Delft University of Technology

Thesis Report

Spoofing detection in a loosely coupled GNSS and IMU system via Synthetic Arrays.

Author:
Kostadin G. Biserkov

Supervisors:
prof.dr.ir. A.J. van der Veen
dr. Faruk Uysal
dr. ir. Andre Bos

August 20, 2019

This page is left blank intentionally.

1 Abstract

With the ever-expanding need for accuracy in the world of navigation, Global Navigation Satellite Systems(GNSS) such as GPS and Galileo have become the primary option around the world. As such, the potential damage that can be caused by malicious tampering with the receivers continues to grow. One such threat, known as Spoofing, comprises of transmission of altered GNSS signals to a target receiver(s). This process leads to false positional and/or time data on the receiver's end. Spoofing has been a topic of discussion for roughly two decades. With many theoretical approaches to its detection, this work focuses on the Angle of Arrival technique via the construction of a synthetic array from a single moving element, aided by positional information provided by Inertial Measurement Unit (IMU). This method relies on the periodic nature of the L1 signal, and focuses primarily on the case of GPS as an example. Using simulation and sample data, the possibility and limitations of constructing virtual antenna arrays is explored. It is shown that despite being viable for low number of sources during the simulation, the complexity of signal propagation within the real world implementation of GNSS system renders this technique inoperable in its first iteration.

Contents

2 Rationale

Global Navigation Satellite Systems (GNSS) are a widely used technology that has become part of user electronics, civil applications, military systems and much more. As an example, GNSS is used to provide Position, Velocity and Time (PVT) information with high accuracy in many sectors, such as telecommunication networks, transportation systems and electrical grids[1]. This reliance has caused multiple studies into the possible effects that malicious attacks against GNSS systems within such infrastructures can lead to [2][3][4][5][6]. Malicious attacks can vary from denial of satellite service (jamming) to attempt of deceiving the GNSS receiver into providing false navigational data. Such attacks are more widely known as spoofing attacks.

2.1 Global Positioning System (GPS)

In this work, the focus will be on GPS - a system that has been operational for almost 4 decades. It is a GNSS constellation built and operated by the United States government. It consists of at least 24 operational satellites at any time, spread in 6 Medium Earth Orbits (MEO) of approximately 22200km altitude. As Fig.1 below shows, this spread of the satellites allows for a receiver on the surface of the Earth to have consistent signal reception from multiple satellites.

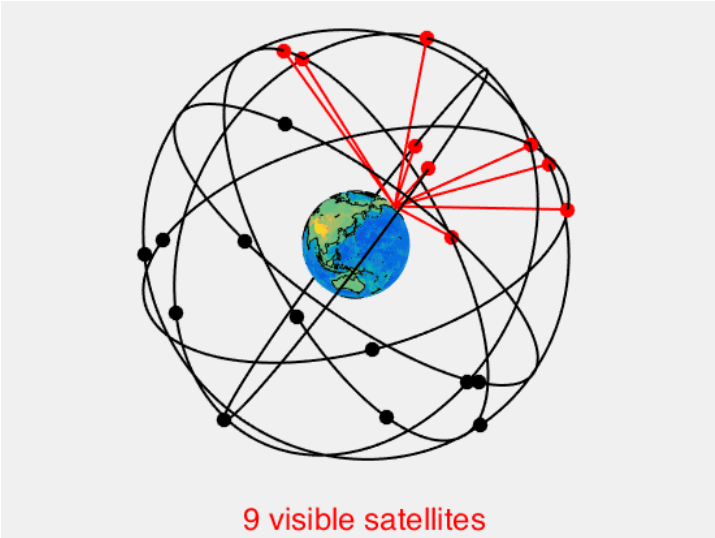

Figure 1: Example of a GPS satellite and receiver configuration.

Each of the Space Vehicles (SVs) transmits a signal that is open to any user on the L1 band (1575.42MHz). This signal comprises of three main components - carrier signal (sine wave at 1575.42MHz), a Pseudorandom noise (PRN) code, and a navigational data part (Fig. 2). The carrier wave is shared by all satellites, while unique PRN codes are assigned to each SV. The PRN sequences used in GPS (also known as Coarse Aquisition codes, or C/A) are 1023 bit unique codes with a rate of 1.023 MHz. They are part of a family of codes known as Gold Codes [7] - periodic codes, which are deterministic by nature, but also have noise-like properties. The three most important properties of the PRN codes are:

- Despite appearing as random, the sequences are entirely deterministic. By following the same instructions, an exact replica of each PRN code can be produced by any receiver, as well as the satellite to which the code has been assigned.
- Correlation is only present when the lag is zero: Every PRN code is almost uncorrelated with itself, unless for the situation where both sequences have no lag between them. By utilizing this property, a receiver can always align (in time) a locally generated replica of a PRN sequence with the one transmitted from the SV.
- There is almost no cross correlation between two different PRN codes. As the signal received by a single antenna on the surface of the Earth is the sum of signal from multiple SVs $s(t) = s_1(t) + s_2(t) + ... + s_n(t)$, this PRN property allows the receiver to isolate the signals from each satellite. Here $s(t)$ is the signal received by the receiver's antenna, and $s_i(t)$ represents the signal transmitted by satellite i.

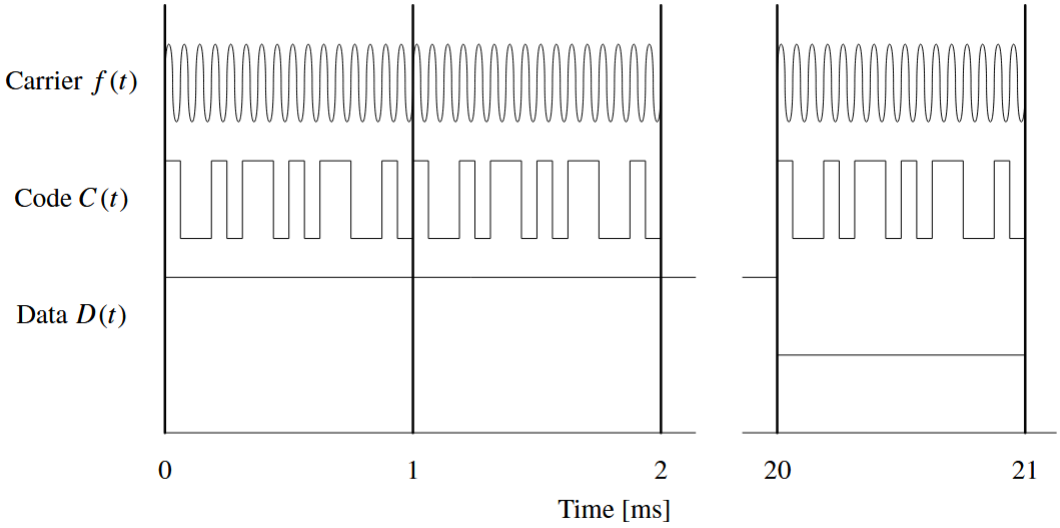

Figure 2: Structure of the L1 GPS signal[7]

2.2 Spoofing

Spoofing is in its essence a malicious interference of communication channels with the goal of "feeding" the target system with data that is under the control of the attacker. In the case of GNSS, such an attack could potentially result in partial or complete "hijacking" of the navigational solution of the target receiver, without the knowledge of the system or the end user.
The fact that the navigational algorithm and signal structure of GPS are both widely known and well documented, helped tremendously in the adaptation of the technology. However, the same facts also allow for the creation of very accurate simulations, capable of generating precisely what the received signal $s(t)$ for any location and time would be. A system could then broadcast the generated simulation via an antenna towards a target receiver. If adjusted correctly, it would be impossible for any typical receiver to identify at any point during its operation whether the source of the information is genuine or not.
There are multiple types of spoofing attacks, with various degrees of complexity. The most basic approach is to use a completely offline simulation, and transmit the generated malicious signal at much signal higher power than what would usually be observed by a receiver from the genuine signals. The way a receiver would usually determine if a satellite is currently visible is by generating a local copy of each C/A code and scanning through all possible time delays if the spreading code. This is done by using the correlation properties mentioned earlier to determine **a)** which satellites are currently visible and **b)** what is the time offset of the PRN code. As this is done via correlation, the typical approach is to find the precise time offset where the correlation result is maximum. As this is also bound to the overall signal power, a correct signal structure at higher power levels will cause a receiver to start tracking the spoofer's signal, instead of the genuine one.
This means that during its normal operation cycles, the targeted receiver will eventually determine that the best quality signal available to it is the spoofed signals, and will result into presenting the user with the falsified positional information. While extremely simple and cheap to create, such spoofers (also known as asynchronous spoofing devices) can potentially be used to interfere with many receivers in a specific area at the same time.
A second attack scenario [7] (Fig. 3) can be achieved by expanding the spoofer with a GNSS receiver of its own. By collecting genuine GPS signal information, the simulator can be first synchronized with the genuine constellation signal. Once the synchronization has been achieved, minor alterations to the signal can be made, and the attack can proceed similar to the previous case. In this scenario, however, the required signal power can be controlled much more precisely (as the power of the incoming genuine signal is known), meaning the entire process of "hijacking" the lock of the receiver becomes a lot more subtle and difficult to detect.

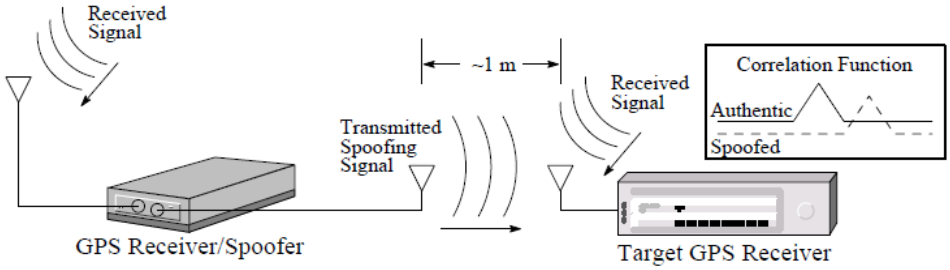

Figure 3: Spoofing attack scenario [7].

