Software to locate avalanche transceivers using drone

Kevin Arnesen

Supervised by Atle Birger Geitung and Harald Soleim

Master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics,
Bergen University College

Department of Informatics,
University of Bergen

June 2020

Abstract

Avalanches are dangerous events that can be lethal to victims caught by its devastating impacts. The survival chance of the victims decreases drastically with time, where after 10 minutes the chance for survival is slim. The rescue operation is therefore very time limited and the majority of the time is spent to locate the victims based on normal search methods. Using a drone-based system to assist the search could reduce the time required for the search process greatly. Thus, increases the survival chances for the victims. The thesis contains research on a solution that uses information gathered by a drone system fitted with both GPS device and avalanche transceiver to create a map over the area showing the potential locations of the victims. A solution was developed and tested in various scenarios, by simulating drone flights over an area with an avalanche transceiver. This will allow the rescue team to focus on a smaller area and save life critical time in locating individuals and dig them out. The results showed that the solution was capable of reducing the search area down to a few square meters a short period of time instead of requiring the entire avalanche area to be searched. The accuracy of the solution is restricted by the GPS accuracy and noise, which prevents the solution from narrowing down the search area further. The results show that the flight speed of the drone should be 5m/s or lower, otherwise it causes inaccuracy in the mapping results.

Contents

Glossary

fake signal A result indicating lower value than it in fact is. 25

false positive A result which indicates stronger value than it in fact is. 42

omnidirectional receiving signals from or transmitting in all directions. 26, 27, 33, 34

ripple effect the continuing and spreading results of an event or action. 24, 25

SDK Software development kit. 14

UAV An unmanned aerial vehicle is an aircraft without a human pilot on board. 5

1 Introduction

Avalanche are dangerous events that can be destructive and lethal, where time is a critical element when it comes to avalanche victims and their survival rate. Rescue operations per today rely on people and dogs when available to physically moving over greater and potentially dangerous distances in order to attempt to locate victims using transceivers. Having a drone system to assist the search could potentially reduce the search duration greatly.

Figure 1: Avalanche[16]

The project is being developed in cooperation with CodeLab[4] which is a Norwegian technology development company and is being supported by the Research Council of Norway. An UAV, unmanned aerial vehicle, also known as a drone, is being piloted from a ground-based controller. Mounting a drone with sensors will give the ability to collect data in flight and transmitting the data to the ground where it will be processed and analyzed by a software. The software will process the data it receives from the sensors, detect and remove noise and by analyzing it. Thereafter create a visual representation of the data that displays the locations of the avalanche devices carried by potential victims. Utilizing the visual representation, it should be possible to quickly locate the victims and plan a safe route for the rescue operation.

1.1 Motivation

Deaths from avalanches occurs every year, and in Norway there has been reported 66 deaths in the last 10 years, which is an average of 6.6 people a year[10]. The cause of death is normally suffocation, which means they were unable to dig themselves out and they were not located in time by rescuers. Out of the 66 avalanche related deaths in Norway in the past 10 years, 92% out of them were skiers, and it's common for skiers to use avalanche transceivers. Therefore, a majority of these deaths could potentially have been prevented by using such a drone system, because it could reduce the search duration greatly.

Locating victims of an avalanche is a difficult and time consuming task which also puts the rescue team at risk. Using drones to locate victims of avalanches will give rescue teams a great amount of information which will enhance rescue operations. Using the visual representation, they have the opportunity of planning a safe path to the victim and save time which is critical for the victim. This will increase the safety of the members of the rescue team and let them reach people trapped by the avalanche that normally could have been deemed too risky to attempt a rescue. When an area has been declared too risky to move in, they send in a helicopter to survey the area while the rescue team has to wait. In situations like these, drones would replace the need for the helicopter and the team could assess the situation much faster in real time. Eliminating the waiting period for the helicopter will improve the chances of the avalanche victim's survival.

1.2 Goal and research questions

The primary goal of the thesis is to utilize the data collected from the sensor system mounted on the drone in order to calculate and map out the most probable locations of avalanche transceivers in the surrounding area. Thereafter generate a readable visual representation for the users. During the process the research questions below will be answered.

- How can avalanche transceivers be located in the surrounding environment and establish a visual representation?
  - How precise are the generated results?
  - What effect does noise have on the results?

The first question will be answered by a proof of concept with a software that has the ability to locate avalanche transceivers that transmit on the standard frequency. Thereafter create a visual representation of the data such that it's readable by the rescue team.

The second question will be answered by both calculating how imprecise readings can be and testing the solution itself, these results will then be evaluated based on practical usage of the software. Meaning that minor inaccurate data will be tolerated because the goal is not to pinpoint the location with 100% accuracy. The goal is to narrow down the search area greatly, giving the rescue team the opportunity to go straight towards the estimated location rather than searching the entire avalanche area.

The third question will be answered by attempting to efficiently remove noise corrupted sensor inputs, such that noise does not affect the data output and the visual accuracy. The optimal solution would be that the noise has minimal effect on the data output generated by the software.

1.3 Structure

Thesis structure overview;

Chapter 1: Brief introduction, goals, motivation behind the work, related work and a quick look at the thesis structure.

Chapter 2: Background, theory and background information necessary to understand the solution and further reading into the thesis.

Chapter 3: Solution and design, explaining and showing the solution and its design.

Chapter 4: Simulation, simulating field-tests to analyse the results

Chapter 5: Results, analyze and evaluating the results

Chapter 6: Conclusion.

Chapter 7: Further work.

Chapter 8: References.

1.4 Related work

A similar solution has been researched before, solving the problem in a different way[14]. Similarly, they mounted the drone with sensor to read the standard avalanche transceiver frequency. However, they chose to have an drone with autonomous flight. The drone gets a predetermined area to cover and flies in a grid formation, illustrated in figure 2.

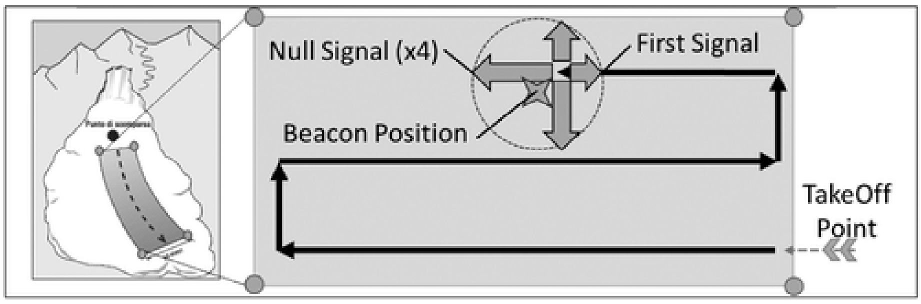

Figure 2: Illustration of grid flight pattern[14]

The moment it reads the first signal, it goes from the grid formation over to cross flight, as seen in figure 2. This means it flies forward until the signal is lost, then flies backwards again until signal is lost. The autopilot thereafter computes the midpoint and does the same procedure on the midpoint, except this time it's rotated 90 degrees forming a cross. This procedure is meant to pinpoint the location of the transceiver. When the drone has located the transceiver, it will either land on the spot or continue to fly in grid formation looking for more victims based on the decision the rescue team makes. They concluded that their solution allows for a quick search process compared to traditional methods and the best performances were in wide areas.

The main difference between the solution mention above and the solution found within this thesis is the drone operational choice, the autonomous flight versus the manual flight. Both solutions are viable and have its unique pros and cons. The autonomous flight relieves a rescue operators piloting assignment and has a very high locating accuracy. However, by using manual flight one could argue that it would be faster because the pilot can follow visual information and experience. If he spots tracks for example, he could fly the drone towards that location with minimal delay, whilst the autonomous flight has to complete its entire grid search before reaching that particular location. Another significant difference is the choice of presenting the results to the users. In the solution above the standard method is for the drone to land at the location of the avalanche transceiver, and given instructions from the operators, it will proceed with the search. This intent of the solution is to present the operators with a map over the entire area marking the probable locations of the victims, which could potentially allow for faster rescues in a multiple victim scenario.

Other projects researching a similar solution exist. However, these are mostly unfinished and requires further work, such as AvaDrone, which was researched but not completed[5]. In the AvaDrone project it was attempted to create a drone system with similarities to the one above using autonomous flight, but did not get so far and concluded with creating a drone system based on existing technology is realistic. There has also been researched a multi UAV solution. However, this was done on an abstract level and not an actual implementation[3].

2 Background

2.1 Avalanche survival and rescue

Majority of avalanche deaths are caused by asphyxiation due to lack of oxygen since victims are trapped in an airtight area and rescue arrives too late. In the period from 01.10.1980 to 30.09.2005 in Canada 81.1% of avalanche related deaths were due to asphyxiation while 18.9% was trauma related[8]. This means up to about 80% of avalanche victims could potentially be saved by enhancing the rescue operations and shorten the search duration.

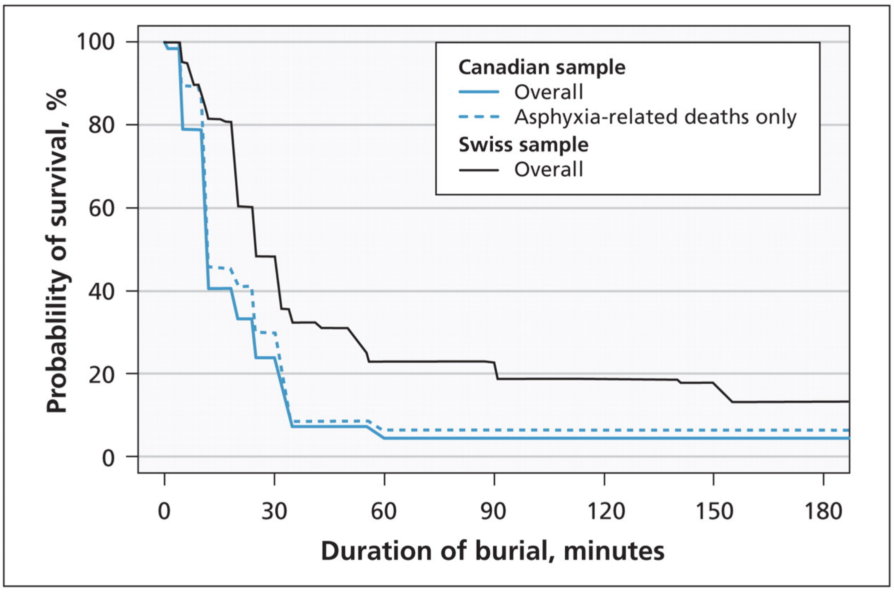

Figure 3: Overall survival curve 1980 to 2005[8]

The rescue time is crucial to the victim's survival chances, as shown in figure 3, about 90% of victims rescued in the first 10 minutes survives, and the survival chances drops drastically after that period.

As shown, the time is essential in avalanche rescue scenarios, where most of the time goes towards the search and localization processes. The process is quite similar for both analog and digital devices used in the search. However, the digital makes the process easier and therefore also faster. The current localization process can be divided into three major phases, signal search, coarse search and fine search.[2]

Signal search, the primary goal to find a signal from a transceiver by moving throughout the avalanche area. Once a signal has been registered the team goes into phase two.

Course search, the objective is to move around the area where the signal was received and attempt to identify the direction towards the transmitter. The use of a digital avalanche transceiver makes this quite easy due to the directional output from the transceiver, but if the team is utilizing an analog transceiver the objective becomes drastically more difficult since unlike digital device, an analog device does not give directional feedback to the user. Once the direction has been identified and the team moves towards the source of the signal the last phase begins.

Fine search, the team starts a slow grid search in the small area where they attempt to find the point with strongest signal strength while holding the transceiver as close as possible to the surface. Alternatively, instead of using the grid method they could use a cross method, similar to the technique described in related work chapter.

2.2 Avalanche Transceiver

Avalanche Transceiver is a commonly used device as a means to improve survivability in avalanche scenarios and is used by both potential victims and rescuers. The device has two core functions, transmitting and receiving. It is used as a transmitter by people who consider themselves potential victims, like mountaineers or skiers. The device periodically transmits as shown in figure 4.

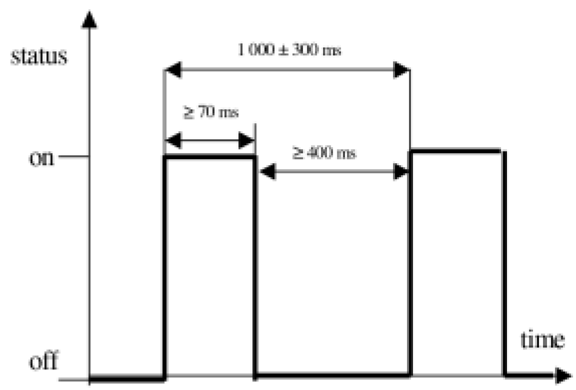

Figure 4: Avalanche Transceiver transmitting period[7]

The time period between the signals is about 1 second which allows the signal to be received, but also restricts the energy usage of the device on transmitting mode. The device can quickly be changed to a receiver with the click of a button which allows nearby people to use the same device to search for surrounding victims. Avalanche transceivers are using a standard frequency of 457kHz defined by ETSI EN 300178[7] which allows compatibility between different models and manufacturers. Figure 5 illustrates how the magnetic field emitted by the avalanche transceiver and how it appears on the surface with a single antenna device.

