpace51421.2021.9511664.

[5] Peter J. Denning et al. *Report of the ACM Task Force on The Core of Computer Science*. Technical Report. Num Pages: 62 ISBN-9: 087912934. New York, NY, USA: Association for Computing Machinery, July 1988.

[6] EN ETSI. "300 718-1." In: *Avalanche Beacons operating at 457 kHz; Transmitter-receiver systems; Part 1: Harmonised Standard for access to radio spectrum* 865 (2017).

[7] EN ETSI. "302 208-1." In: *Electromagnetic compatibility and Radio spectrum Matters (ERM): Radio frequency identification equipment operating in the* 865 (2006).

[8] *Fatalities – EAWS*. en-US. URL: https://www.avalanches.org/fatalities/ (visited on 01/26/2023).

[9] Richard Hedlund. *Design of a UAV-based radio receiver for avalanche beacon detection using software defined radio and signal processing*. eng. 2019. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-377025 (visited on 01/26/2023).

[10] Pietro Iob et al. "Avalanche Rescue with Autonomous Drones." In: *2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*.

ISSN: 2575-7490. June 2020, pp. 319–324. DOI: 10.1109/MetroAeroSpace48742.2020.9160116.

[11] Michal Janovec, Branislav Kandera, and Kristína Šajbanová. "Using unmanned aerial vehicles during the search of people buried in an avalanche." en. In: *Transportation Research Procedia*. 11th International Conference on Air Transport – INAIR 2022, Returning to the Skies 65 (Jan. 2022), pp. 350–360. ISSN: 2352-1465. DOI: 10.1016/j.trpro.2022.11.039. URL: https://www.sciencedirect.com/science/article/pii/S2352146522007062 (visited on 01/26/2023).

[12] R.C. Johnson and H. Jasik. *Antenna Engineering Handbook*. Electronics Electrical Engineering. McGraw-Hill, 1993. ISBN: 978-0-07-032381-0. URL: https://books.google.no/books?id=xTSNJhVlHGgC.

[13] James H. McClellan, Ronald W. Schafer, and Mark A. Yoder. *DSP First*. en. Google-Books-ID: XLRKPgAACAAJ. Pearson, 2016. ISBN: 978-0-13-601925-1.

[14] Justin J Modroo and Gary R Olhoeft. "Avalanche Rescue using Ground Penetrating Radar." en. In: (2004).

[15] Rafael Olmedo et al. "SICRA: A GNSS cooperative system for avalanche rescue." In: *2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing* (Dec. 2012). Conference Name: 2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing ISBN: 9781467320115 9781467320108 9781467320092 Place: Noordwijk, Netherlands Publisher: IEEE, pp. 1–7. DOI: 10.1109/NAVITEC.2012.6423107. URL: http://ieeexplore.ieee.org/document/6423107/ (visited on 05/03/2023).

[16] M. Pasquier et al. "On-site treatment of avalanche victims: Scoping review and 2023 recommendations of the international commission for mountain emergency medicine (ICAR MedCom)." en. In: *Resuscitation* 184 (Mar. 2023), p. 109708. ISSN: 0300-9572. DOI: 10.1016/j.resuscitation.2023.109708. URL: https://www.sciencedirect.com/science/article/pii/S0300957223000217 (visited on 04/21/2023).

[17] D.M. Pozar. *Microwave Engineering, 4th Edition*. en. 4th. Wiley, 2011. ISBN: 978-1-118-21363-6. URL: https://books.google.no/books?id=JegbAAAAQBAJ.

[18] *Snøskredulykker | Varsom.no*. URL: https://varsom.no/snoskred/snoskredulykker/ (visited on 01/26/2023).

[19] Per Sverre Opedal. *UD 6-81-9 Veiledning i Vintertjeneste – Redning ved snøskredulykker*. Norwegian. Dec. 2010. URL: https://s3-eu-west-1.amazonaws.com/turistforeningen/files/5f0e80e9ba16ae7917635dfd369317915022a0ac.pdf (visited on 05/03/2023).

[20] Federico Toson et al. *AVERLA: AUTONOMOUS DRONE FOR AVALANCHE RESCUE*. Sept. 2021.

[21] Bruce Tremper. *Staying Alive in Avalanche Terrain*. English. 3rd. MOUNTAINEERS BOOKS, Sept. 2018. ISBN: 978-1-68051-138-3.

[22] Colin Zacharias. *AIARE: Avalanche Rescue Student Handbook*. English. 2019. URL: https://www.nemountaineering.com/wp-content/uploads/2019/11/Avalanche-Rescue-Student-Handbook-2019-20.pdf (visited on 04/18/2023).