2.3 Detection approaches

Having covered the basics of GPS spoofers, the next step is to understand what can a user do to detect and potentially prevent spoofing attacks from succeeding. This question has been the subject of research projects for more than a decade, and as such there have been many approaches already explored. Given below are the general directions that have been considered previously.

2.3.1 Automatic Gain Control

One of the approaches that has been considered in many publications is the concept of using the Automatic Gain Control (AGC) functionality that is already applied to the majority of multi-bit receivers. To understand its application in spoofing detection, it is first needed to understand better the relevant properties of the signal. The power level of a GPS signal, received by a Right Hand Circular Polarized (RHCP) antenna is well below the thermal noise floor - in fact, according to the GPS Interface Specification [8], the guaranteed minimum signal power is -160 dbW. To put this into perspective, the thermal noise floor for the GPS C/A code (L1 band, carrier frequency of 1575.42 MHz, bandwidth of approx. 2 MHz) can be calculated using

$$P_{TN} = kTB$$

where k is Boltzmann's constant ($1.3810^{-23} J/K$), T is the absolute temperature in Kelvin, and B is the bandwidth of the given signal. Using the characteristics of a standard GPS L1 receiver, the thermal noise floor can be calculated to be $-$ 140.97 dbW, roughly 20 dbW higher than the guaranteed minimum signal power [7]. However, one must also take into consideration the additive effects of noise generated at the front-end of the receiver. This requires the receiver to optimize the gain of the front-end to match the levels of the resulting signal to the input levels of the Analog-to-Digital Converter (ADC). The two driving reasons are the need to allow for the receiver to adjust to varying front-end gain levels, but more importantly to adjust the gain to the presence of RF interference (RFI). Most AGC algorithms adjust the gain to ensure that the captured signal follows an expected Gaussian distribution. This could allow the user to closely monitor the behaviour the AGC adjustments over time, and gain a better understanding of the relative evolution of the power levels. The spoofing detection based on AGC is based on the fact that in order to "hijack" the victim's receiver, more power must be injected into the signal. In such a situation, the AGC values will adjust for the change in the total signal power (combination of genuine and malicious signals). Akos et al. (2012) [9] show experimentally the change in the AGC values as their test platform approaches a spoofer, with the changes becoming obvious well before the position solution becomes affected by the spoofer signal. However, this approach is not a universal solution - while it can be very effective for spoofers that emit signals with power significantly higher than the genuine GPS signal, with minimum computational complexity and no need for introduction of additional system components, it has two major disadvantages. As its purely based on evaluation of signal power, it becomes difficult to distinguish between intentional spoofing, or any other source of RFI. In addition, for more complex spoofing attacks (e.g., a system that matches the power of the genuine signal, then gradually increases the spoofed signal power in order to drift the lock of the receiver away from the genuine signal) the approach can fail. As explained in more details by Hagarty et al. (2018) [10], in order for AGC levels to increase beyond what is typically observed in normal situations, the power of the added signal must be at least 15 dB stronger that the true GPS C/A levels (for attack scenarios with a synchronized simulator).

2.3.2 Signal Quality Monitoring

Another approach for spoofing detection relies on the operation and results of the so called Code and Carrier Signal tracking. The main purpose of this stage is to fine-tune the rough estimations for the carrier frequency,

extract the navigational information from specific satellite(s), and also give insight regarding the pseudoranges. This is a fundamental stage in the processing of GNSS information, and as such it has been widely documented and explained (a detailed breakdown can be found in "A Software Defined GPS and Galileo receiver" by K. Borre, Chapter 7) [7]. While there are several different implementations, there are two basic operations. The first is to determine the Doppler shift with respect to the carrier frequency, caused by the moving satellite. The implementation of this is usually referred to as a Phase Lock Loop (PLL). The second component is to find the relative offset between the local generated version of the PRN sequence, corresponding to a specific satellite. This process is known as a Delay Lock Loop (DLL), tracking an incoming signal means that the receiver locates the correct values for the code phase and Doppler shifts that maximize the Cross Ambiguity Function (CAF) - a 2D correlation function.
The CAF is a fundamental component of GNSS signal processing, and as such, any significant distortion in the CAF can lead to de-synchronization between the lock of the locally generated version of the PRN and the one of the incoming signal. A common effect that can lead to such distortions is multipath. Studies, such as Cavaleri et al. (2010) [16] investigate the effects of intermediate spoofing (synchronous spoofing attacks) on the CAF. Their findings show that the effects of such spoofing attacks are very similar to those of strong multipath events. In the field of GNSS, such techniques of evaluation of the signal lock state are also known as Signal Quality Monitoring (SQM), and previous works have provided ways of classifying different distortions and determining their causes. Different SQM metrics have been proposed to help in these processes. For example, the Delta metric allows the identification of asymmetric correlation peaks. This is usually done by looking at the normalized difference between the early and late correlation outputs. Such asymmetric peaks can be caused during the event of "hijacking" the signal lock from the genuine to the malicious signal, meaning that such metrics could provide a trigger for intermediate spoofing attacks. It must be noted that there are other SQM metrics (Ratio Tests, Cavaleri (2010) [16]), which could provide a basis for more complex spoofing detection techniques that rely on multiple-layered SQM. However, a major drawback in the application of SQM is the similarities of the effects of repeater spoofers and strong multipathing.

2.3.3 GNNS Coupled systems

While GNSS is a standalone system, meaning that it does not require input from an additional navigational source to determine its location, there has been a lot of work on using additional positional information to enhance the location solutions. For example, Differential GPS (DGPS) makes use of reference receivers, with exact known locations, to calculate corrections models, and transmit them to other DGPS-enabled receivers in the vicinity. Another approach to the problem of increasing the reliability and accuracy is to use other sensor systems, available on the same platform as the GNSS receiver, and estimate the precise position from the combination of the two sources. For example, Inertial Measurement Units (IMU) are standard equipment for many aircraft. By continuously tracking the motion information of the platform, it is possible to calculate the position of the aircraft without the use of GNSS (such systems are called Inertial Navigation Systems, or INS). However, it is also possible to combine the outputs from GNSS and INS (without the need for close coupling). One implementation involves the use of an Extended Kalman Filter - a non-linear version of the standard Kalman filter [17]. This estimator provides an algorithm that can predictively estimate the next position based on the measurement history, and adjust the model once sensor information about the current position is available. In such a system, an internal metric known as innovation provides the user information regarding the difference between the predicted and the measured positions. In steady-state operation, the innovation's expected value becomes 0, and its covariance becomes known [18]. These properties are often used for fault detection. However, according to Liu et al, both synchronous and asynchronous spoofing attacks lead to a change in the statistics of the innovation metric. This can allow for the use of statistic tests to find the changes in the distribution of the innovation, and use it as a detection mechanism for spoofing attacks. The major benefit that IMUs (or other sources of motion data) is their independence of RF transmissions - the remote spoofer cannot influence the data coming from the IMU without physically tampering with the sensors, a scenario that is not in the scope of this study. Even after taking into account the known disadvantages of IMUs (build up of sensor drift with time), studies such as Curran et al. (2017) [19] show that even low-cost, uncalibrated IMUs can contribute to the building of a spoofing detector.

2.3.4 Angle of Arrival

Another family of spoofing detection algorithms are based on the spatial understanding of the genuine versus malicious GPS signals. Known as the Angle-of-Arrival (AoA) approach, it is based on monitoring the approximate direction from which each signal component originates. In the genuine case a GNSS constellation relies on the fact that every receiver has sight of multiple space vehicles (SVs) at any given point of time. These vehicles

are spread across the visible part of the sky, and are located at different angles with respect to the receiver (azimuth and elevation). In addition, these directions will change over time (as the SVs move, and gradually disappear from the sight of the receiver, to be replaced by other SVs), and (in most cases) the elevation angles will be significantly higher than the horizon. In the case of spoofer signals, the spatial profile will differ significantly. In the case of a synchronized spoofer, the adopted assumption is that the system will receive the genuine signal, create a copy of the signal with slight (malicious) adjustments, and re-transmit the signal from a single antenna towards the victim's receiver. This would mean that the victim will receive all signal components (from all visible SVs) from a single direction, which is in direct contradiction with the spatial distribution of GNSS constellations. In addition, in most cases the spoofer will appear to be at or below the horizon (elevation angles), which is once again impossible for a genuine satellite signal.
However, the process of determining the angle of arrival of the different GPS signals from the visible satellites is not trivial. The fundamental technique of calculating the direction of arrival of a radio signal is to capture the same incoming signal with multiple antennas, placed in a predetermined geometry. As the distance between the source and each of the antennas will be slightly different, the received copies of the signals will be slightly offset in time (as the signal needs to travel different distances with constant speed). In repetitive signals (such as the C/A code for example), this is also known as a Code Shift. By analysing all shifts, it becomes possible to calculate the direction of the incident signal. This would allow the enabled receivers to identify the direction of a spoofer/RFI source/jammer, and could also be combined with spatial nulling techniques in order to filter out their effects. This technique has been widely used in many other fields (satellite communications, radar technology, radio astronomy, etc.), and has also been considered by many as a promising solution to the GNSS Spoofing problem [11][12][13][14][15]. Unfortunately, most GNSS receivers on the market employ a single antenna, making it impossible to make use of this technique without making significant hardware changes. In addition, the system is not impervious to all possible spoofing attacks. More complicated spoofers could employ a distributed network of mobile platforms (for example drones), each transmitting only the modified signal that corresponds to one SV. This would mean that the spatial profile of the spoofed signal will replicate that of a genuine GNSS constellation. It is clear that the computational and implementation complexities of such a spoofing scenario is magnitudes higher than most, but it remains a plausible case.