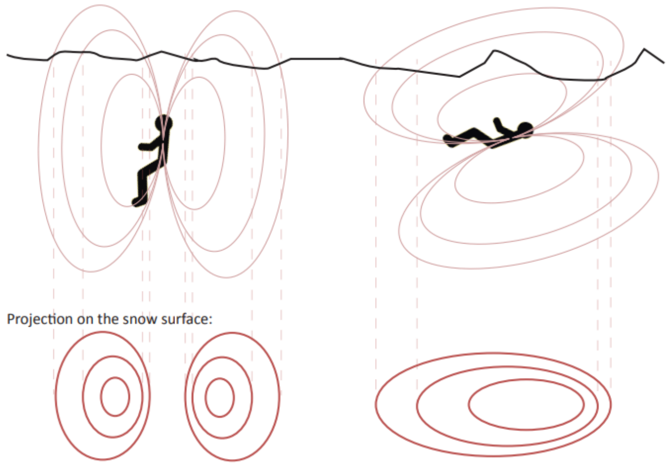

Figure 5: Magnetic Field surrounding transceiver[12]

The orientation of the transceiver determines how the field appears on the surface level, which complicates the search process. The analogue transceivers reads the signal and converts it to a sound representing the strength of the signal which means the analogue design does not give the user any directional feedback. Combining the field projection on surface level variation based on the orientation of the device and analogue devices giving no directional feedback, this means searching with such device requires experience to be done efficiently. However, there is another type of avalanche transceiver, digital, which is easier to use. Due to the standardized frequency, both analog and digital devices work together. The digital version gives directional output and has multiple antennas. The multiple antennas give the advantage of mitigating the negative effects from the devices orientation which is shown in figure 5. Digital transceivers have built in signal processing to have more functionality, such as directional output, supporting multiple victims.

2.3 Geographic coordinate system

Geographic coordinate system is a system that allows any place on earth to be mapped by coordinates. [9]

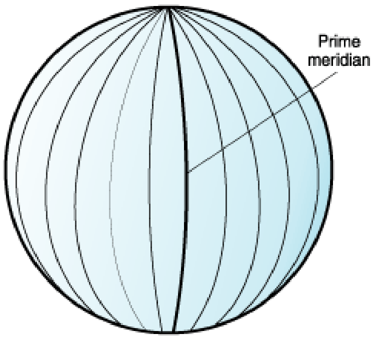

Figure 6: Illustration of latitude lines

Latitude represents where on earth this point is between north and south by drawing lines around the globe that goes through the poles, illustrated by figure 6. Every line has a fixed value and distance, therefore the value to distance ratio is constant. This means regardless of the location, one latitude degree always maps to the same value, unlike longitude lines.

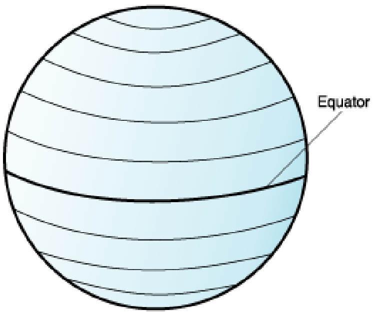

Figure 7: Illustration of longitude lines

Longitude represent where on earth this point is between west and east by drawing lines from north to east, illustrated by figure 7. Every line is parallel with a fixed value. However, the length of the line decreases the closer to the poles it is. This causes a dynamic ratio between value and distance which means one longitude degree is not bound to a fixed distance, instead it depends on how far north/south it is. An example is shown in figure 8.

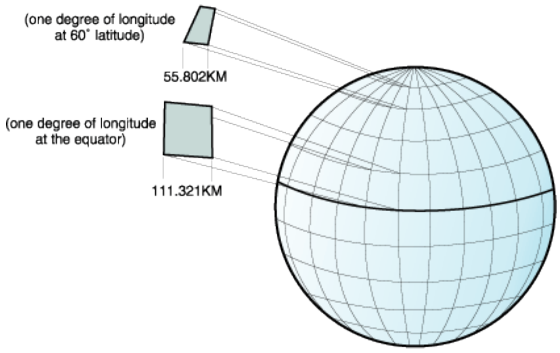

Figure 8: Illustration of the graticule

Combining the latitude and longitude lines defines a grid, called graticule. At the equator the latitude has the largest distance/value ratio where it is almost 1:1 ratio with longitude. While the further north or south you get, the ratio shifts because of the dynamic value of latitude. This makes it complicated to calculate a distance based on coordinates. The figure on the right side shows an example of the variation. (Numbers are based on Clarke1866 spheroid).

Coordinates can refer a fixed point on earth by using both longitude and latitude values. There are multiple formats to how to represent these values, such as:

- Decimal Degree(DD): Representing coordinates by a single value for latitude and a single value for longitude in the format: latitude°, longitude°.
  Example: -5.12345°, 165.54321°
- Degree Decimal Minutes(DDM): Representing coordinates by using degrees and decimal minutes in the format: Latitude degrees° latitude minutes' South/North, Longitude degrees° longitude minutes' East/West.
  Example: 5° 7.407' S, 165° 32.5926 E
- Degree Minutes Seconds(DMS): Representing coordinates by using degrees, minutes and seconds in the format: Latitude degrees° latitude minutes' latitude seconds" South/North, Longitude degrees° longitude minutes' longitude seconds" East/West.
  Example: 5° 7' 24.42" S, 165° 32' 35.556" E

In the formats above DD has negative values in the range and therefore when written in DDM or DMS it has to be specified North/West/South/East because those formats do not include negative values. Latitude ranges from -90 to 90 in DD format, while in DDM and DMS it ranges from 0 to 89 which means it needs to specify whether it is North or South. Longitude ranges from -180 to 180 in DD format, while in DDM and DMS it ranges from 0 to 179 which means this needs to specify whether i is West or East.

2.4 Hardware

2.4.1 Drone

A drone suited for this usage needs to be both portable and lightweight, such that it is easy to carry it into the field. It has to be weather resistant, to sustain in weather conditions like cold, rain, snow and wind. Battery life is crucial because the search must not be interrupted due to lack of power. The drone used is a DJI Mavic Pro 2 which has the following specifications.[6]

- Takeoff Weight: 907g
- Max Speed (near sea level, no wind): 72 kph
- Folded Dimensions: 214x91x84mm
- Max Flight Time: 31 minutes (consistent speed and no wind)
- Max Wind Speed Resistance: 29-38km/h
- Operating Temperature Range: -10°C-40°C

2.4.2 Transceiver

The transceiver used in the original hardware system was an Ortovox M1 which is an analogue device with a single antenna released in 1998. The device was replaced in the simulations due to its age and performance. The avalanche transceiver used in the simulations are purely based on the 3-antenna property and the signal strength, not using additional data from the transceiver, such as directional data. Therefore, any 3-antenna device can be used with the solution.

2.5 Noise

Noise is broad term for something unwanted that causes disturbance, more specifically, in this case it is something that impacts or modifies the signals received by devices. There are various types of noise that impacts signals, but in this case the most relevant type is the additive noise. This is noise that modifies the original signal by adding something more. White noise is an additive noise, which is constant noise with a finite variance[15]. Allowing the impact to be reduced by identifying the variance levels, assuming the signals of interest has a frequency outside this range.

2.6 ArcGIS SDK for Java

ArcGIS Runtime SDK for java is an SDK which gives various functionalities revolving around maps. It is mainly used to develop native mapping and location analysis. It allows you to build maps and add layers including graphical layers, image layers, grid layers and more. It can be used to collect data and compare it to ArcGIS database for further analysis. It includes capabilities such as geocoding and routing for various platforms. Which means it can be used to return a map of an area when it's fed an input, such as address. It has routing capabilities which allows it to generate routes between points.

3 Design and solution

3.1 Design and development process

The main goal is to locate potential avalanche victims and this is done by flying the drone over the avalanche area. The drone is equipped with an avalanche transceiver which is connected to an analog-to-digital(AD) converter. The AD converter samples the analog signal and passes it through a bandpass filter to remove noise before it's sent to a raspberry board. There is a GPS device mounted on the drone, which also sends data to the raspberry board. The raspberry board collects and timestamps the data from both the GPS and AD converter, which allows the usage of interpolation to synchronize the data. This results in an improved data accuracy. The data gets transmitted to a ground station in real time where it proceeds to be processed and analysed.

The software was developed with simple, but effective method, inspired by an agile mindset. By identifying the current issue and implement what gives value at this moment and by welcoming change. This gave a natural start to begin developing the input section. Creating a custom parser for the input format that is being fed by the drone system. Thereafter, a new problem identified was how to process the data, giving the solution of converting it to grid, such that it could be processed. By developing with such a method, the software flow shown by figure 9 was mainly implemented in the very same order, where every cell represents an iteration. Every iteration allowed for optimization of earlier produced code based on new findings.

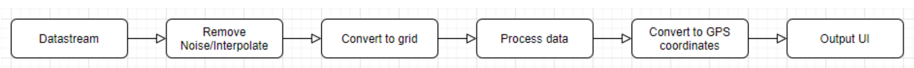

Figure 9: Software flow

The software begins with processing the datastream sent by the drone system, where it begins with removing the stable and continuous noise readings caused by the drone system. Thereafter the GPS data gets interpolated to match the timer on the signal readings, such that the GPS data and signal reading has identical timer and can therefore be merged to a single dataset. The datasets are then plotted into a grid based on the GPS coordinates. While the data is in a grid format it gets processed updating the surrounding cells, which represent the area. After the processing, the grid gets converted back to GPS coordinates and creates new datasets, consisting of GPS coordinates and signal strength, which are then outputted to the user interface.

3.2 Solution

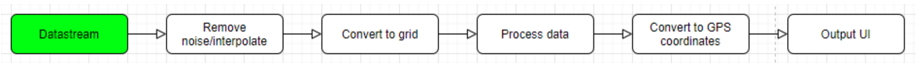

Figure 10

The first action is to read the data coming from the drone system. Figure 11 shows a snippet from the datafile, which displays an example of what the stream looks like. The datastream consist of two elements, the data collected from the avalanche transceiver and the data collected from the GPS device.

```
 1 0.509,1,$OXSIG,99,*4A
 2 1.115,2,$OXSIG,99,*4A
 3 1.377,3,$GPGGA,193709.00,6027.78496,N,00519.73816,E,1,05,1.54,100.9,M,43.8,M,,*5A
 4 1.733,4,$OXSIG,99,*4A
 5 2.105,5,$GPGGA,193710.00,6027.78481,N,00519.73818,E,1,05,1.54,100.6,M,43.8,M,,*55
 6 2.338,6,$OXSIG,99,*4A
 7 2.952,7,$OXSIG,100,*7B
 8 3.214,8,$GPGGA,193711.00,6027.78475,N,00519.73819,E,1,05,1.54,100.5,M,43.8,M,,*5D
 9 3.558,9,$OXSIG,99,*4A
10 4.159,10,$OXSIG,143,*7C
11 4.214,11,$GPGGA,193712.00,6027.78473,N,00519.73834,E,1,05,1.54,100.2,M,43.8,M,,*50
12 4.778,12,$OXSIG,99,*4A
13 5.174,13,$GPGGA,193713.00,6027.78484,N,00519.73874,E,1,05,1.54,100.0,M,43.8,M,,*5F
14 5.383,14,$OXSIG,100,*7B
15 5.991,15,$OXSIG,99,*4A
16 6.185,16,$GPGGA,193714.00,6027.78480,N,00519.73856,E,1,05,1.54,100.1,M,43.8,M,,*5D
17 6.609,17,$OXSIG,99,*4A
18 7.209,18,$OXSIG,100,*7B
19 7.245,19,$GPGGA,193715.00,6027.78473,N,00519.73855,E,1,05,1.54,99.9,M,43.8,M,,*6A
20 7.841,20,$OXSIG,97,*44
21 8.448,21,$OXSIG,100,*7B
22 8.574,22,$GPGGA,193716.00,6027.78466,N,00519.73847,E,1,05,1.54,99.6,M,43.8,M,,*61
23 9.057,23,$OXSIG,99,*4A
24 9.605,24,$GPGGA,193717.00,6027.78475,N,00519.73870,E,1,05,1.54,99.7,M,43.8,M,,*67
25 9.658,25,$OXSIG,134,*7C
26 10.268,26,$OXSIG,105,*7E
```

Figure 11: Data file

The first number on each line is the timer, and the $OXSIG tag indicates signal input while $GPGGA indicates GPS input. The first line on the figure is a signal input, which is following the format:

Timer, packet number, type tag, signal strength, checksum.
"0.509" is the time spent from system boot to input received.
"1" is packet number, which is used to determine if packets has been lost.
"$OXSIG" is type tag, indicating this is a signal input.
"99" is the signal strength.
"*4A" is the checksum, to validate the integrity of the data.