2.3.5 Control Reception Pattern Antenna

Controlled Reception Pattern Antenna (CRPA) arrays is a technique that allows to suppress signals coming from specific directions. Similar to the technique described in the Angle of Arrival problem, a CRPA-enabled system uses an array of spatially-distributed antennas to track a fixed number of channels. Each channel focuses on a single satellite signal, and uses techniques such as Minimum Variance Distortionless Response (MVDR) to estimate steering vectors for each satellite signal or nulling vectors for when a jammer/spoofer is detected. If done dynamically, the resulting antenna pattern (combining all beams and nulls) can adjust as the different sources of original and/or malicious signals change their positions or profiles [21]. To visualize the combination of beamforming and nulling technique, Fig. 4 [22] below shows three separate responses of the CRPA system in different use cases. The first case (left) has a single jammer source, resulting in a null present in the reception response of the antenna array. The second case (middle) shows the reception pattern where three jammers are present at different locations. The last case shows the results of the beamforming when a satellite signal is present - the antenna beam pattern is maximized in the direction corresponding to the estimated signal source and minimized elsewhere.

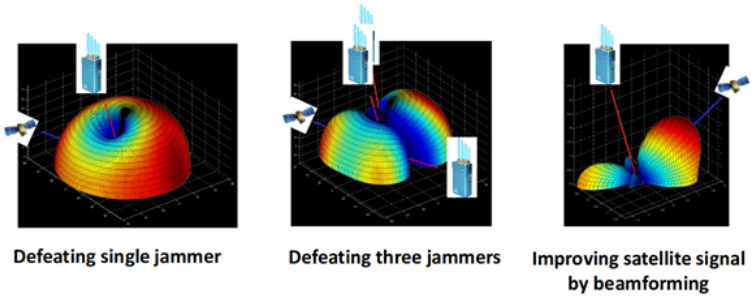

Figure 4: CRPA antenna pattern in different use cases [22].

This technique has already found application in military applications, but has also reached the Civilian

market. Products such as the Landshield (Raytheon UK) provide direction-finding for spoofers and jammers for GPS and Galileo open codes. Other products offering similar functionality for civilian applications include the 'Helium' antenna. The downside of this approach is the requirement for replacement/addition of hardware (front-end) to the system, which may not always be possible.

2.3.6 Doppler monitoring

Focusing on Asynchronous spoofing attacks, it becomes apparent that there are some detection techniques that are not possible in the case of a Synchronous spoofing scenarios. This is because of the inherent properties of the signal emitted in the case of an Asynchronous attack - since the spoofing device cannot take into account any carrier frequency change (due to Doppler shift), the signal must simply "overpower" the genuine signal, in order to remove the lock of the receiver to the original GPS signal. As discussed earlier, this allows easy detection using AGC, but it also provides an opportunity to detect the sudden change in the estimated Doppler shift, caused by the change in lock [23]. If the receiver is capable of storing short-term history of the estimated Doppler shifts, such discontinuities should become easy to measure.

2.4 Problem Statement

After assessing the different methods of detecting a spoofer, a decision was taken to look further into the Angle of Arrival approaches. As briefly mentioned earlier, in order to extract the information regarding the positions of each source, the system must be able to sample "spatially". The fundamental way of providing this information to a system is by using a number of antennas, with a known geometry. This, however, increases the complexity of the overall receiver, as more hardware needs to be installed. In addition, in order to enable this on an existing system, one must modify significantly the available hardware. The physical installation of multiple antennas, with a given distance between each pair, limits the applicability of the system. For example, portable devices and drones often do not have sufficient size to allow for such a modification. The problem at hand becomes whether it is possible to estimate the Angle of Arrival of sources by only having one standard antenna element. A possible solution to the problem is to employ a technique known as Synthetic Array. In its essence, the approach relies on tracking the motion of a single antenna, and using its change in position to provide the spatial sampling. As the trajectory of the moving antenna needs to be precisely measured during the process, it becomes apparent that an additional system, capable of providing motion and position information (outside of GNSS)needs to be included. A widely used approach is the use of Inertial Measurement Units (IMUs). While such devices can provide information about orientation (angular) and motion (acceleration along multiple axis), IMUs are generally known to suffer from bias and sensor drift. Both of these approaches are discussed in further details in Section 3.

2.4.1 Research Question

How can a GNSS receiver employ estimation of Angle of Arrival to detect spoofing signals, ensuring that only one physical antenna is used in its Front End? How does such an approach compare to using a conventional physical antenna array?

2.4.2 Secondary Research Question

What are the uncertainties of using an Inertial Measurement Unit (IMU) as the source of positional information for a Synthetic Array?

3 Signal model

The following chapter focuses on the derivation of the theoretical signal model of both conventional and synthetic antenna arrays. This is necessary to establish the similarities between the two methods, and can serve as an initial verification on the applicability of synthetic arrays in this use-case.

3.1 Antenna arrays

3.1.1 Phased arrays

A conventional Antenna array is made out of M antennas, positioned at a known geometry. Fig. 5 below shows an example of such a system. For simplicity, the given example assumes:

- Uniform Linear Array (ULA). All elements are at equal distances, along a single line.
- The problem is for the 2 dimensional case. All sources are positioned on the same plane (including the line of antenna elements).
- All sources are in such distances that all antennas are in the Far Field, or in other words - the incoming wave front is a plane (planar waves).

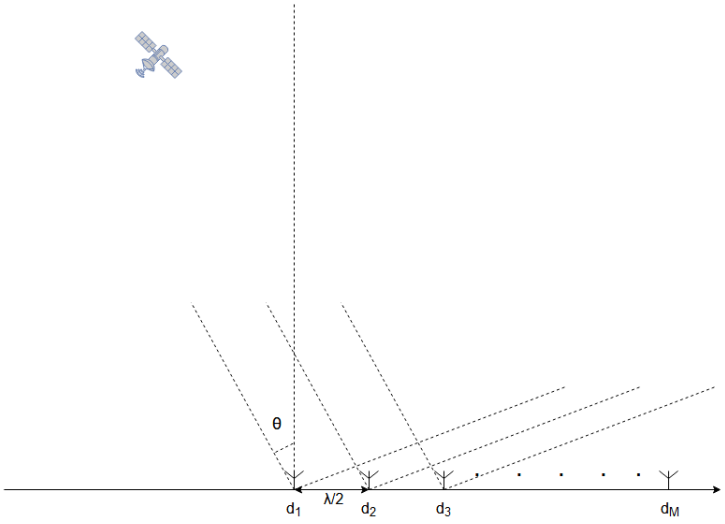

Figure 5: A Phased Array setup with a single source.

If only one source is present, the signal sampled by each antenna becomes:

$$x_i(t) = \alpha_i s(t - \tau_i) + n_i(t) \tag{1}$$

where $x_i(t)$ is the sampled signal at antenna $i = 1, 2, \ldots, M$, $s(t)$ is the signal transmitted by the source, τ_i is the delay at antenna i due to the physical propagation of the signal from the source to the antenna, α_i is the antenna gain, and $n_i(t)$ is noise. In the case of a GNSS constellation (for example GPS), each antenna would receive signals transmitted by multiple sources (SVs) (Fig. 6).

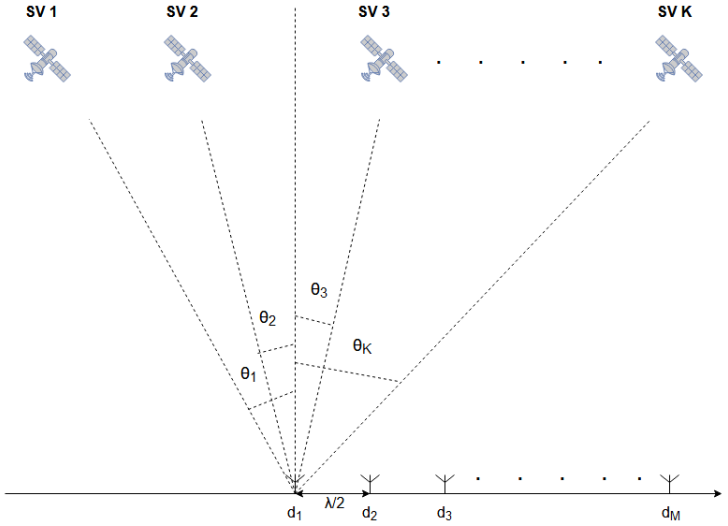

Figure 6: A Phased Array setup with multiple sources (GNSS).

If there are currently K satellites visible to the receiver, equation (1) becomes:

$$x_i(t) = \sum_{j=1}^{K} \alpha_i s_j(t - \tau_{i,j}) + n_i(t) \tag{2}$$

Where $s_j(t)$ is the signal transmitted by source $j = 1, 2, \ldots, K$, $\tau_{i,j}$ is the delay at antenna i due to the physical propagation of the signal from SV j to the antenna, and $n_i(t)$ is noise. In this case, the data matrix (the signals received by all antennas) becomes:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{K} \alpha_i s_j(t - \tau_{1,j}) + n_1(t) \\ \sum_{j=1}^{K} \alpha_i s_j(t - \tau_{2,j}) + n_2(t) \\ \vdots \\ \sum_{j=1}^{K} \alpha_i s_j(t - \tau_{M,j}) + n_M(t) \end{bmatrix} \tag{3}$$

Taking a step back and assuming only one source is present (e.g., SV j), as well as the idealistic "no noise" scenario, the data matrix becomes (for simplification, the gain α_i is omitted):

$$\mathbf{x}(t) = \begin{bmatrix} s_j(t - \tau_1) \\ s_j(t - \tau_2) \\ \vdots \\ s_j(t - \tau_M) \end{bmatrix} \tag{4}$$