An example of a GPS input is line 3 on the figure, the first two values are added by the system, and the following values follows the NMEA GPGGA format[13].
"1.377" is the time spent from system boot to input received.
"3" is packet number, which is used to determine if packets has been lost.
"$GPGGA" is the type tag, indicating this is a GPS input.

"193709.00" is the UTC time of the position.
"6027.78496" is the latitude coordinate.
"N" indicates whether the latitude degree is north or south.
"00519.73816" is the longitude coordinate.
"E" indicates whether the longitude degree is east or west.
"1" indicates the GPS quality, 0 for invalid, 1 for valid.
"05" number of satellites in use.
"1.54" horizontal dilution of position
"100.9" Altitude, above/below sea level
"M" altitude units, M indicates meters, which means it's 100.9 meters above the sea level.
"43.8" geoidal separation, the difference between WGS-84 ellipsoid and sea level.
"M" geoidal separation unit, M indicates meters.
The next two are blank space, which is followed by "*5A" which is the checksum to validate the integrity of the data.

The file gets processed from top to bottom while being filtered, the system jumps to the 3rd value of every line, the type tag, and distributes the input based on the result. Creating separate lists for GPS inputs and sensor inputs. The sensor input indicates how close to the avalanche transceiver the drone was when it received the input. However, this is not useful information unless we know where the drone was when the sensor received the signal. The GPS and avalanche transceiver are separate independent devices, which means the data collection is not synchronized and therefore the timers does not align. Merging GPS and sensor input to datasets allows for further analysis as it indicates both where and how strong the signal was.

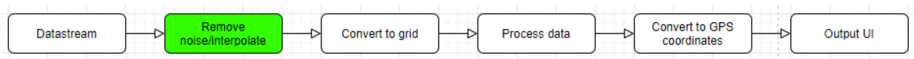

Figure 12

Creating datasets based on the GPS and sensor input can be done by interpolation. There are multiple interpolation techniques and two different ways to apply them to this case. Interpolating to find GPS for a specific sensor input, or interpolate to find sensor data for a specific GPS input. In both cases time is the main factor in the interpolation, to interpolate to find sensor data for specific GPS input. The GPS input has a fixed time, and by using that time, two different sensor inputs are chosen. One input that was collected before, and one that was collected after the GPS input, with minimal variation. For example, using figure 11, if line 11 is chosen as the GPS input, the fixed time is 4.214. Now a sensor reading prior to this time will be picked, with minimal variance, line 10. Lastly a sensor reading after 4.214 will be picked, again with minimal variance, line 12.

Time: 4.158, Strength: 143dB
Time: 4.214, Strength: Unknown(this is the time of the GPS input, goal is to find the strength at this point)
Time: 4.778, Strength: 99dB

Illustrating interpolation on using these values using linear technique, linear interpolation should not be used on signal that does not have linear properties, this is chosen in this instance purely for concept illustration purposes.

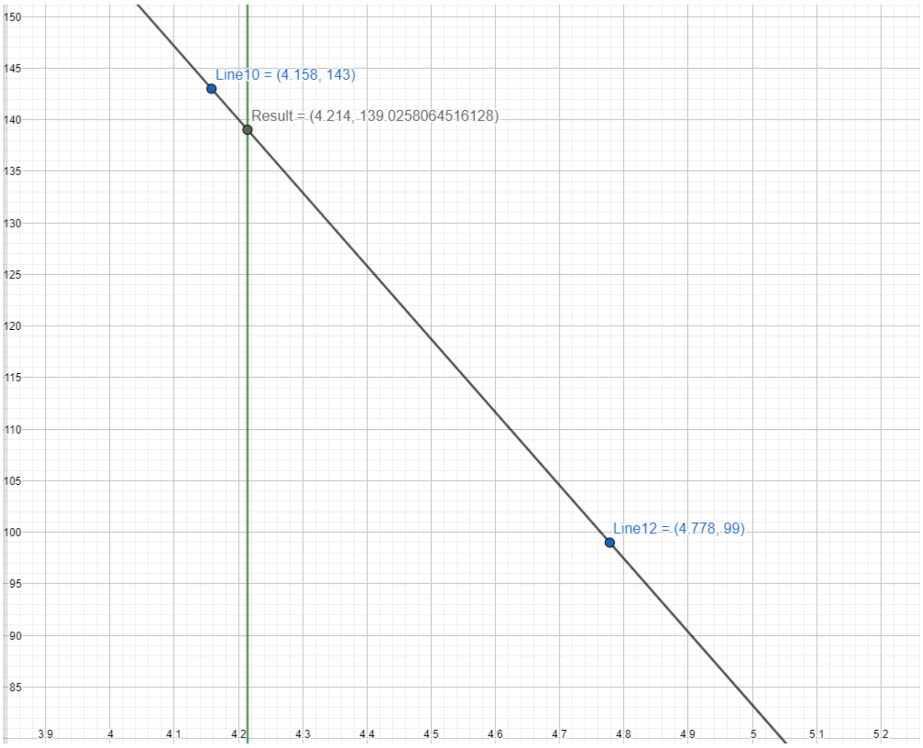

Figure 13: Linear interpolation to find the signal for the GPS input at line 11 which is shown by the point named Result

Given the points (x_0, y_0) and (x_1, y_1) one can find a value y for any x in the interval (x_0, x_1) with the formula:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}$$

Applying this on the example above gives:

$$y = 143 + (4.214 - 4.158)\frac{99 - 143}{4.778 - 4.158} = 139$$

The interpolation gives a strength level of 139dB, which can be paired together with GPS value given by the fixed time. Giving a dataset with the format $(coordinates, signalstrength)$.

There are various consequences of interpolating to find signal for GPS coordinates. If the noise gets removed prior to the interpolation, it would cause larger gaps in the data, which would cause the interpolation to pick inputs that are too far away and has no relevance to the fixed point it was trying to calculate. However, as shown, interpolation takes two value sets and if the noise readings are still present, a noise input and a valid input could be used to interpolate, which causes the noise input to corrupt the valid input.

In this case, noise is referring to the continuous noise originating in the drone system and its mounted devices. The noise is maintaining a stable level throughout flights and therefore it can be removed by identifying the noise levels. Thereafter, create a threshold, where all the inputs that are below this level will be removed and therefore not affect the results.

For arguments sake, assuming the stable and continuous noise levels are around 100dB, the interpolation example above illustrates what could happen if interpolating happens before noise is being handled. The system interpolates a valid input(143) with a noise input(99) and slightly corrupts the valid input. The impact on this example is quite minimal compared to the potential impact due to the GPS timer(4.214) is close to the valid inputs timer(4.158), as seen in figure 11. The noise readings have the potential to make a valid input almost invisible.

If the noise readings were to be removed prior to interpolation, keeping the assumption the noise levels are around 100dB. All the inputs in figure 11 except for line 10 would be removed because they are considered noise. This causes the interpolating to pick line 10 and next valid input, which has the time 9.658. Looking at the timers of these two inputs, there is clearly a gap between them, and during this period the drone could have flown far making the valid inputs irrelevant for the actual position that was attempted to estimate signal strength.

Interpolating to find GPS coordinates for sensor input avoids the issues mentioned. Instead of finding a signal strength for every coordinate, noise or not, it would be better to find coordinates for every valid sensor input. This allows to remove noise readings before starting to interpolate values. By doing this it reduces the continuous noises impact drastically compared to the other method mentioned above. The amount of data sets will be reduced, but the trustworthiness of these values will be greater. Interpolating to find GPS coordinates for a signal is slightly different from interpolating to find a signal for GPS coordinates. This is because GPS coordinates consists of two values. Finding the coordinates for a signal is done by:

$$\phi = \phi_0 + (x - x_0)\frac{\phi_1 - \phi_0}{x_1 - x_0}$$

$$\lambda = \lambda_0 + (x - x_0)\frac{\lambda_1 - \lambda_0}{x_1 - x_0}$$

Where ϕ, λ, where ϕ is latitude, λ is longitude, and x is time.

Interpolating to find GPS coordinates creates a new issue impacting accuracy. Interpolating the value for latitude has no issues, because as mentioned in chapter 2.3 the distance to degree value is constant, meaning a latitude value always has a fixed distance and therefore linear interpolation can be applied safely. However, longitude behaves differently, the degree to value ratio is dynamic, so applying linear interpolation will impact the accuracy, the question is just how much?

Assuming a bad case scenario to see how inaccurate it can be and establish impact relevance; The longitude degree to distance value depends on how far away from equator the point is, which is determined by the latitude degree. An example of this is shown in figure 8 in chapter 2.3. Due to the longitude and latitude behavior, assuming the drone flies 45 degrees off north, this covers the most change in both longitude and latitude. By assuming the drone flies at 18km/t we can calculate the variation: Figure 11 shows the sensor frequency is about 1 second.

18km/t is the same as 5m/s

5m/s with a sensor frequency of 1 second means the drone would travel 5 meters between the sensor inputs.

Y-axis represents north, which means due to the 45 degree angle off north, the x value equals y value.

Using pytagoras theorem we can determine that $x = y = \sqrt{(\frac{5^2}{2})} = 3.54$meters

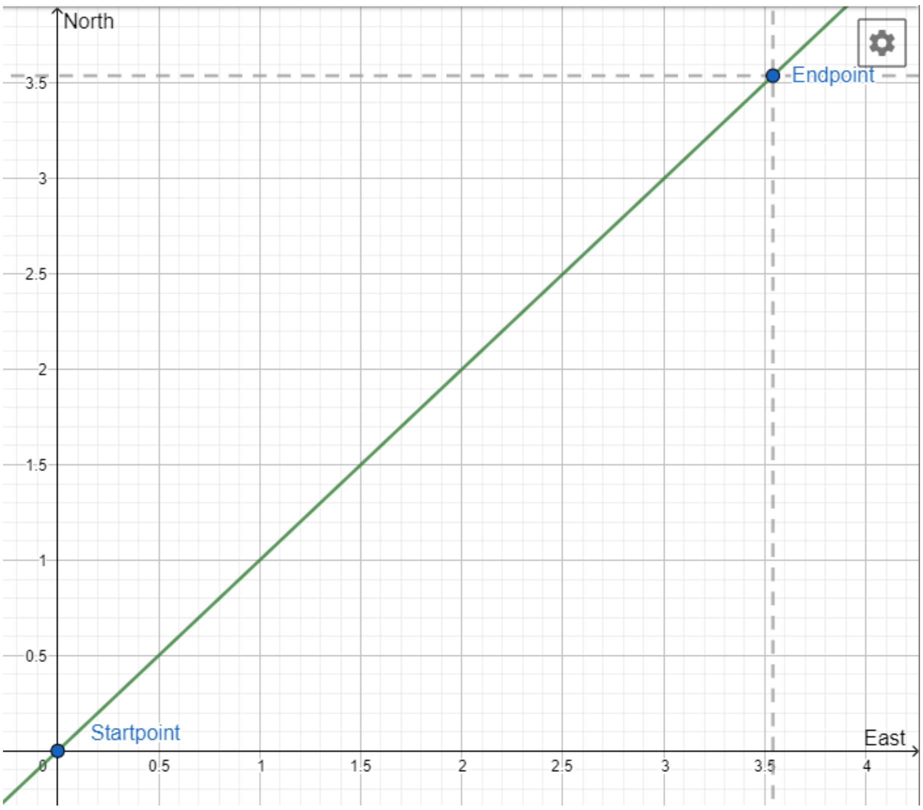

Figure 14

Setting the start point to the coordinates: 60.4630788333333334, 5.3289723333333334.

Traversing 3.54 meters towards east, shown by the red line in figure 15, gives the new coordinates of: 60.463078833333334, 5.329036911422549.

The difference between the longitude value from the starting point and the new point equals to 0.000064578089215.

Figure 15

Doing same procedure as above with a start point that is 3.54 meters further north, point A in the figure, to calculate the difference in longitude value at a different latitude value. Which will show the variation caused by using linear interpolation on coordinates.

The start point now has the coordinates 60.463110669318205, 5.328972333333334.

Traversing 3.54 meters towards east, shown by the blue line in figure 15, gives the new coordinates of: 60.463110669318205, 5.329036911485876.

The difference between the longitude value from the starting point and the new point equals to 0.000064578152542.

Comparing these two different longitude values gives the variation of $6.3327 \cdot 10^{-11}$.

Converting this longitude into meters gives a distance of $3.47 \cdot 10^{-5}$ meters.

This shows that the linear interpolation on coordinates, which are not linear, does impact the accuracy. However, as shown above in the bad case scenario in flight pattern, including a very high flight speed only gives about 0.000035 meters inaccuracy. This is mainly due to the limited travel speed and the sensor reading frequency, which causes the latitude difference between inputs to remain minor. Therefore, the inaccuracy caused by longitude interpolation is irrelevant on an operational level. There are various other sources of inaccuracy which makes the inaccuracy caused by linear interpolation on coordinates minor in comparison.