From (4) it can be seen that given a single source, the only difference between each element is introduced by the different delays $\tau_{i,j}$, caused by the different positions of each antenna. Without breaking generality, let us assume that the position of antenna 1 (reference antenna) is at the origin of the coordinate system. This leads to the conclusion that each $\tau_{i,j}$ can be written in the form:

$$\tau_i = \tau_1 + \Delta\tau_i \tag{5}$$

With $\Delta\tau_{i,j}$ being the time difference between the delay at antenna i and the reference antenna, for the signal coming from source j. Given that the distance between each antenna is $\frac{\lambda}{2}$, following Fig. 5, $\Delta\tau_{i,j}$ is given by:

$$\Delta\tau_i = \frac{\Delta d_i}{c} = \frac{\sin(\theta) d_i}{c} = \frac{\sin(\theta)(\frac{\lambda}{2})(i-1)}{c} \tag{6}$$

Once the delay difference is known, it can also be represented as a phase shift imposed on the incoming signal. The data matrix (4) becomes:

$$\mathbf{x}(t) = \begin{bmatrix} s_j(t - \tau_{1,j}) \\ s_j(t - \tau_{1,j} - \Delta\tau_{2,j}) \\ \vdots \\ s_j(t - \tau_{1,j} - \Delta\tau_{M,j}) \end{bmatrix} = \begin{bmatrix} s_j(t - \tau_{1,j}) \\ s_j(t - \tau_{1,j}) e^{-j2\pi f_c \Delta\tau_{2,j}} \\ \vdots \\ s_j(t - \tau_{1,j}) e^{-j2\pi f_c \Delta\tau_{M,j}} \end{bmatrix} = \begin{bmatrix} s_j(t - \tau_{1,j}) \\ s_j(t - \tau_{1,j}) e^{-j2\pi \frac{sin(\theta_j)}{2}} \\ \vdots \\ s_j(t - \tau_{1,j}) e^{-j2\pi \frac{sin(\theta_j)(M-1)}{2}} \end{bmatrix} \tag{7}$$

Equation (7) can be further simplified to:

$$\mathbf{x}(t) = s_j(t - \tau_{1,j}) \begin{bmatrix} 1 \\ e^{-j2\pi \frac{sin(\theta)}{2}} \\ \vdots \\ e^{-j2\pi \frac{sin(\theta)(M-1)}{2}} \end{bmatrix} \tag{8}$$

Since $s_j(t - \tau_{1,j})$ is the signal received at the reference antenna, the remaining matrix would allow (knowing the precise angle and distance between antennas) to re-create the entire data matrix from the data of a single antenna. However, in order to obtain the data vector $\mathbf{x}(t)$ without knowing the direction of the source, there is still a requirement to have the physical antenna array.
Let us take into consideration the structure of the signal coming from the source $s_j(t)$. In the case of GNSS (and more specifically GPS), the signal is created by mixing three components: a carrier (sine) wave with period 0.634ns, a C/A code with a period of 1 ms and a data stream, with bit duration of 20 ms. Assuming that the data bits are removed, the remaining signal (carrier + C/A) becomes a periodic signal, with repetition period of 1ms. In other words, for any τ_p that is an exact multiple of 0.001s:

$$s_j(t) = s_j(t - \tau_p) \tag{9}$$

3.1.2 Synthetic arrays

Now, let us consider a single antenna element, moving at a constant velocity v and direction (Fig. 7).

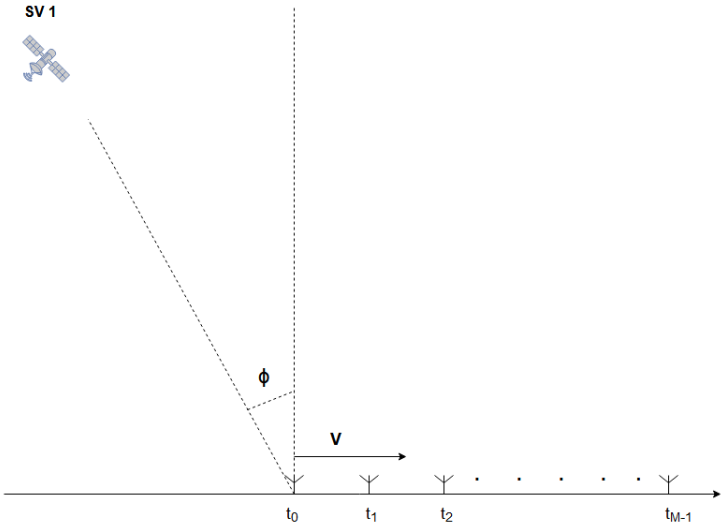

Figure 7: A moving antenna setup with a single signal source.

For this case a single source, identical to the one used previously, is present at an angle ϕ. Assuming that the source is in the far field (angle ϕ does not change for each of the antennas) and that for a "starting time" t_0 the delay between the signal at the source and the signal at the antenna is τ_0, the data received by the antenna becomes (assuming the ideal case of no noise):

$$x_{ma}(t) = s(t - \tau_0 - \tau(t)) \tag{10}$$

In this case, τ_0 becomes a constant, related to the position of the antenna at time t_0, and $\tau(t)$ is the additional delay. This delay is due to the fact that the antenna is changing its position in time (it is not stationary). In fact, the delay due to movement can be expressed as

$$\tau(t) = \frac{sin(\phi) d_{ma}(t)}{c} \tag{11}$$

with $d_{ma}(t)$ being the distance the antenna travelled at the given velocity from the reference point at t_0. This, however, bears strong resemblance to the time delay in the phased array scenario (equation (6)). In fact, let

us compare the signals received in the moving antenna scenario at two time instances: at the reference time t_0 and at time $t_1 = t_0 + n \cdot 0.001s$, where n is an integer. The two signals will become:

$$x_{ma,0} = s(t_0 - \tau_0) \tag{12}$$

and

$$x_{ma,1} = s(t_1 - \tau_0 - \tau(t_1)) = s(t_0 - n0.001s - \tau_0 - \tau(t_1)) \tag{13}$$

Combining (13) with (9), the x_1 signal can be simplified to:

$$x_{ma,1} = s(t_0 - \tau_0 - \tau(t_1)) \tag{14}$$

Once again, if we represent $\tau(t_1)$ as a phase shift due to time delay, the signal $x_{ma,1}$ can be expressed as:

$$x_{ma,1} = s(t_0 - \tau_0)e^{-j2\pi f_c \tau(t_1)} \tag{15}$$

Written in this form, $x_{ma,1}$ becomes of the same form as the signal received at antenna 2 in the phased array case (given in (7)). If the same strategy is used to generate signals $x_{ma,2}$,...,$x_{ma,M-1}$, with $t_i = 0.001ni$; $i = 1, 2, \ldots, M-1$, the following data matrix can be obtained:

$$\mathbf{x}_{ma}(t) = \begin{bmatrix} x_{ma,0}(t_0) \\ x_{ma,1}(t_1) \\ \vdots \\ x_{ma,M-1}(t_{M-1}) \end{bmatrix} = \begin{bmatrix} s(t_0 - \tau_0) \\ s(t_0 - \tau_0)e^{-j2\pi f_c \tau(t_1)} \\ \vdots \\ s(t_0 - \tau_0)e^{-j2\pi f_c \tau(t_{M-1})} \end{bmatrix} = s(t_0 - \tau_0) \begin{bmatrix} 1 \\ e^{-j2\pi f_c \tau(t_1)} \\ \vdots \\ e^{-j2\pi f_c \tau(t_{M-1})} \end{bmatrix} \tag{16}$$

In this case, $\tau(t_i)$ becomes dependent on three factors: angle of the source (unknown), speed of the antenna (known) and the chosen time gap between the two positions of the antenna $t_i = 0.001n_i$, so in fact the selected n). This means that by selecting appropriate positions for the antennas, we can re-create the data matrix (8) entirely using a single moving antenna.

4 Conceptual design

Assuming that we have constructed the data vector $\mathbf{x}_{ma}(t)$ from (16), we have achieved the same spatial sampling as in a conventional physical array. This means that it is possible to treat the data matrix as such, and proceed with the angle of arrival extraction of the source within the dataset. For example, one could decide to follow a subspace approach of the likes of MUltiple SIgnal Classification (MUSIC)[24] or Minimum Variance Distortionless Response (MVDR) to extract the angle of arrival of all sources within the sampled data. However, there is one major difficulty to directly applying such techniques - in the case of GNSS, the signal at the receiver's end is buried below the noise floor. In fact, as shown in Section 2.3.1, it is roughly 20dbW below the thermal noise floor. This low signal-to-noise ratio (SNR) would mean that the direct application of angle estimation techniques on the raw data becomes difficult.
The way GNSS receivers get around this is by, once again, utilizing the correlation properties of the C/A code. As mentioned in Section 3.1.1, the C/A code of a specific satellite provides a way to deterministically generate a local copy of the C/A code, align it with the received signal (in time), and use it to boost the signal above the noise floor. This is why it is more advisable to perform angle of arrival estimation on the signal after the correlation stage, and not prior to it.

4.1 GPS Acquisition and Tracking

To understand better the exact benefits of using the post-correlation signals, this section will focus on two main components of the inner workings of GPS receivers: acquisition and signal tracking. First, Fig. 8 below shows the typical building blocks of a GPS receiver's front-end.

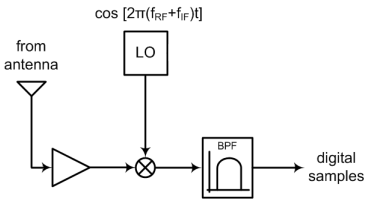

Figure 8: Front-end setup of the GPS receiver used to gather the data used in the examples below. (Original image from [33])

As the L1 carrier frequency is 1575.42 MHz, it is not feasible to directly sample the raw incoming signal (due to Nyquist, the required minimum sampling rate would be too high for implementation in accessible commercial products). Instead, GPS receivers pass the incoming signal through a mixer for down-conversion. This is done via mixing of the incoming signal c_{inc} and a locally generated c_{LO}, which is typically at a slightly lower frequency. In the case of L1, if $f_{c_{inc}} = 1575.42$ MHz and $f_{c_{LO}} = 1565.872$ MHz, the result would be the shift of the information embedded within L1 at two new frequencies: $f_{c_{out,low}} = f_{c_{inc}} - f_{c_{LO}}$ and $f_{c_{out,high}} = f_{c_{inc}} + f_{c_{LO}}$. Adding a low-pass filter would result in preserving the L1 information at much lower Intermediate Frequency (IF) of $f_{IF} = 9.548$ MHz. This intermediate signal can be sampled by widely available ADCs. Fig. 9 shows an example of the power spectrum of a sampled GPS signal (hardwdare setup used can be found in Appendix A). The sample was taken on the 21st of June 2019 in the area of Delft, and will be used to visualize the processes explained in the next parts of this chapter. Fig. 10 shows the sky plot of the GPS constellation at the given time and location.