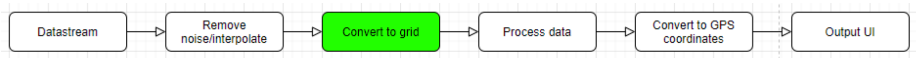

Figure 16

Now the system has the datasets which consists of signal strength and GPS coordinates. Processing and analyzing data in this format is difficult due to the longitude and latitude behavior, therefore the data is converted to a grid where x and y is fixed and always represents a constant length. While in grid it's easier to calculate how the signal affects the area around its location. Converting to a grid is done by selecting a reference point, coordinates that represent a single cell in the grid. Thereafter when the system receives new datasets, it takes the coordinates, calculates the distance between the reference point and the new point. Then the angle from the reference point and new point gets calculated. Using the distance and angle between the two points the variation in x and y value gets calculated, which determines where in the grid the new dataset belongs.

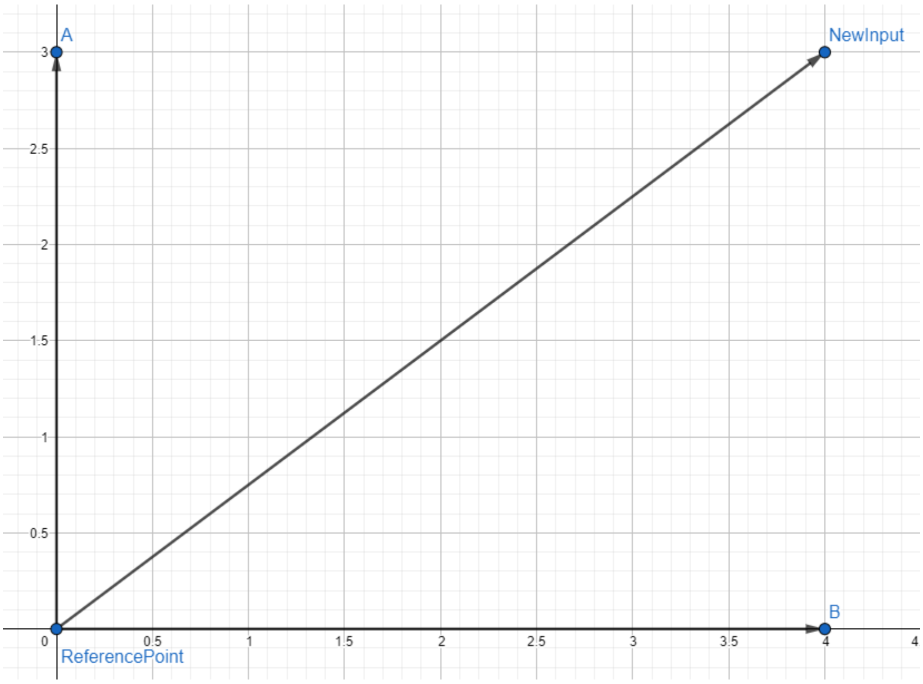

Figure 17

Calculating the distance from the reference point to the new input is done by utilizing the haversine formula:

$$a = sin^2(\Delta\phi/2) + cos\phi_1 \cdot cos\phi_2 \cdot sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Where ϕ is latitude, λ is longitude, R is earths radius and d is distance.

The angle between the reference point and the new input is calculated with the following formula:

$$a = atan2(sin\Delta\lambda \cdot cos\phi_2, cos\phi_1 \cdot sin\phi_2 - sin\phi_1 \cdot cos\phi_2 \cdot cos\Delta\lambda$$

Where ϕ is latitude, λ is longitude.

Now having the distance and angle, finding the x and y value is done by simple trigonometry:

$$x = distance \cdot sin(angle)$$

$$y = distance \cdot cos(angle)$$

Having the x and y value, the new input can now be placed into the grid by counting rows and columns from the reference points location in the grid. If the reference point has the location $[x_1, y_1]$ the new input would be $[x_1 + x, y_1 + y]$. Using figure 17 as an example it would be $[0 + 4, 0 + 3] = [4, 3]$ and the signal strength can then be plotted into that location in the grid.

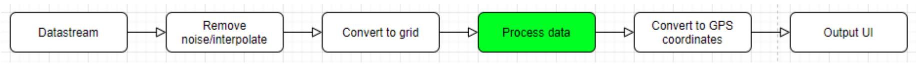

Figure 18

The data in the grid can be processed and determine how a signal should be perceived from the area around the original signal location. When a new input gets plotted into the grid it causes a ripple effect that traverses outwards, updating all the surrounding cells with new signal values. The value for each cell is therefore based on all the inputs received. However, the further away the signal originated, the less impact it has on the cells value since the closer the signal was to said cell, the more accurate it is. The number of inputs every cell receives is based on the speed of the drone. If the drone flies slowly, more signals are received.

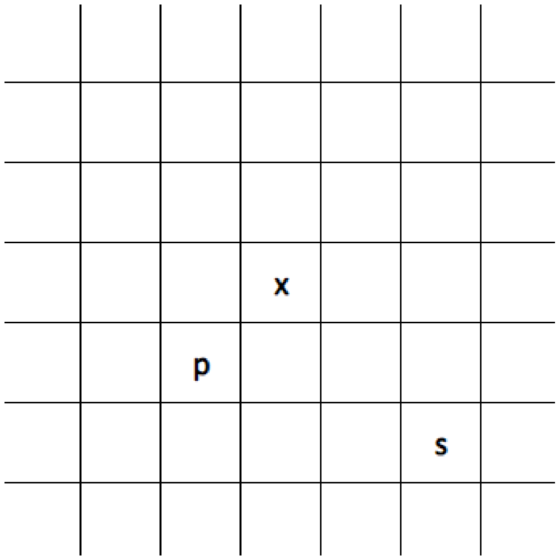

Figure 19

Using figure 19 as example, if a signal is received in the location marked as x. X is a value that represents the signal strength in this location. Therefore, the cell is given the value x. The surrounding cells within a certain distance of the signal's location are also given the value x. If a cell has no preexisting value, it is simply given this value. However, if the cell already has a value, the value gets updated using $x \cdot p + (1 - p) \cdot x_0$ where x is the new input, x_0 is the preexisting value and p is the weight. The weight is based on both distance from the signal's origin and the amount of inputs the cell has received. The weight given by the distance has the reduction properties of $\frac{1}{r^2}$ which is thereafter combined with the amount of inputs the cell has already received. This means if we again look at 19, if x is a new input, it would impact p value a lot more than it would s value, since s is further away from x. If the cell only has received one input prior, the new input will have a bigger weight, and a large impact. However, if the cell has already received many inputs, the new input will have a lesser weight and therefore a smaller impact. This is meant to stabilize the value and make it more trustworthy since it's based on various signal readings and not just a single one. Therefore, if there were noise readings or fake signal it will fade away due to the impact from surrounding signal inputs. The noise readings being referred to are signal inputs that are corrupted due to external noises from various sources, causing the signal to appear stronger than it actually is. Fake readings refer to the avalanche transceiver fails to pick up the signal due to the transmission format seen in figure 5. This means there are anomalies were there are noise readings, such as a reading that picks up signal when there is none. Another type of anomaly is a fake signal, where the reading does not pick up the signal when there actually is a signal. The impact from these anomalies will fade due to the design decision, where a cell value is affected by multiple signals from the surrounding area.

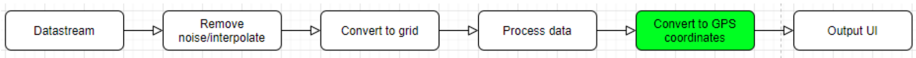

Figure 20

Now having a grid that represents the area with signal strengths calculated for every affected cell, the next step is to present this to the user. Due to the ripple effect explained above, the datasets being outputted are multiplied compared to the datasets that were inputted. Iterating through the grid to create datasets containing signal strengths and GPS coordinates that represents a cell such that it can be plotted into the user interface. The signal strength is simply retrieved from the cell, while the GPS coordinates are calculated using a reference point. The reference point is a cell that was selected upon creating the grid and this cell has an assigned coordinate set. Now using the basic geometric formulas, where x and y is the cells indices, to calculate both the distance and angle between them.

$$Distance = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}$$

$$Angle = atan2(y_0 - y, x_0 - x)$$

Having the GPS coordinates for the reference cell, as well as both the angle and distance between the reference cell and the current cell being processed. It is possible to calculate the GPS coordinates for this cell with the following formula;

$$\phi_2 = asin(sin\phi_1 \cdot cos(d/R) + cos\phi_1 \cdot sin(d/R) \cdot cos\theta)$$

$$\lambda_2 = \lambda_1 + atan2(sin\theta \cdot sin(d/R) \cdot cos\phi_1, cos(d/R) - sin\phi_1 \cdot sin\phi_2)$$

Where ϕ is latitude, λ is longitude, d is distance, R is earth radius, theta is the angle between the two points.

There is now a list of datasets consisting of the strength and GPS coordinates for every cell in the grid.

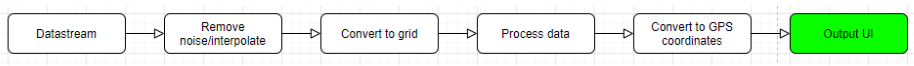

Figure 21

The user interface created with help from ArcGIS functionalities, but despite all the functions the SDK offers, the only functionality used is the retrieval of the map and creation of a graphical layer. The intention was to create an independent solution where the logic does not rely on entities, such as an SDK like ArcGIS. This allows for replacing the mapping service if needed, while the solution remains fully functional on a logical level. The SDK is only used to retrieve the map of the area and the graphical layer is added. Using the graphic layer together with the datasets which consists of GPS coordinates and signal strength, the signal strength gets represented by a color and plotted directly on the graphic layer based on the GPS coordinates. The colors used to represent the signal's strength goes from a green to red scale where red is the strongest signal and green is a weak signal.

4 Simulation

Simulating drone flight has various benefits for testing the system. The main benefit is the control over the entire environment and scenario. By simulating the flight, all hardware issues are bypassed and hardware malfunctions will not impact the results. Using simulation makes it easier to test what-if scenarios and edge cases which shows the systems robustness. While there are various good benefits from simulating there are also advantages to field-tests, because when simulating, one is usually operating on assumptions.

The simulations are based on a few assumptions such as;

- The drone velocity is constant
- The avalanche transceiver on the drone has omnidirectional properties in the near field.
- The distance between the drone and the ground is constant.

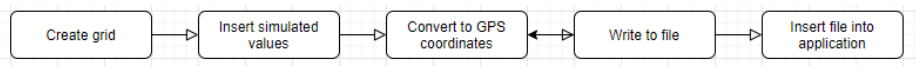

Figure 22: Simulation process

The simulation is done by representing a larger area using a grid and by selecting a single cell as a reference point by giving it a set of coordinates. The cells in the grid represent areas, the default is one cell is 1x1 meters. By selecting a location of the simulated avalanche transceiver and the signal strength in the surrounding cells are given by the formula[2] ;

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

Where H is the signal strength, M is the strength of the field, r is the distance and theta is the angle between the avalanche transceivers and the orientation of the device. Figure 23 shows the distance r with the length of the Q vector.

Vector P represents the devices orientation, P and Q gives the angle that is θ in the formula which is 0° in this case.

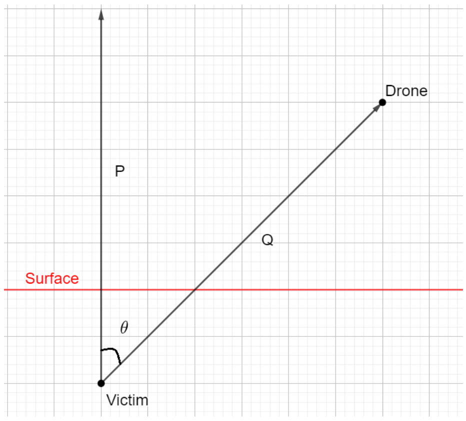

Figure 23: The angle used in formula 1 is shown by θ

Due to the altitude difference between the drone and the buried victim the calculations have to be done in 3-dimensions. The distance between two points in 3d can be calculated by $Dist = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2)}$ but due to the avalanche transceiver being the origin it can be shortened to

$$Dist = \sqrt{x^2 + y^2 + z^2}$$

The angle between the antennas orientation and the vector given by the two avalanche transceivers can be calculated by using the formula $Angle = arccos(\frac{P \cdot Q}{|P||Q|})$ where P is the vector that represents the antennas orientation of the, shown in figure 24, and Q is the vector between the simulated avalanche transceiver, the origin, and the avalanche transceiver on the drone. Since the magnetic field is omnidirectional in the horizontal directions, which happens to be parallel with the surface levels in figure 24, this allows to use a vector in the formula since angle is all that matters, the direction is irrelevant to the signal strength. The P vector represents the orientation of the device, and by rotating the vector itself, the entire field will rotate with it, and it can therefore be used to simulate other orientations of the device. Q can be represented by [x, y, z] since the vector starts in origin, [0, 0, 0]. The x-axis and y-axis is the plane that represents the surface, while the z-axis represents altitude. The simulations will contain the edge cases, where the orientation of the device is 90°, and 0°, as well as 45°. When the orientation of device is spoken about it is the orientation of the currently active antenna where 0 degree orientation is shown by figure 24, 45 degree orientation is shown by figure 26 and 90 degree orientation is shown by figure 25.