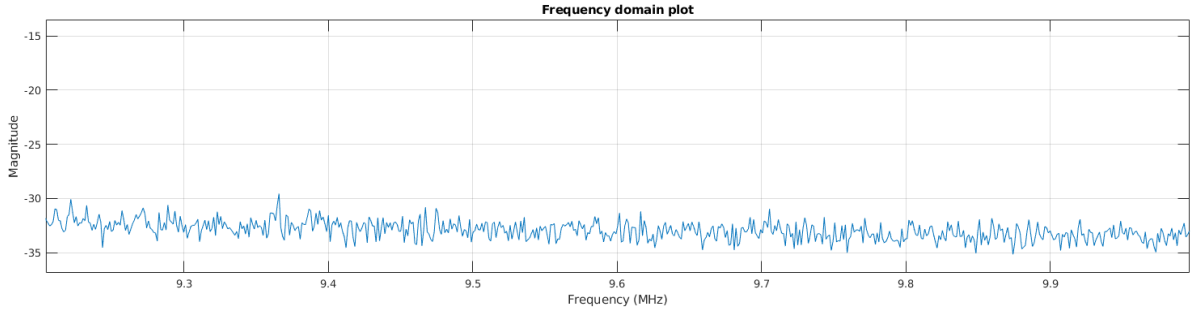

Figure 9: Power spectrum of captured L1 signal at IF of 9.548 MHz.

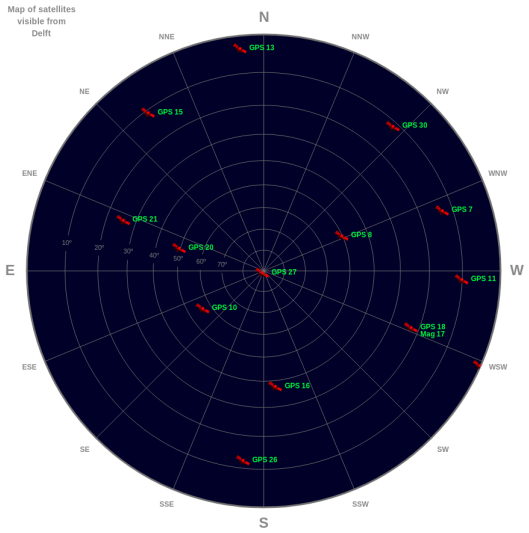

Figure 10: Skyplot of visible satellites for 21.06.2019 at 16:22 near Delft.

4.1.1 Acquisition

The acquisition step of a GPS receiver serves to determine which are the GPS satellites that are currently available. As mentioned earlier, each satellite is assigned one of 32 unique PRN sequences, the properties of which were previously given in Section 1.1. The receiver can use these properties to confirm which of the PRN sequences is present within the combined sampled signal, which was defined in (2). While there are different implementations of the acquisition algorithms, the flow chart given in Fig. 11 shows the general steps.

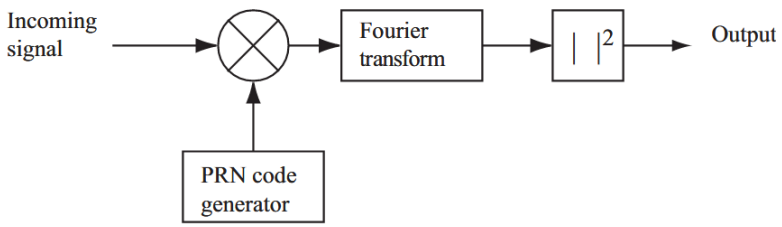

Figure 11: Flow chart of the Parallel Frequency Space acquisition algorithm[?].

The first step is to select a PRN sequence that will be checked for. For the purposes of this explanation we will focus on PRN 15 (part of the visible constellation during the aforementioned experiment). The receiver then generates the 1023 bit PRN sequence (as one of the properties is that the code itself is deterministic). This local copy will be identical to the one used by the GPS 15 satellite, but with an unknown phase shift. As the code is repeated every 1 ms, at a sampling speed of 20 MHz, each PRN code will have a length of 20000

samples. Due to the auto-correlation property of the PRN sequences, any time shift between the received and locally generated copies of the PRN sequence will result in close to no correlation. This means that during acquisition, the receiver takes a 1 ms sample, and generates 20000 copies of PRN 15, each with a different offset of the PRN sequence. Each of these copies is then separately correlated with the sample. Fig. 12 shows the result of this procedure for PRN 15. In this example, a single peak is visible, with significantly larger correlation magnitude compared to any other offset. This allows for the acquisition algorithm to confirm that the provided 1 ms sample contains signal coming from satellite GPS 15, and also allows for a perfectly aligned copy of the C/A code to be generated (in this case, the offset of PRN 15 is measured to be 7373 samples).

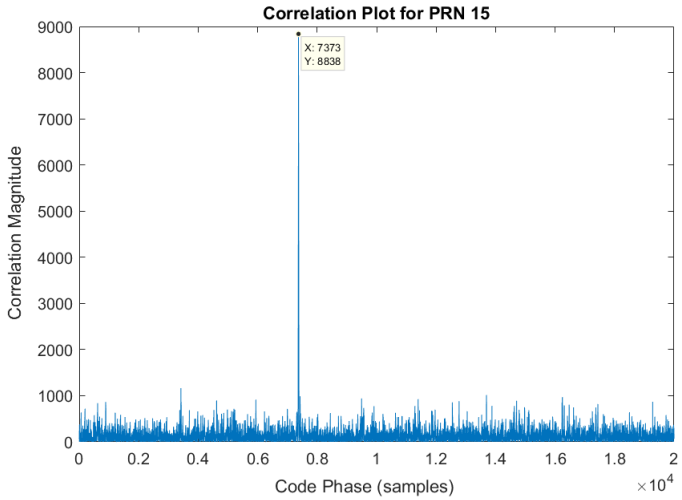

Figure 12: Correlation results for PRN 15 over different offsets of the locally generated sequence.

In addition, the receiver has information regarding the original carrier frequency $f_{c_{inc}} = 1575.42$ MHz, and the down-converted signal $f_{IF} = 9.548$ MHz. However, these do not take into consideration any shifts in the frequency, caused by the relative motion between the satellite and the receiver (Doppler shift). It has been estimated that in the worst case, the largest Doppler shift on the carrier signal would be ± 10 kHz. This means that the receiver needs to find the Doppler shift for each of the incoming signals, as this is needed for the extraction of the navigation message later on. Thus, one of the "tasks" of the acquisition module is to find the precise intermediate frequency for each of the visible satellites.
To do so, the receiver performs demodulation of the aligned PRN sequence by using the previously found offset. This is done by multiplication of the sample by the aligned local copy (in this case of PRN 15 with offset 7373 samples). A conceptual visualization of the demodulation process is given in Fig. 13 - on top we have the modulated L1 carrier + PRN sequence signal. When multiplied with the aligned PRN sequence, the result (bottom) is a perfect sine wave.

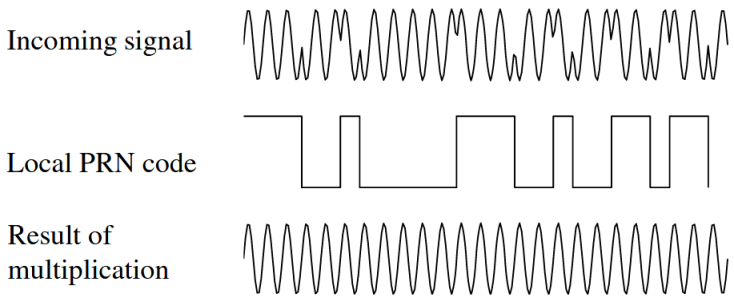

Figure 13: Demodulation of the incoming signal with a locally generated PRN sequence [**?**].

This would mean that within the power spectrum of the demodulated 1 ms sample there should be a spike at the frequency of the demodulated carrier frequency (or more precisely, at the down-converted IF). Fig. 14 below shows the power spectrum of such a demodulated sample for the case of PRN 15. On the left image the local copy of PRN 15 was intentionally offset by 100 samples, while the right image depicts proper demodulation with an aligned sequence. There is a noticeable peak at $f_{IF,PRN\_15} = 9.5478$ MHz.