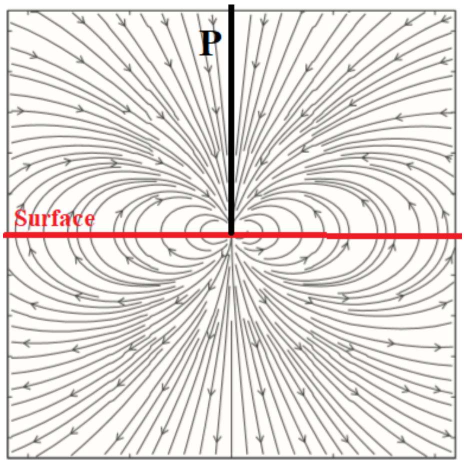

Figure 24: Magnetic field

The 0° case can be simulated by using formula 1 from chapter 4 where the angle is calculated by using a unit vector in the z-axis, [0,0,1], as seen in figure 24 where the green line is the unit vector, and the red line represents the surface plane. The 90° case can be simulated by rotating the entire field 90°, as it would if the avalanche transceiver were rotated 90°. This can be done by instead of using a unit vector for the z-axis, a unit vector for either x or y can be used, therefore using the vector [1,0,0]. This rotates the green vector seen in 24 and causing the field to rotate, illustrated in figure 25. The green line is parallel to the surface plane now, unlike the 0° case. In both cases above the P vector is a unit vector, which simplifies the calculates since the angle formula can be shortened to

$$Angle = arccos(\frac{P \cdot Q}{\sqrt{x^2 + y^2 + z^2}})$$

However, since P is a unit vector $P \cdot Q$ will be simply z in the 0° case and x in the 90° case.

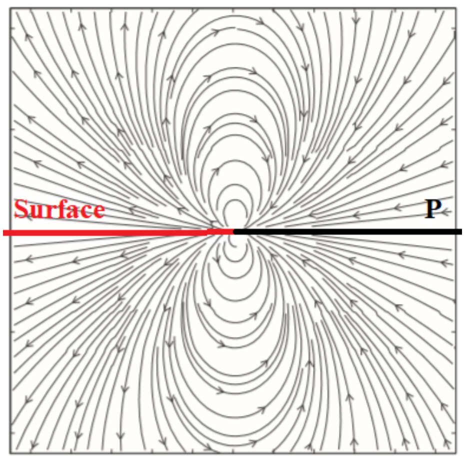

Figure 25: Magnetic field in a 90°orientation

The 45° case is different to simulate since the P vector will not be a unit vector. Choosing a vector [1,0,1] that is 45° off the default case and the angle can be calculated with

$$Angle = arccos(\frac{x + z}{\sqrt{2} \cdot \sqrt{x^2 + y^2 + z^2}})$$

This will result in the simulation of a field rotate 45° over the x axis which is illustrated in figure 26 where the green line represents the orientation of the device and the red line represents the surface.

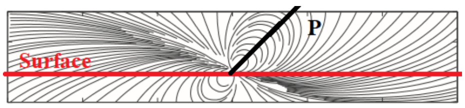

Figure 26: Magnetic field in a 45°orientation

Applying the distance and angle into formula 1 from chapter 4, the strength of the signal at any location can be calculated and therefore simulate a single antenna device on the victim, which means by altering the angle it's possible to simulate different orientations of the avalanche transceiver. This will show how the system handles different scenarios and show the impact of good versus bad orientations.

The application iterates through the grid in a pattern shown in figure 27, where the x's represent plotted values. The preset velocity determines number of cells to skip before plotting a value into a cell, which represents the signal strength. The value H from the formula above gets plotted into the cell, where r is the distance to the avalanche transceiver and theta is the angle. In the current simulations M is set to 7dBμA/m.[11]

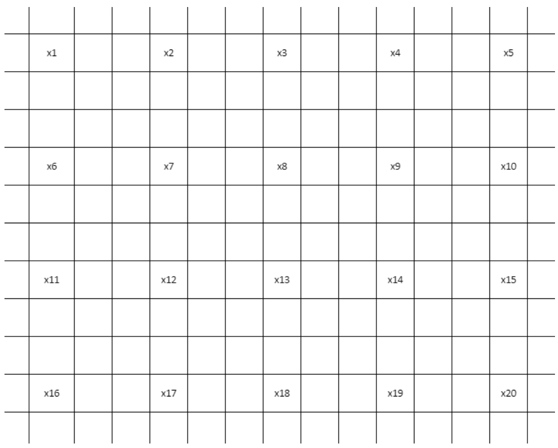

Figure 27

After the grid has been created and values assigned to the selected cells, the system calculates the GPS coordinates for the selected cells. This is done by using the predetermined GPS coordinates as a reference point, which has been assigned to a cell. Using both the reference cell and the current cell indexes as values x and y;

$$Distance = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}$$

$$Angle = atan2(y_0 - y, x_0 - x)$$

The angle calculated is the angle between the signal location, the reference point and north, which are two cells in the grid and therefore 2 dimensional. Now having the reference point's GPS coordinates, and both angle and distance from the selected cell, GPS coordinates for the cell can be calculated.

$$\phi_2 = asin(sin\phi_1 \cdot cos(d/R) + cos\phi_1 \cdot sin(d/R) \cdot cos\theta)$$

$$\lambda_2 = \lambda_1 + atan2(sin\theta \cdot sin(d/R) \cdot cos\phi_1, cos(d/R) - sin\phi_1 \cdot sin\phi_2)$$

Where ϕ is latitude, λ is longitude, d is distance, R is earth radius, theta is the angle between the two points.

At this point the system has a grid representing an area and cells plotted with values that represent the signal strength, and using the formulas above calculating the GPS coordinates for every signal strength. Having these datasets containing signal strength and GPS coordinates, the system adds timers to this data with a small deviation. The reason for adding the minor deviations on the timers is due to the GPS device and the avalanche transceivers receiving systems are not synchronized, as mention in chapter 3, therefore deviation must be added to simulate more realistically and test the interpolation module. Now this gets written to a file with the same format as the file in figure 11 and can therefore be inserted into the program to test the functionalities.

The figures 28, 29 and 30 are the results of the three different simulations that were ran where core difference is the victim's device orientation. This has been altered by changing the θ value from the magnetic field formula 1 located in the beginning of this chapter. These three simulations are not taking into account the inaccuracy caused by the GPS device. The simulations below and all the simulations results further into the thesis will have a black cross marking the actual location of the transceiver such that the results can be compared to the true location of the victim. The figures are a capture of the results that represent a larger area where the color indicates the signal's strength, with a scale from green to red where red is the stronger signal.

Figure 28: Capture of the results with a 0 degree orientation of the avalanche device and a speed of 5m/s

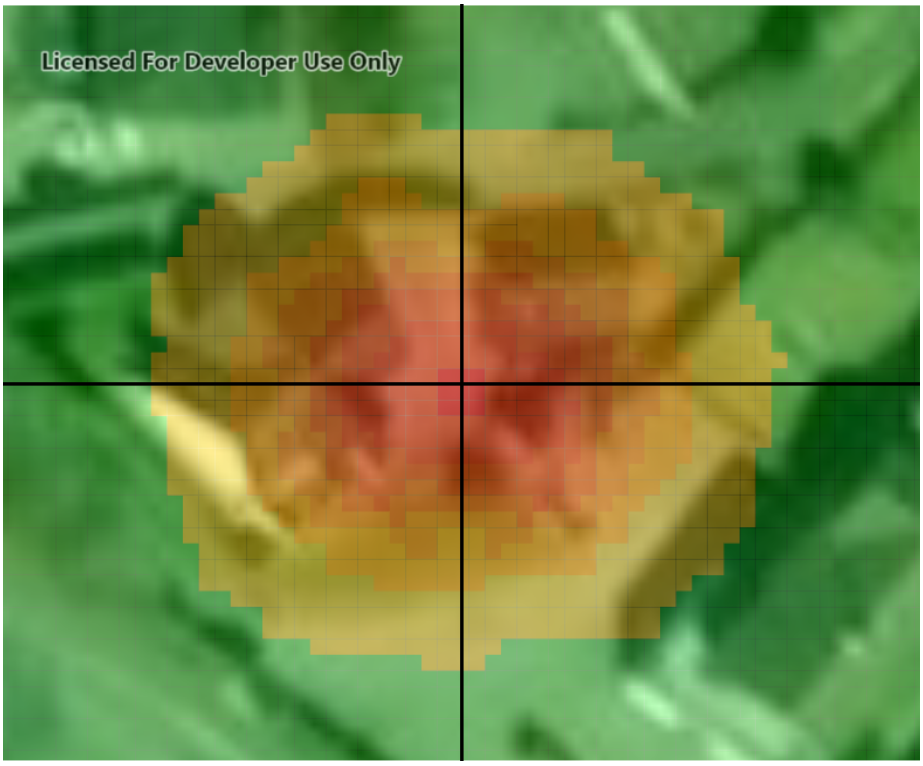

Figure 29: Capture of the output with a 45 degree orientation of the avalanche device and a speed of 5m/s

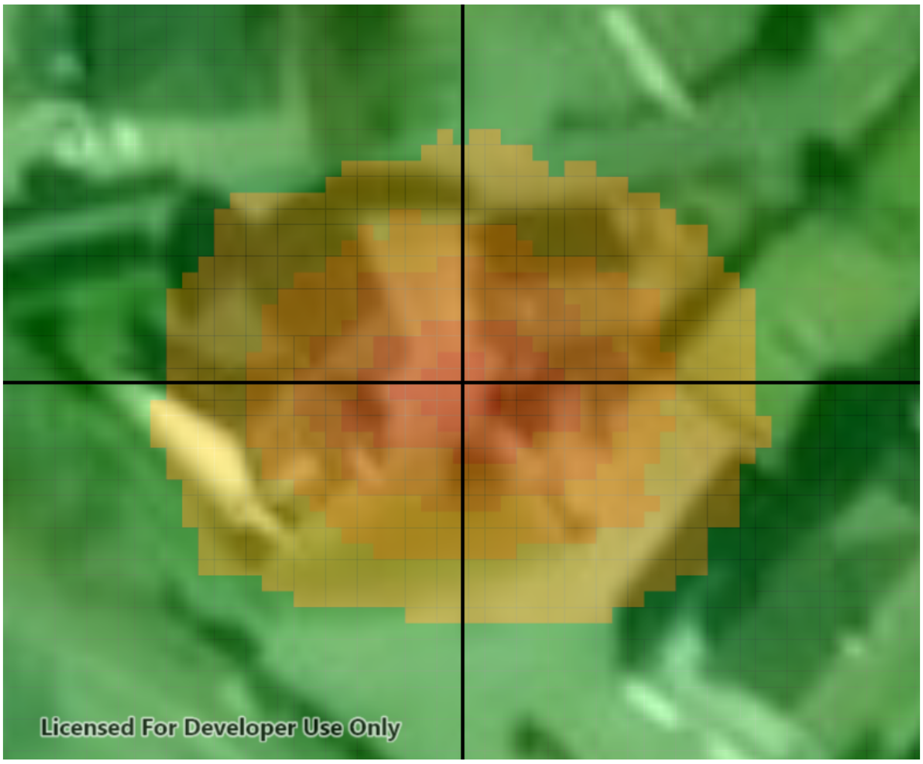

Figure 30: Capture of the output with a 90 degree orientation of the avalanche device and a speed of 5m/s

As witnessed in the figure 30 the results are weaker because of the weaker signals received with a 90 degree orientation. However, this scenario can only occur with an out of date avalanche transceiver that has 1-antenna and is rotated 90 degrees.

As mentioned in the beginning of this chapter, the simulations are working under a few assumptions. One of the assumptions were that the avalanche transceiver device on the drone has omnidirectional properties in the near field. This is justified by looking at figure 31.