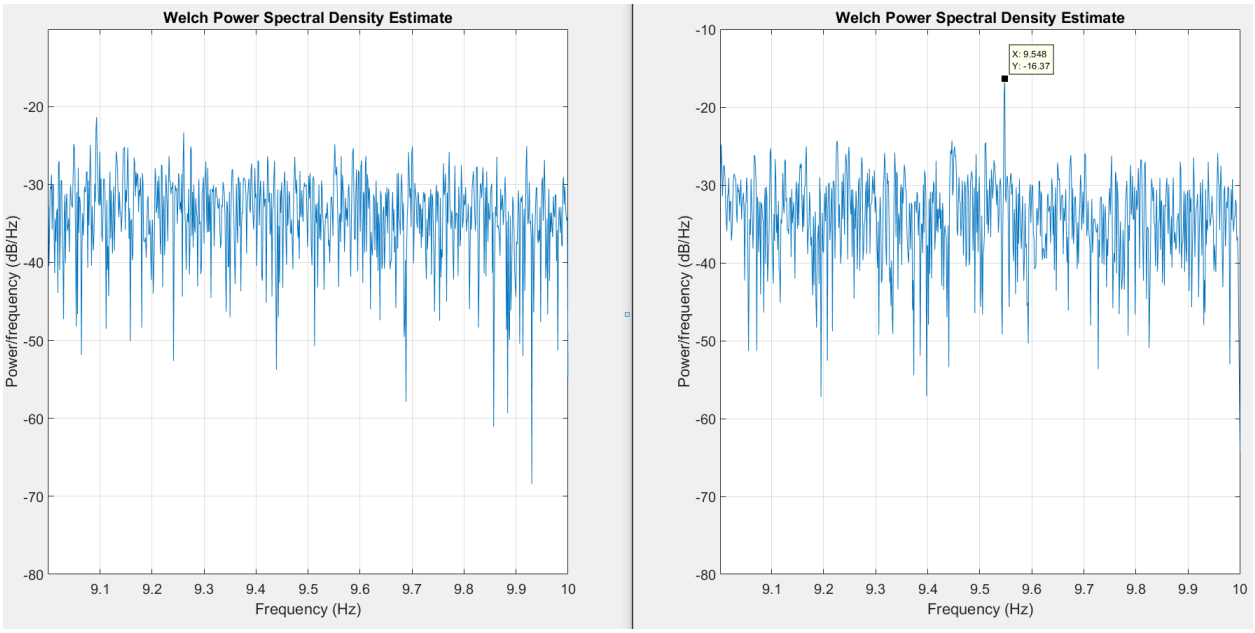

Figure 14: Power spectrum of the demodulated signal for PRN 15. Left: miss-aligned PRN 15 replica (with 100 samples). Right: perfectly aligned PRN 15 replica.

After repeating the same process for every possible PRN sequence, the Acquisition step establishes a "starting" point for the receiver operation. The combined C/A code phase and IF search results for PRN 15 can be seen in Fig. 15. This information is later used during the carrier and code tracking stage.

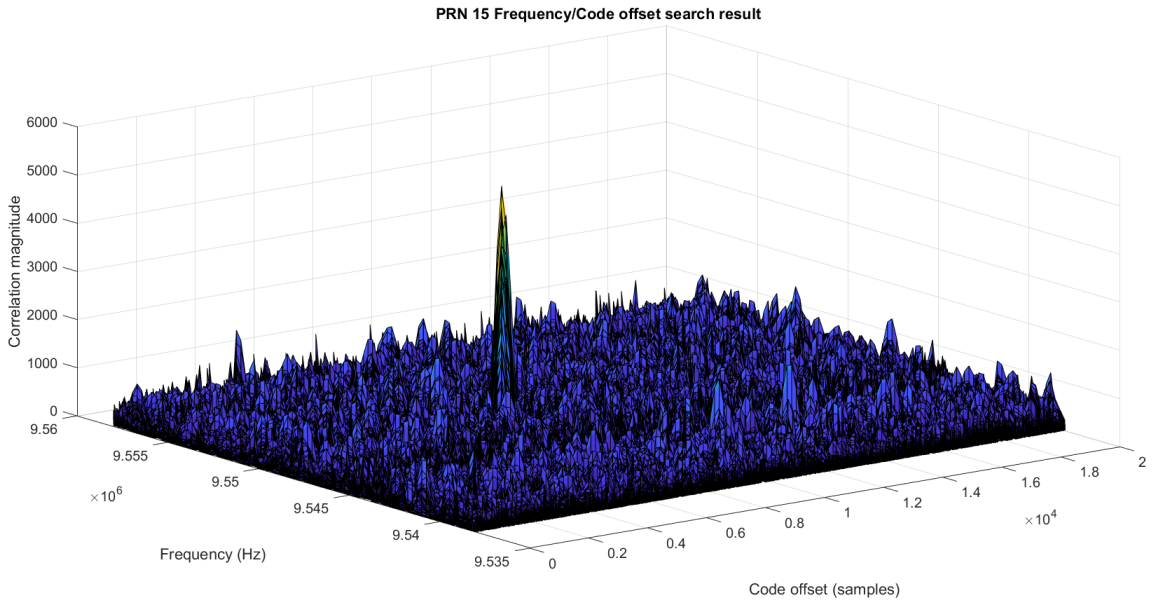

Figure 15: PRN 15 Frequency and Code offset search results.

4.1.2 Tracking

Once the initial state of the incoming signal is established, the functional goal of the GPS receiver is to extract the navigational data bits embedded within each of the incoming satellite signals. Let us consider a receiver in a situation with satellite j within its field of view. The received signal $x_j(t)$ can be expressed as:

$$x_j(t) = \sqrt{2P_j^C} C_j(t) D_j(t) cos(2\pi f_{L1} t) \tag{17}$$

where P^C is the power of the L1 C/A signal; $C_j(t)$ is the C/A code assigned to satellite j; the navigation data is labeled as $D_j(t)$ and f_{L1} is the carrier frequency of the L1 signal. Following the operational sequence previously described, the signal passes through the down-conversion and sampling portion of the front-end, resulting in:

$$x_j[n] = \sqrt{2P_j^C} C_j[n] D_j[n] cos(\omega_{IF} n T_s) \tag{18}$$

with n denoting the sample index of the ADC samples ($n = 1, 2, \ldots$); ω_{IF} being the intermediate frequency resulting from the down-conversion. As mentioned, the goal is to extract the navigational data bits $D_j[n]$. To do so, the signal must first be converted to baseband, by mixing it with a locally generated replica of the down-converted carrier signal at the intermediate frequency ω_{IF}. This would result in [1] [7]:

$$x_j[n] \cos(\omega_{IF} n T_s) = \sqrt{2P_j^C} C_j[n] D_j[n] \cos(\omega_{IF} n T_s) \cos(\omega_{IF} n T_s) = \frac{\sqrt{2P_j^C}}{2} C_j[n] D_j[n] (1 + \cos(2\omega_{IF} n T_s)) \tag{19}$$

Which could be filtered via an LPF to maintain only the first component $\frac{1}{2} C_j[n] D_j[n]$. As displayed earlier, removing the C/A code (the PRN sequence) can be done by multiplication by an aligned, locally generated PRN code, which would result in [7]:

$$\sum_{n=0}^{N-1} \sqrt{2P_j^C} C_j[n] D_j[n] C_j[n] = \sqrt{2P_j^C} N D_j \tag{20}$$

with N being the number of samples within one C/A code repetition. From equations (19) and (20) it becomes obvious that for the correct extraction of the navigation bits, the receiver needs to continuously adjust for any change in the phase of the intermediate frequency carrier replica and/or in the code phase of the locally generated PRN sequence. This is where the carrier and code tracking loops come into play.

In order to continuously adjust for any change in the intermediate frequency of the L1 carrier, GPS receivers utilize two lock loops. A phase lock loop (PLL) is used to adjust for phase in the locally generated carrier, and a delay lock loop (DLL) is used to adjust for and code phase that is experienced throughout the processing of the incoming signals.

4.1.3 DLL loop operation

First, let us consider the acquired signal matrix $\mathbf{x}(t)$ from (3), focusing on the signal at the reference antenna, or $x_1(t) = s_j(t - \tau_{1,j})$. Following the earlier described GPS signal structure, the signal at the source j is of the form:

$$s_j(t) = C_j(t) D_j(t) cos(2\pi f_c t) \tag{21}$$

where $C_j(t)$ is the C/A code assigned to satellite j; the navigation data is labeled as $D_j(t)$ and f_c is the carrier frequency of the L1 signal. At the receiver's end:

$$x_1(t) = C_j(t - \tau_1) D_j(t - \tau_1) cos(2\pi f_c (t - \tau_1)) \tag{22}$$

After down-conversion and sampling (front-end), the signal is ran through acquisition, which (as shown before) provides us with the $f_{IF,j}$, or the intermediate frequency for source j, with adjustment for any Doppler shift, as well as the exact offset for the locally generated PRN sequence for the source at hand. Let us assume, for simplicity, that throughout the operation, the PLL is indeed providing us with carrier information sufficient to always keep a perfectly matching locally generated copy of the carrier signal. The baseband signal (after mixing with the local copy) at the tracking stage will be of the form:

$$x_1[n] = \frac{1}{2} C_j[n - \frac{\tau_1}{T_s}] D_j[n - \frac{\tau_1}{T_s}] \tag{23}$$

In order to simplify, let us consider that τ_1, the time it takes for the signal to travel from the satellite to the first (reference) antenna position, to be 0. At this point in the operation of the receiver, the goal is to generate a perfectly aligned local copy of the $C_j[n]$ component of the baseband signal. As described earlier, the properties of the PRN sequences allow for very precise alignment by evaluating the correlation values for different offsets.

[1] Formula (19) uses $\cos\alpha \cos\beta = \frac{1}{2}(\cos\alpha + \beta \cos\alpha - \beta)$

The way the DLL loops performs this is by generating 3 local copies of the PRN sequence: an *early* code, which is offset by $-\frac{1}{2}$ bit of the last known offset; a *prompt* code, which is at the same offset as the last known offset; and a *late* code, which is offset by $\frac{1}{2}$ bit of the last known offset. Each of the 3 local copies of the PRN sequence is then correlated with the sample at hand, resulting in 3 numeric measures, I_E being the result of the *early* code ($C_{E,j}$) correlation, I_P being the result of the *prompt* code ($C_{P,j}$) correlation and I_L - the result of the *late* code ($C_{L,j}$) correlation[7].

$$I_E = x_1 * C_{E,j} \tag{24}$$

$$I_P = x_1 * C_{P,j} \tag{25}$$

$$I_L = x_1 * C_{L,j} \tag{26}$$

The goal of the DLL is to ensure that *prompt* value I_P is the highest of the three. If, for example, the code is falling behind, the highest correlation value would be the *late* correlator I_L. In this situation, the DLL will adjust the internal offset for the PRN generation, making sure that in the next iteration, the *prompt* code has the same offset as the *late* code from the previous iteration. Fig. 16 visualizes this case.