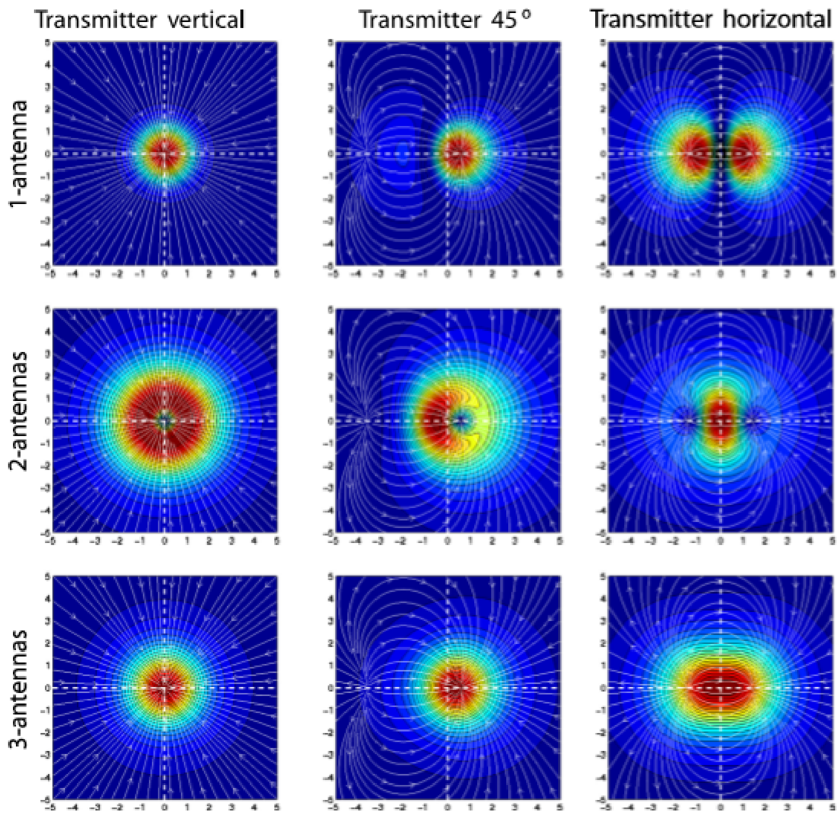

Figure 31: Magnetic field strengths perceived by avalanche transceivers with different amount of antennas[2]

A quick look at the 1-antenna device imagines shows that it fails to pick up signals for example at horizontal orientation of the victim's device and causes there to be two signal peaks. Looking at the 2-antenna device it also struggles with an accurate image of the horizontal orientation. However, by looking at the 3-antenna device it shows that regardless of the orientation of the victim's device, it manages to pick up all the signals due to the 3-antenna design. These images therefore show that the 3-antenna device has omnidirectional properties in near field therefore by expecting the system to use 3-antenna device the assumption for the simulation is justified.

5 Results

Modern avalanche transceivers have at least two antennas, this is because it allows to swap between the antennas based on necessity. Multiple antennas in a search situation allows the device to easier pick up signals, thus locate the victim. However, in a victim scenario the benefits of multiple antennas are also great. By taking a look at figures 32, 33 and 34 we can see the difference in signal strengths based on the orientation of the victim's single antenna device. The graphs show the signal strengths of a flight with 10 meter height over the victim's transceiver.

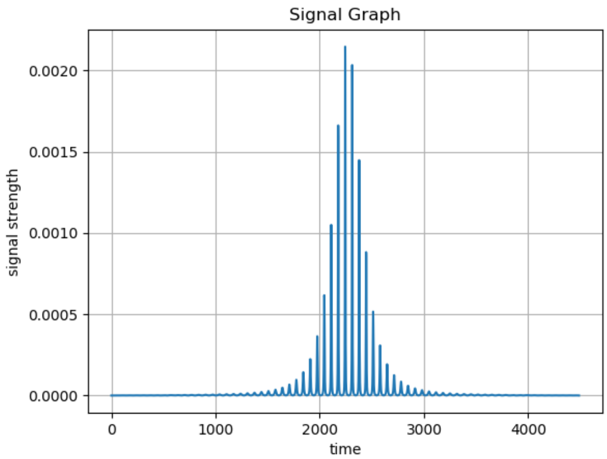

Figure 32: Graph showing signal strength over time at 0 degree orientation

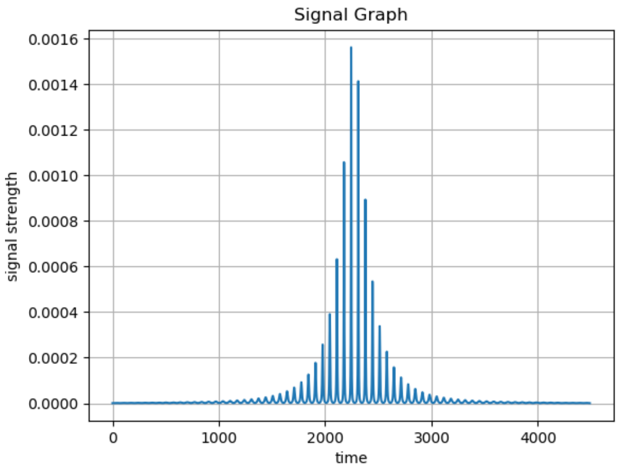

Figure 33: Graph showing signal strength over time at 45 degree orientation

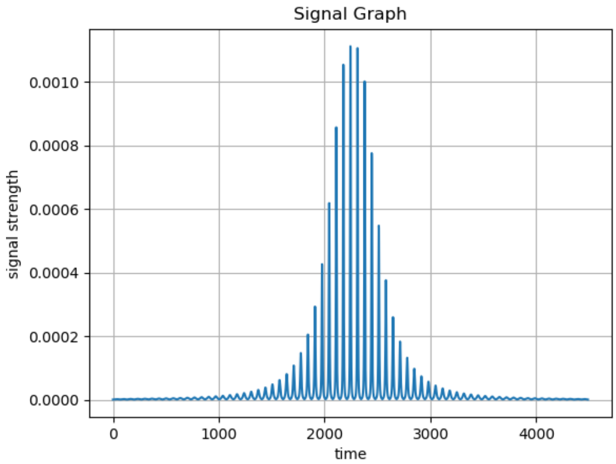

Figure 34: Graph showing signal strength over time at 90 degree orientation

The magnetic field used to simulate the single antenna device is the formula 1 from chapter 4. By taking a look at the formula and extracting the right side, $\sqrt{1+3cos^2\theta}$, which is a multiplier, we can see that at best this will be $\sqrt{1+3\cdot 1}$ which is 2, while at worst will be $\sqrt{1+3\cdot 0}$ which is 1. This means that the best orientation will have up to 2x signal strength compared to the worst orientation, which we can see by comparing the peak values of the simulations seen in figure 32 and 34. However, as modern devices have multiple antennas[1], the scenario where the antenna is rotated 90 degrees is rare since multiple antenna devices would simply swap the transmission to another antenna and therefore this scenario will only occur when the victim's device is a single antenna device. Realistically the most common scenarios will therefore be with the orientation ranging between 0-45 degrees.

5.1 Signal location accuracy

The accuracy of the signal's location compared to its true location gets affected by multiple factors. The main factor is the accuracy of the GPS device itself, which will vary based on hardware. The chapter 3.2 describes the design of the application, and as mentioned the GPS coordinates read by the applications gets converted into a grid, where any GPS coordinate located within an area gets assigned to the cell. After the cells are processed and updated based on surrounding signal inputs, the cells are converted back to GPS coordinates. When converting back, the cells GPS coordinates are calculated from the center of the area the cell represents, which means in a worst case scenario, the signal's true location is in a corner of the square area. This means that a signal at worst case scenario can be rounded from a corner of the area to the center, while default resolution of the grid is 1x1 meter area per cell, causing a maximum potential inaccuracy of $\frac{\sqrt{1^2+1^2}}{2} = \frac{\sqrt{2}}{2} \approx 0.7$ meters. This could be reduced by changing the grid resolution, which will cause the processing resource costs to increase drastically. Another factor impacting the accuracy is the linear interpolation on GPS coordinates which was shown to be insignificantly small compared to the inaccuracy from the GPS, thus ignorable. At a high flight speed and a bad flight angle it could cause an inaccuracy of $3.47 \cdot 10^{-5}$ meters, which was calculated in chapter 3.2.

The GPS devices inaccuracy depends on hardware, GPSmap 76S is a well performing GPS device and by using it can be expected to be approximately 1.4 meters inaccuracy caused by the GPS[17]. GPS coordinates are used twice for every value, for the signal itself, but also for the reference point used to convert the grid back to GPS coordinates. This means the inaccuracy from GPS device has to be counted twice, but independently, meaning both the amount of deviation and the direction is unrelated to each other. It can be expected that 95% of the readings from the GPS device has less than 2.6 meters inaccuracy[17].

The four accuracy impacting factors mentioned above are all independent, which means that they are unrelated to each other and this means the direction of the deviation is not necessarily the same. Since for 95% of the GPS device readings the inaccuracy is below 2.6 meters, so realistically the chance that both GPS readings have a deviation higher than 2.6 meters and they have the same general direction is quite low. Defining general direction by a 30° cone, out of 360 degrees this is $\frac{360}{30} = 12$, and it is 5% chance for a inaccuracy to be above 2.6 meters and the solution relies on GPS device for the reference point as well. This all gives $0.05 \cdot 0.05 \cdot \frac{1}{12} \cdot 100 \approx 0.02\%$ chance that both readings have a deviation higher than 2.6 meters and have the same general direction. The chance for two average readings has the same general directional deviation is given by $\frac{1}{12} \cdot 100 \approx 8.3\%$ which causes deviations up to 2.8 meters. Two examples are shown in figure 35 where A represents the actual location, u represents the GPS accuracy of the first signal with an arbitrary direction. The first example is represented by w, where b is the second inaccuracy source which actually causes the overall accuracy to be lower. The second example is represented by a, where the second inaccuracy source is represented by v. This causes the total inaccuracy to increase greatly, which is what is being calculated above. This shows that the independent sources of inaccuracy can both reduce inaccuracy, but also increase the inaccuracy.

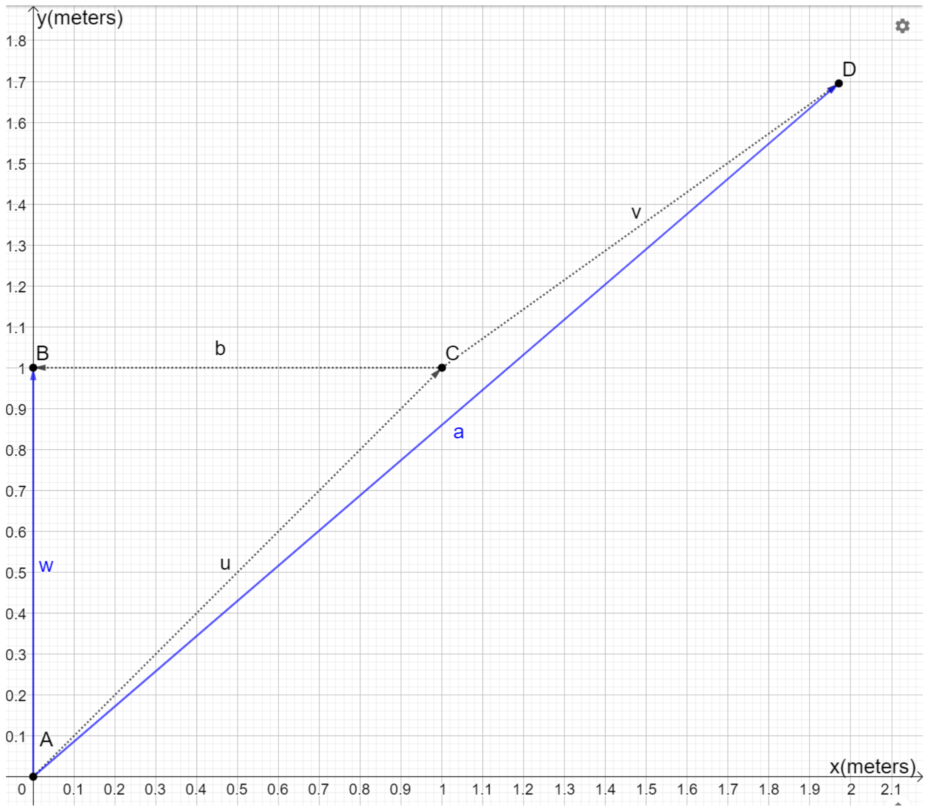

Figure 35

This is excluding the deviation caused by the coordinates to grid conversion which can add or reduce the inaccuracy by an additional 0.7 meters. This means that half of the 8.3% readings will have an inaccuracy in the range 2.8-3.5 meters while the other half will have inaccuracy in the range 2.1-2.8. Realistically it's therefore expected to have inaccuracy up to 3.5 meters from the true location of the signal.

In order to see the potential impact from inaccuracy on the overall results, simulations with added inaccuracy caused by the GPS device were ran. The figures 37 and 38 are the results of the simulations done where the inaccuracy was added on both the signal inputs and the reference point used in the simulations. The inaccuracy was added using

$$x = x_0 + R_1 \cdot cos(A_1) + R_2 \cdot cos(A_2)$$

$$y = y_0 + R_1 \cdot sin(A_1) + R_2 \cdot sin(A_2)$$

Where R_1 and R_2 are both distances chosen randomly between 0-2.6 meters, one for the signal input and one for the reference point,. A_1 and A_2 are angles representing the direction of the inaccuracies chosen randomly in the range 0 to 360°. Figure 36 shows 1000 plotted cases of inaccuracy to illustrate the distribution.