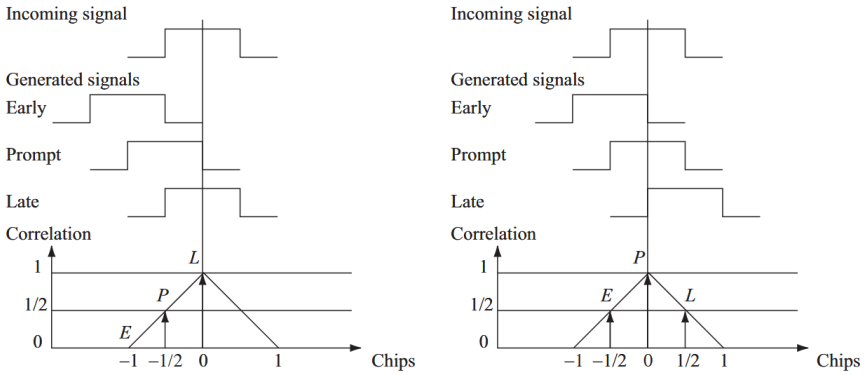

Figure 16: Operation of the DLL loop. Left: a situation where the late correlator has the highest resulting correlation value. In this case the DLL loop will make an adjustment. Right: a situation where the prompt correlator has the highest resulting correlation value. In this case the DLL loop is in sync [**?**].

In order to increase the robustness of the DLL loop when errors of the carrier sync are present, the sampled data is turned to I/Q samples (by mixing with a 90 degree offset version of the locally generated carrier during baseband conversion). The same 3 metrics are calculated for the Q component, meaning that the DLL now has a total of 6 measures of its performance (including $Q_{E,j}$, $Q_{P,j}$, $Q_{L,j}$, all calculated in a an equivalent way to the in-phase components). After the correlators, we can express the *prompt* branch of the processing chain as

$$x_{prompt} = I_P + jQ_P \tag{27}$$

As there might be potential frequency/phase missmatch between the carrier and PRN codes during the calculation of x_{prompt}, when derived for the non-ideal case, it becomes

$$x_{prompt} = \alpha D_j R_j[\omega_{prompt,j}] e^{j(\omega_e[n]t[n]+\theta_e)} \tag{28}$$

With $D_j[n]$ being the navigational data bit, $R_j[.]$ being the normalized auto-correlation with the locally generated C/A code, $\omega_{prompt,j}$ is the code phase between the received and locally generated PRN codes for satellite j, ω_e being the frequency difference (error) between the locally generated carrier and the received carrier, and θ_e being the initial phase difference of the received signal.
The most important factors that must be kept in mind is that during tracking, the GPS receiver adjusts for any change in the carrier frequency/phase shift and for any drift in the alignment of the locally generated PRN sequences. This also means that if (due to Doppler) the number of samples holding one period of the PRN sequence changes, the tracking loop will accommodate for it during the processing of incoming data. However, it provides an already implemented system, capable of describing the current state of all incoming GPS signals.

Having described both acquisition and tracking, it is now clearer that within the workings of all GPS receivers there are already components that allow for isolating of a specific signal from the constellation, boosting its signal power through PRN demodulation, and carefully tracking for any changes in the system during its operation cycle. This provides a strong point for considering the possibility of utilizing the same approach in the attempt to estimate the directions of arrival of the signals from the visible GPS constellation.

4.2 Angle of arrival estimation based on Carrier Signal

4.2.1 Classical Beamscan

The first approach to be considered is the classical way of scanning an array of sensors for directional information, versions of which were presented by Lacoss [26] and Capon [27]. Looking back at the data model derived in (8), the received data $\mathbf{x}$ can be described in the general form $\mathbf{x}(t) = s(t)a(\theta)$, where $s(t)$ is the data as emitted from the source, and $a(\theta)$ is the directional vector, describing the phase delays on the signal caused by the distance between the sensor elements. The most simplistic way of working our way back from X to S is to reconstruct $a(\theta)$. As $a(\theta)$ is dependant on three components: wavelength, sensor spacing and angle of arrival (assuming EM waves in free space), provided that the frequency of the incoming signal and the sensor spacing are known, it is still possible to find the unknown angle of arrival. By constructing all possible $a^H(\theta)$, and calculating the power of the $\mathbf{x}(t)a^H(\theta)$ estimation of S, we essentially scan the directional relationship of the constructive and destructive interference of the signals coming from each sensor element. By locating any maxima on the resulting curve, one would be able to estimate the origin direction of the signal. However, this approach has shown to low have resolution, and relatively poor performance when trying to distinguish between sources that are close in either frequency or direction. Another drawback is that due to the periodic nature of the $a(\theta)$ component, there will be a significant number of sidelobe effects present in the power spectrum.

While this approach could prove to be effective when dealing with a single source, it is expected that in a GNSS application, the close frequencies of incoming signals could impact the overall performance of this approach. Another strong dependency is on the SNR of the present signals. As the GPS signal is with low power upon reception, it is expected that direct application on pre-correlated data would be pointless. However, it might be possible to use information obtained from the Acquisition or Tracking stages to demodulate data samples, and extract carrier phase for a specific satellite. This could allow for a single-source estimation, which could eliminate the negative effects of sources with similar frequencies.

4.2.2 Minimum Variance distortionless response (MVDR)

The Minimum Variance Distortionless response approach aims to construct a beamformer that is dependant on the environment (noise), in order to decrease the distortions on the original signal. As the data correlation matrix $\mathbf{R}_x$ is effectively an index for the spatial correlation between noise and the signal incident on the sensor array. Instead of limiting the approach to dealing only with Gaussian noise, such adaptive beamforming techniques rely on measuring the array correlation matrix, in order to calculate the spatial filter weight. The estimated power at a given angle θ_i can thus be calculated as: [28]

$$P(\theta_i) = [a^H(\theta_i)\hat{\mathbf{R}}_x^{-1}a(\theta_i)]^{-1} \tag{29}$$

It should, however, be noted that previous studies on this approach have pointed out circumstances and effects that could decrease the performance of the MVDR AoA approach. For example, in [28] it is stated that due to the fact that the algorithm relies on the extraction of the spatial correlation between the signal and the noise, any errors within the placement(spacing) of the sensor elements could degrade the performance of the algorithm. As the premises of this work is the substitution of conventional (and typically precise) phased antenna arrays with a single antenna element that is in motion, this could lead to serious potential problems. While the displacement (and thus the antenna spacing within the model proposed in Section 2.1.2) is to be monitored via an IMU or another source of positional information, it is highly likely that for larger synthetic arrays, the accuracy of the antenna displacement will degrade (accelerometer drift and bias).

4.2.3 Multiple Signal Classification (MUSIC)

A slightly different approach to the problem was proposed by Ralph O. Schmidt (1985) [29]. In his work, Schmidt describes the measured signal at an M element antenna array in the form $\mathbf{x}(t) = \mathbf{a}(\theta) + \mathbf{n}(t)$, or more precisely:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} a(\theta_1) & a(\theta_2) & \dots & a(\theta_D) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_D(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{bmatrix} \tag{30}$$

with $x_i(t)$ being the received data, $a(\theta_i)$ contains the delay information due to angles of the D signal sources and M is the number of antennas.

The proposed approach focuses on the fact that in most applications, the noise and the signals are uncorrelated, with the noise being a zero mean Gaussian white noise. For the purposes of this research this describes a received constellation signal by a conventional antenna array, so the approach is applicable in the ideal situation. Given the uncorrelated nature of the signal components, the covariance matrix $\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^H]$ can be calculated as:

$$\mathbf{R}_x = E[(\mathbf{a}\mathbf{s} + \mathbf{n})(\mathbf{a}\mathbf{s} + \mathbf{n})^H] = \mathbf{a}E[\mathbf{s}\mathbf{s}^H]\mathbf{a}^H + E[\mathbf{n}\mathbf{n}^H] = \mathbf{a}\mathbf{R}_s\mathbf{a}^H + \sigma^2 I \tag{31}$$

where $\mathbf{R}_s$ is the signal correlation matrix; σ^2 is the variance (power) of the noise and I is the identity matrix. As $\mathbf{R}_x$ can only be approximated using the observed samples, or via its maximum likelihood estimation $\hat{\mathbf{R}}_x$:

$$\hat{\mathbf{R}}_x = \frac{1}{N}\sum_{i=1}^{N} x[i]x^H[i] \tag{32}$$

In this case, as the number of observed samples N approaches infinity, the two correlation matrices will become effectively equal. If we assume that $\sigma^2 = 0$, or that there is effectively no noise present, then:

$$\hat{\mathbf{R}}_x = \mathbf{a}\mathbf{R}_s\mathbf{a}^H \tag{33}$$

And, as long as all sources have different angles of arrival,

$$rank(\mathbf{a}\mathbf{R}_s\mathbf{a}^H) = D \tag{34}$$

$\mathbf{R}_x$ becomes a semi-positive definite, with D positive eigenvalues and $M - D$ zero eigenvalues. In the case where the noise power is larger than 0, the data correlation matrix follows equation (31). In such a case, if we perform eigenvalue decomposition over $\hat{\mathbf{R}}_x$, we will be left with D eigenvalues (and corresponding eigenvectors) related to the actual signals present within the signals, and $M - D$ eigenvalues related to the noise within the samples (relying on the fact that the noise and signals are uncorrelated). This allows for the spatial spectrum to be calculated via [29]:

$$P_{MU}(\theta) = \frac{1}{a^H(\theta)\mathbf{e}_n\mathbf{e}_n^H a(\theta)} \tag{35}$$

where $\mathbf{e}_n$ is known as a noise matrix, constructed from the $M - D$ noise-related eigenvectors. As can be seen from here, provided a low SNR, the MUSIC algorithm would start performing less and less accurately. Another prerequisite is the need for prior knowledge on the number of incoming signals, and the requirement for them to be uncorrelated with the noise.