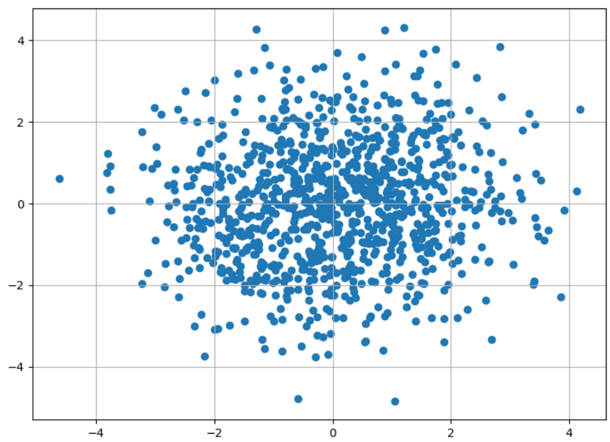

Figure 36: Simulated inaccuracy caused by GPS device for both signal input and reference point. Both axis represents distance from the signal's true location in meters

By comparing the results in figure 37 and 38 to the earlier simulations done without the GPS inaccuracy, shown in figures 28 and 29, it is clear that at the maximum recommended flight speed the inaccuracy impacts the results, but the victims predicted location is still within the marked area. The most important result of the application is location of the strongest signals which shows the location of the victim. By looking at the figures we can see both results points towards the same area with some deviations. This is because of the limited flight speed, which therefore allows every cells value to be based on multiple signals and therefore balance out the inaccuracy, but if the speed was higher the combination of the lack of signals and the inaccuracies could have critical impacts on the results, as shown in figure 41. It is important to note that in these simulations the entire flight is done at the maximum recommended operational speed and if it were flying slightly slower the results would be even more accurate. In simulations there is the benefit of not being affected by hardware and performance. However, in order to simulate as realistically as possibly hardware and its performance has to be taken into account. This is why inaccuracy from the GPS device has been considered and added to the simulations.

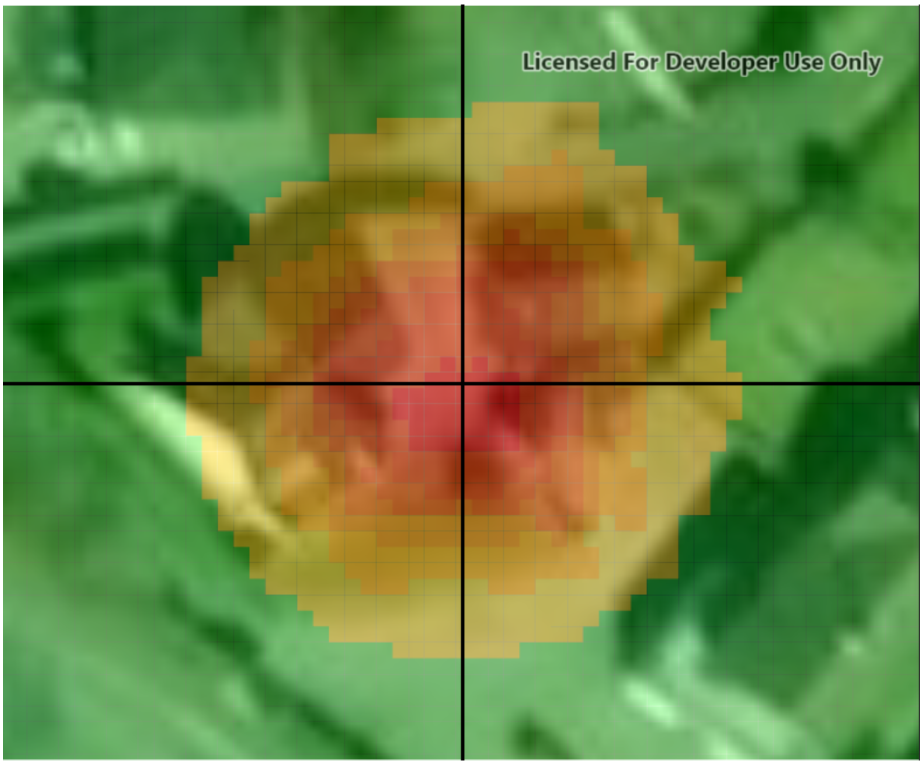

Figure 37: Capture of the results of a 5m/s speed flight with added inaccuracy caused by GPS device with a 0 degree avalanche transceiver orientation

Figure 38: Capture of the results of a 5m/s speed flight with added inaccuracy caused by GPS device with a 45 degree avalanche transceiver orientation

5.2 Noise

There are two different types of noise impacting the software and the generated results. There is a noise emitted by the drone system and its devices which causes a continuous noise with stable levels. Then there is external noise which is all types of noise that affects the sensors however not originating from the drone system. The external noise can affect the sensory inputs differently and with unpredictable timing. The continuous noise from the drone system is being handled by calibrating the software based on the hardware in use and defining a threshold. All signals with a strength below the threshold is categorized as noise and is therefore removed which causes the software to not use any of these signals. The solution is very lightweight and requires minimal processing power, but the downside of handling the noise this way is that all signals that are below this threshold gets removed together with the noise, as distinguishing the signals from noise is a difficult task. However, the method is justified since the signals removed are so weak their value to the results are minimal. The impact of the noise therefore shortens the range of the avalanche transceiver as the weaker signals gets masked behind the noise from the drone systems.

The external noise can cause a signal reading to appear greater than its true value which could cause a fake victim scenario if left unhandled. Fake victim scenario is due to the signal appearing greater than its true value, and therefore causing the UI to give the appearance of a victim when in reality there is none, also known as false positive. Due to the software design every single cell represents an area of 1x1 meter by default and the cell has a value that represents the signal strength in the area. When flying at or below the recommended operational flight speed, every cells value is based on at least 5 signal inputs. This means if external noise causes a signal reading to appear greater than it is, the cell will indeed be given a value higher than it truly is. However, since there is no victim there, the 4 or more inputs from the surrounding area will have lower values which impacts the fake signal and reduces the cells values greatly. This method therefore prevents the fake victim scenario given the drone is flown respectively. The concept is illustrated in figure 39 where the first image shows a single false positive input. The second image shows after receiving two surrounding inputs, and 3rd shows after 4 surrounding inputs has been received and the false positive has been handled. The figure is purely for illustrational purposes and the values are not real.

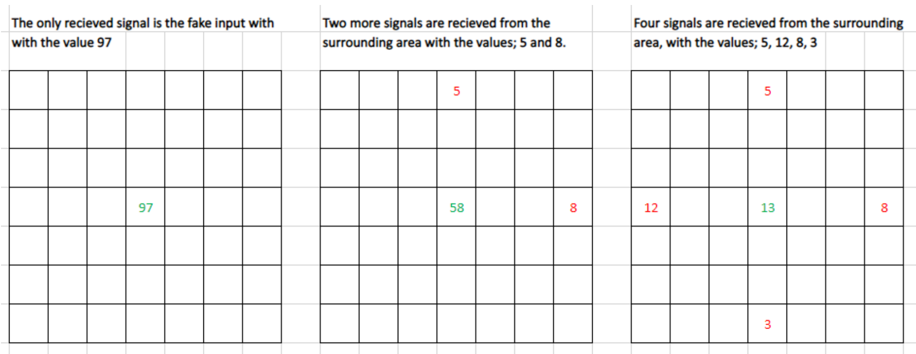

Figure 39: Concept illustration of the handling of false positive signal value

It is also possible that a signal input is a fake value, which happens when the avalanche transceiver fails to pick up the signal, causing the signal to appear lower than it actually is. If left unhandled this could cause it to appear as no victims, when there actually is a victim. This is prevented with the same method as above, but reversed. If flown at or below the recommended operational flight speed, every cell will be based on 5 or more signal inputs. If there is a fake signal causing a signal to be low, when it should be high, the surrounding signal inputs will be high and therefor greatly affecting the signal strength in the cells, and increase the values respectively such that it appears to be a victim there. The concept is illustrated in figure 40 where the first image shows a single fake input. The second image shows after receiving two surrounding inputs, and 3rd shows after 4 surrounding inputs has been received and the fake input has been handled. The figure is purely for illustrational purposes and the values are not real.

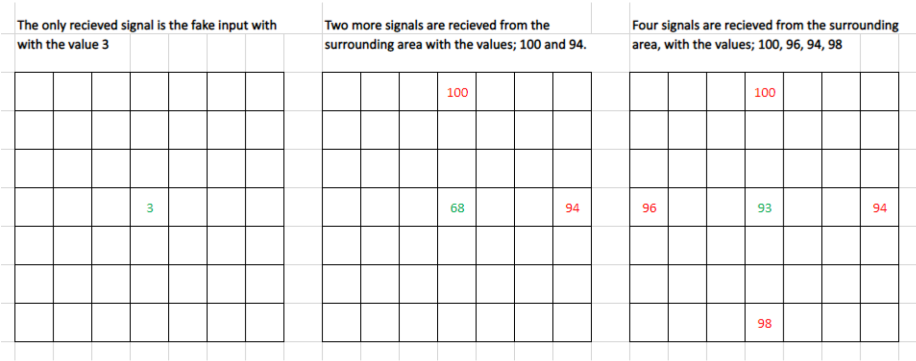

Figure 40: Concept illustration of the handling of fake signal value

5.3 Flight Speed

Recommended operational maximum flight speed is the maximum speed that is recommended for the pilot to fly when the goal is to search and locate victims. Flying above this speed limit will reduce the trustworthiness of the results greatly. Due to the avalanche transceiver design standards the transmission protocols transmit only once a second. Flying too fast will therefore cause a larger distance between the points where the software receives a signal strength. Adding the fact that anomalies can occur, such as explained above, the drone can potentially fly above a victim without detecting it due to the only signal from that area being corrupted. Figures 41 and 42 shows what can happen when flying too fast. Due to the avalanche transmission protocols explained in chapter 2.2 and the design of the solution, the maximum recommended flight speed is at 5m/s since flying faster has critical consequences. The figure is showing what could happen at a 6m/s flight speed.

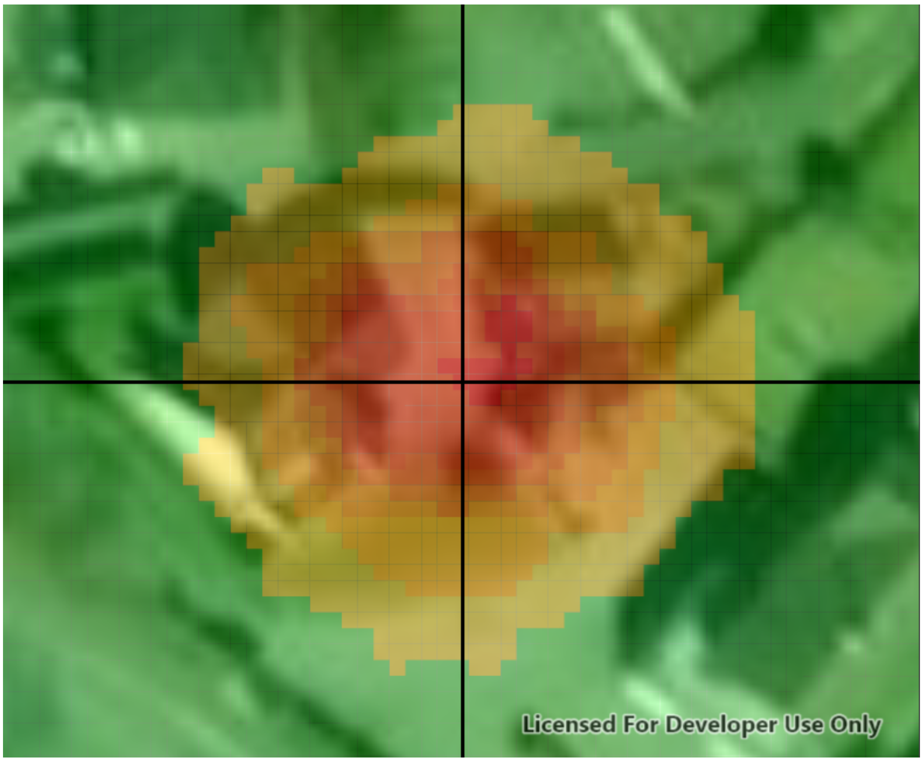

Figure 41: Capture of flight speed at 6m/s with a 0 degree device orientation

Figure 42: Capture of flight speed at 6m/s with a 45 degree device orientation

The software is trying to illustrate where the victim is, but the larger gaps between the signals causes the results to be untrustworthy and inaccurate compared to the victim's true location. In these two simulations, shown by figure 41 and 42 the anomalies with fake signals and false positives are not taken into account. Adding this will only cause even worse results and can therefore possibly cause the fake victim scenario. This is why the maximum recommended operational speed is set at 5m/s, because it is quite fast, but yet gives enough signals to create a trustworthy result. In chapter 4 it was mentioned that the simulations has a constant velocity which is unrealistic for a pilot to maintain, there will some variation in the velocity especially when the drone has to turn. However, as long as the drone does not go above the maximum recommended operational speed the variation in the velocity will not have a negative effect on the results. If the drone slows down because of a turn or any other reason, it will give more signal inputs in that area and only reinforce the results and therefore working under that assumption is justified.

5.4 Flight Pattern

The simulations flight pattern is based on a grid pattern shown by figure 43. This pattern is used for various reasons, such as its simple pattern which is easy to follow while piloting the drone. The pattern never crosses its own path which means it will never scan the very same position twice which in theory allows it to be optimal when it comes to area covered to time ratio.

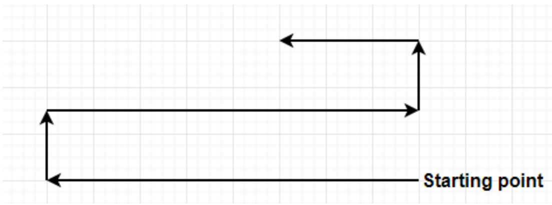

Figure 43: Illustrating grid flight pattern

The software is designed in a way such that the pilot is not restricted to any specific pattern even though grid pattern is used in the simulations. Therefore, the choice for both manual piloting and the design decisions made in the software allows the pilot to make adjustments. In theory the grid flight pattern is one of the better patterns since it never crosses its own path, which means it never scans the very same location multiple times. However, since avalanche rescue situations are unique, other patterns might be very relevant in a real-life scenario. This solution allows the pilot to change pattern and starting point based on judgement calls, both at the start, but also during the search. This means that the rescue team can make a judgement call based on their experience, witnesses and evidences such as tracks, to for example fly directly towards where they believe a victim is located and start a search pattern from that area. For example instead of doing grid search from bottom to top like shown in figure 43, the pilot can fly directly to the location they predict a victim and for instance do a circular search outwards like in figure 44, but whether it's faster or not depends entirely on the scenario.

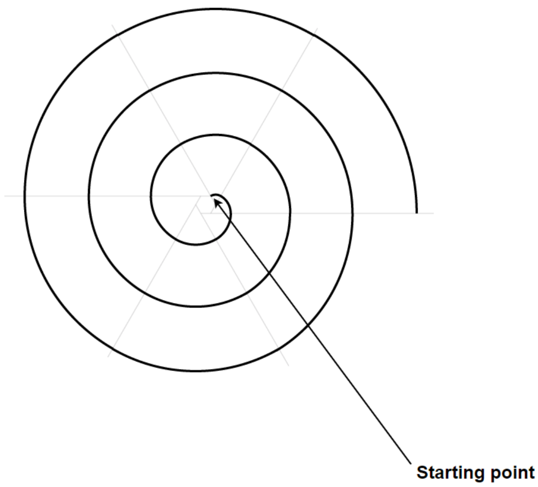

Figure 44: Illustrating circular flight pattern from pilot chosen center

5.5 Flight altitude

Currently the simulations are working under the assumptions mentioned in chapter 4 where one of the assumptions are as following; "the distance between the drone and the ground is constant". Realistically it can be expected that the drone flies in a stable height with minor variations. However, there will be unpredictable events that could cause a larger variation in the drone height such as weather and nature, like trees and rocks. The data received from the GPS device mounted on the drone gives data about the drone's altitude, but as this is measured as the drone's height with respect to the sea levels and therefore does not describe the height from the drone to the surface level. Even if data was pulled from maps the avalanche would have affected the surface levels and therefore it becomes unpredictable with the current available information. At current state of the software it will only be able to handle minor deviations in height, and larger deviation could potentially impact the results greatly. If the drone flies up and down again within a small period of time, only a few signals would be affected and as these signals would appear lower than they should be since the increased distance from the avalanche transceiver causes the signal to be weaker. This would be similar to the fake signal scenario explained above and would be handled the very same way. However, if the drone were to fly up and keep the new height for a longer period of time, the number of signals affected would be greater and therefore potentially cause huge negative effects on the results if the drone were to keep this new height while being above victim. If the drone quickly flies up and down, as mentioned it would be handled similarly to the fake signal scenario. Which means the surrounding signals would thin down the impact from this anomaly shown by figure 45, where the green input is the location of the altitude change and the red inputs are the surrounding inputs.

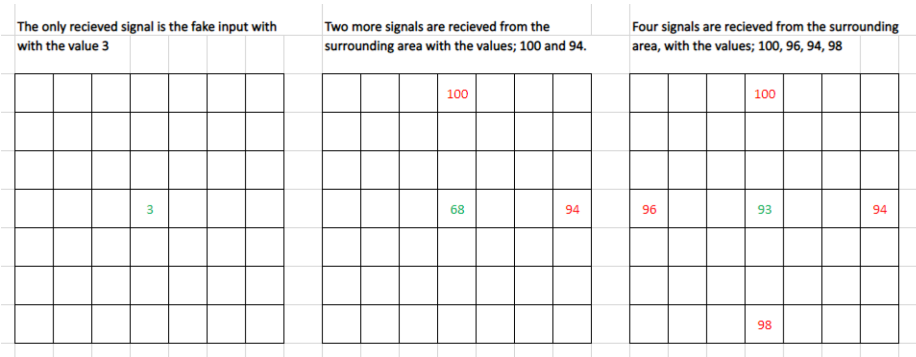

Figure 45: Concept illustration of the handling of short period altitude variation

However, if the drone maintains this new altitude for a longer period of time. There surrounding signals which would reduce the effect of the anomalies, be anomalies as well. Thus, if the altitude is increased and maintained, it could cause the software to be unable to detect the victim, as shown by figure 46. The green input is the first input and the red ones are surrounding inputs which are used to stabilize it.

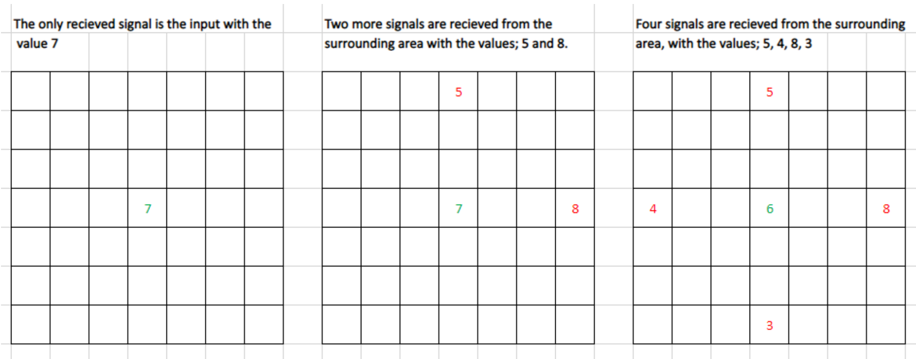

Figure 46: Concept illustration of the longer period altitude variation

The problem occurs because the surrounding inputs are used to handle anomalies such as fake inputs, false positive and height variation. However, as seen on the figure all the surrounding inputs are low values since the drone is flying too high in this area for a longer period of time. Causing a larger distance between the victim and the drone which causes the signal's strength to be low. If the victim was laying below the drone, and the drone was flying at a reasonable height the results would show the victim's location. The problem is that the area the drone is operating at a too high altitude and maintaining the altitude in the surrounding area cause all the surrounding inputs to also appearing as low signal strengths when in reality they should be higher. The chances that a pilot is flying very high over a longer period of time while being above the victim is relatively low, but remains a possibility and should therefore be handled.

6 Conclusion

The purpose of the research was to find out how avalanche transceivers could be located via drone. The solution needs to establish a visual representation of the potential locations of the victims caught by the avalanche. While the results generated by the solution has to be be accurate enough, such that the results are trustworthy and therefore usable. The results of the various simulations shows that the solutions precautions against different sources of inaccuracy allows the software to generate the mapping with trustworthy data. There are multiple scenarios, anomalies and other sources that causes inaccuracy of different magnitudes. The main source of inaccuracy is the inaccuracy caused by the GPS device that is mounted on the drone, since the result mapping is based on GPS. It can be expected that the GPS device causes up to 3.5 meters and potentially more due to its inaccuracy, but also the solutions design. However, the solutions design is basing every cells value on at least 5 different inputs which reduces the impact of the GPS devices inaccuracy greatly. The solution also has an inaccuracy caused by converting GPS coordinates to a grid and back to GPS coordinates after the processing of the data. This inaccuracy has the potential to reach up to 0.7 meters, but its impact is also reduced because every cells value is based on 5 different inputs. The method of basing every value on multiple inputs also counters the impacts on the results caused by fake signals and false positives. The inaccuracy of applying linear interpolation on GPS coordinates, which are not linear, has been calculated in chapter 3.2 to be up to $3.47 \cdot 10^{-5}$ meters which causes its impact on the results to be minimal. As seen by the results, the accuracy of the results is mainly restricted by the GPS devices inaccuracy. However, the main purpose of the solution is to reduce the time required to locate the victims in an avalanche. The solution allows the rescue team to move directly towards the estimated location and start a fine search in that area. This allows the rescue team to save time since they do not have to go through the signal search and course search processes. The solution is designed in such a way that the pilot is not limited or restricted by any specific flight pattern, this allows the team to choose which area to search first with the drone based on their experience, witnesses or for example ski tracks. The two restrictions the pilot has are the flight speed and height variations. The flight speed is restricted mainly because of the avalanche transceiver standards, which causes them to emit about once per second, for battery saving purposes. The altitude variation restriction is caused by the fact that the system is currently unable to tell how far above the ground or victim the drone is, which means it's unable to take into account whether the signal is low because of high altitude or because there is not a victim located below. Further research is required to figure out how to handle the larger flight altitude variations impact on the mapped results. If the pilot flies at a speed respecting the recommended maximum operational speed of 5m/s and does not fly too high for a longer period of time the results generated will be trustworthy and reduce the search area to a few square meters. This allows the rescue team to save time and potentially increases the victim's survival chances.

7 Further work

As mentioned in section 6, the main limitation is the requirement of flying with a stable altitude. Further work should therefore be focused on finding out how to handle altitude changes with lesser impacts on the results. The GPS data includes altitude, but this is not sufficient data to predict the altitude between the surface and the drone since the altitude given by the GPS is based on sea level, while the surface is uneven and unpredictably so after an avalanche. The solution has an integrated weight system where the weight describes how much an input should affect a value represented in a cell. The weight is based on a 2 dimensional plane by x,y which is used to lower the weight the further away from the inputs location. This specific drone has built-in sensors that allows to measure range to the ground precisely within 11 meters and detect ground within 11-20 meters[6]. Further work could attempt to retrieve this data from the drone itself to test if it's accurate enough. However, this will restrict the solution to a specific drone type. Another way could be trying to mount a range indicator on the drone to get data on how far above surface level the drone currently is, and change the weight to a 3-dimensional system using the range indicator to represent z axis, the height. This allows the system to reduce the weight of a signal if it for example were retrieved while the drone was 20 meters above the surface.

References

[1] A. Alderson. “Avalanche safety [Sports Tech]”. In: *Engineering Technology* 9.11 (2014), pp. 84–85.

[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] Guilhem Bryant. “An autonomous multi-UAV system for avalanche search”. In: (2019).

[4] *CodeLab*. URL: https://www.codelab.no/.

[5] Benjamin D. Dickensheets. “AvaDrone: An Autonomous Drone for Avalanche Victim Recovery.” In: (2015).

[6] *DJI Mavic Pro 2*. URL: https://www.dji.com/se/mavic-2/info#specs.

[7] “ETSI EN 300 718”. In: (2017).

[8] Pascal Haegeli et al. “Comparison of avalanche survival patterns in Canada and Switzerland”. In: *CMAJ* 183.7 (2011), pp. 789–795. ISSN: 0820-3946. DOI: 10.1503/cmaj.101435. eprint: https://www.cmaj.ca/content/183/7/789.full.pdf. URL: https://www.cmaj.ca/content/183/7/789.

[9] IBM. *Geographic coordinate system*. URL: https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.spatial.doc/ids_spat_407.htm. (accessed: 11.11.2019).

[10] Norwegian Geotechnical Institute. *Ulykker med død*. URL: https://www.ngi.no/Tjenester/Fagekspertise/Snoeskred/snoskred.no2/Ulykker-med-doed. (accessed: 11.11.2019).

[11] *Mammut Barryvox S*. URL: https://static.mammut.com/file/2710-00140_man_en_3-2_Barryvox_S_Extended_Reference_Guide.pdf.

[12] Luc Oth Manuel Grauwiler. “Fully Autonomous Search for Avalanche Victims Using an MAV”. In: (2010).

[13] *NMEA GPGGA*. URL: http://aprs.gids.nl/nmea/#gga.

[14] Mario Silvagni et al. “Multipurpose UAV for search and rescue operations in mountain avalanche events”. In: *Geomatics, Natural Hazards and Risk* 8.1 (2017), pp. 18–33.

[15] Vyacheslav Tuzlukov. *Signal Processing Noise*. CRC Press, 2018. ISBN: 9781420041118.

[16] *What I Learned in Avalanche School*. URL: https://www.nytimes.com/2019/12/31/magazine/avalanche-school-heidi-julavits.html.

[17] Michael G. Wing, Aaron Eklund, and Loren D. Kellogg. “Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability”. In: *Journal of Forestry* 103.4 (June 2005), pp. 169–173. ISSN: 0022-1201. DOI: 10.1093/jof/103.4.169. eprint: https://academic.oup.com/jof/article-pdf/103/4/169/25133856/jof0169.pdf. URL: https://doi.org/10.1093/jof/103.4.169.

Sourcecode available at: https://github.com/Kevinar294/master2020

Backup link: https://1drv.ms/u/s!Aj1wkZDT-KnllXJh1KMKe_FeHGSn?e=Icxgpb