Regarding its application in the field of GNSS signals (GPS in this case), if applied on post-correlation signals (where the PRN demodulation is executed, but the signal is still not converted to baseband), it should be possible to extract the AoA information from the demodulated signal. As shown previously, in the ideal case the demodulated signal consists of a single, amplified carrier signal. This means that for each PRN channel, the MUSIC algorithm can be run for a single incoming source. Potential causes for problems would be remnants from other down-converted carrier signals (from multipath or other satellites), which are not completely nulled in the real life implementation [7].

4.2.4 Carrier Phase Difference Extraction Model (CPDE)

The CPDE approach is different than the previously discussed algorithms. In fact, CPDE was proposed by B. Wang et al. [25] as a low complexity angle of arrival estimation approach, reliant on classical phased array setup for data capture. It relies heavily on data manipulations that are part of the tracking stage of the operation of GPS receivers.

Consider the DLL operational loop, described in greater detail in Section 3.1.3. In a classical receiver (single antenna element), the DLL loop is iterated over every repetition of the PRN code of the sampled signal (or, with every 1 ms of sampled data). In a receiver with M sensor elements (or a synthetically constructed antenna array), if the same procedure is used for each of the M channels, each correlator will be adjusted for any shifts in carrier and code phases. However, this would include the time delays related to spatial sampling, and thus

cause a loss in the ability to extract the spatial information. To avoid this, Akos et al. [25] propose to use locally generated copies of the carrier and the C/A code from channel 1 (reference antenna) in the correlator and integrator steps for each of the M channels. This means that for each channel (aside from the reference), the metric would become:

$$x_i^C[n] = \alpha D_j[n] R_j[\omega_{prompt,j}] e^{j(\phi_e + 2\pi f_c \Delta\tau_{i,j})} \tag{36}$$

With $\Delta\tau_{i,j}$ being the delay between channel i and the reference channel, earlier defined in (6). As mentioned in the beginning of Chapter 3, the data bits of the signal are not important for the evaluation of the angle of arrival, so assuming that this part of the signal can be removed, we are left with:

$$x_{prompt} = \alpha R_j[\omega_{prompt,j}] e^{j\phi_e} \tag{37}$$

and

$$x_i^C = \alpha R_j[\omega_{prompt,j}] e^{j(\phi_e + 2\pi f_c \Delta\tau_{i,j})} \tag{38}$$

For each of our channels $i = 2, 3....M$, the positional time delays are preserved as phase shifts in the post-correlation metrics (28). Using the Carrier Phase Difference Extraction model (CPDE) described by B. Wang et al (2018) [25], it is possible to estimate the time delay $\Delta\tau_{i,j}$ from the product

$$\begin{aligned} x_{prompt} \cdot x_i^{C*} = \{\alpha R_j[\omega_{prompt,j}]\}^2 \cdot [e^{j\phi_e} e^{-j(\phi_e + 2\pi f_c \Delta\tau_{i,j})}] = \\ \{\alpha R_j[\omega_{prompt,j}\}^2 \cdot e^{-2\pi f_c \Delta\tau_{i,j}} \end{aligned} \tag{39}$$

where x_i^{C*} is the complex conjugate of x_i^C. By extracting the *angle* information from (39), one can calculate $\Delta\tau_{i,j}$. Knowing the time delay and the distance between the reference antenna and antenna i, one can simply use (6) to estimate the angle θ_j.

4.2.5 Code Phase AoA

The last method that can be attempted is mostly employing the GNSS PRN structure, and should not be considered as a spatial processing approach by definition. Nevertheless, it could prove to be quite reliable in certain situations. Looking back at equations (13) and (14), it is visible that after waiting for a precise number of periods, the signal is identical since $S(t)$ is periodic. However, for a moving target, an additional delay $\tau(t)$ is present, based on the elapsed time and speed of the receiver. If we take a τ that is sufficiently big enough to be larger than the sampling period, then this would mean that the C/A code phase would change, depending on τ. As explained in Section 3.1.1, running a 1 ms data sample through the Acquisition loop would provide a Code Phase indicator ϕ_0 (given in number in samples) of the current alignment of the local replica of the PRN sequence versus the received version. For a static receiver, working under the assumption that over short periods of time the distance between the source and the receiver will not change sufficiently to cause a PRN shift, this offset will remain the same with the next 1 ms sample. However, for a moving receiver, this PRN offset would change, dependant on the angle between the movement direction and the position of the signal source. If at that point the same Acquisition sequence is performed, a slightly different Code Phase indicator ϕ_1 will be given. The difference between the two Code Phases $\Delta_\phi = \phi_0 - \phi_1$ would in fact be a rough estimate of the path difference that the signal had to travel between the two locations. Similar to the derivation of equation (6), the angle of arrival then becomes:

$$\theta = \arccos \frac{\Delta_\phi * (1/f_s) * c_0}{d} \tag{40}$$

Where f_s is the sampling rate; c_0 is the speed of EM waves in free space and d is the distance between the two locations of the antenna.

It should be kept in mind that while this approach is very low-complexity, it is also very dependent on a big number of variables. For example, the angular resolution would decrease if the distance l is small (directly related to the velocity of the moving antenna). This means that at low speeds this approach may prove to yield no useful data. One way of compensating for this is to increase the sampling rate, which would in turn increase the angle resolution. However, most receivers use down-converters that bring the IF lower, so that the L1 information can be sampled with less expensive ADCs. This could find potential use in high speed platforms, such as airborne drones or airplanes.

4.3 Position estimation via IMU

Having described the proposed signal model, as well as approaches for extraction of spatial information, it is also important to examine the second key part of such a system - the estimation of the position of the antenna during operation. While such information can be provided by many sensor systems with different reliability and complexity (for example linear motion encoders can provide high accuracy positional measurements, but bind the motion of the system only along pre-defined tracks[34], Inertial Measurement Units (IMUs) are widely integrated into many platforms from different fields of application - from agriculture [35] to aviation [36]. IMUs typically consist of accelerometers and/or gyroscopes along a number of axis. For example, a 6 Degree Of Freedom (DOF) IMU such as [37] can measure acceleration and rotation along each of the three main axis X, Y and Z. The obtained data is (typically) a sampled version of an analog representation of each of the 6 measured components.

The gyroscopes in this 6 DOF IMU measure the angular velocity $\omega_k(t)$, where k indicates the index of the axis and t is time. These sensors are typically corrupted by a bias $\beta_{\omega,k}(t)$, which is not constant, but time-variant. In addition, the measured values are also affected by measurement noise $e_{\omega,k}(t)$. This means that the resulting sampled signal would be:

$$y_{\omega,k}(t) = w_k(t) + \beta_{\omega,k}(t) + e_{\omega,k}(t) \tag{41}$$

The measurements of gyroscopes can be described as Gaussian [38]. Assuming proper sensor calibration (not covered in this work), the three axis will result in independent measurements, with error $e_{\omega,k}(t) \sim N(0, \sigma_{\omega,k})$. In some use cases, the bias can be modelled as a value that is constant over short periods, and that can be obtained via calibration tests before each use. In order to serve the purpose of providing positional information regarding the system motion over extended periods of time, this approach can be considered unsuitable. Another way of modelling the bias is to assume that it can be expressed as a random walk [38]

$$\beta_{\omega,k}(t+1) = \beta_{\omega,k}(t) + e_{\beta,\omega,k}(t) \tag{42}$$

where $e_{\beta,\omega,k}(t) \sim \mathcal{N}(0, \sigma_{\beta,\omega,k})$. In this such case, the variation of the bias and the overall angular velocity can either be determined via calibration experiments, or can also be obtained from manufacturer information.

The 3 accelerometers (assuming the same 6 DOF IMU used as an example so far) measure the acceleration along the three axis. The acceleration $a_k(t)$ measured by the sensor for the corresponding axis k is also similarly affected by a bias $\beta_{a,k}(t)$ and instrumentation noise $e_{a,k}(t)$.

$$y_{a,k}(t) = a_k(t) + \beta_{a,k}(t) + e_{a,k}(t) \tag{43}$$

It is important to note that apart from the linear components of the acceleration of the system, the accelerometers will also measure the effect of gravitational acceleration, meaning that the measurements provided by the acceleromenters also contain information about the angular orientation of the system (the acceleration g). In fact, accelerometer measurements can be typically described as dominated by the vector of gravitational acceleration, meaning the data estimation model can be given:

$$y_{a,k}(t) = -\mathbf{r}_k(t)g + \beta_{a,k}(t) + e_{a,k}(t) \tag{44}$$

where $\mathbf{r}_k(t)$ would be the rotation matrix ($\mathbf{r}_k$ is a matrix that retains the length of a vector, but modifies its direction upon multiplication) required to obtain the current acceleration component from the gravity acceleration g at the current orientation.

4.3.1 Modelling the dynamics of the system

The dynamics of motion of systems represent the relation between the different components that describe the motion itself - position p_k, velocity v_k, acceleration a_k, etc. Limiting the system to acceleration (ignoring effects of jerk, snap and so on), means that the equations describing this system are:

$$v_k = \frac{\partial p_k}{\partial t} \tag{45}$$

$$a_k = \frac{\partial v_k}{\partial t} \tag{46}$$

From these, assuming constant acceleration between two consecutive samples (simplifying the system), the system can also be described by:

$$v_k(t+1) = v_k(t) + a_k(t)\Delta t \tag{47}$$

$$p_k(t+1) = p_k(t) + v_k(t)\Delta t + \frac{(\Delta t)^2}{2}a_k(t) = p_k(t) + \Delta t(v_k(t-1) + a_k(t-1)\Delta t) + \frac{(\Delta t)^2}{2}a_k(t) \tag{48}$$

where Δt is the time step (time between two sequential samples). Adding the directional information, it becomes apparent that the *state vector* $x(t)$ will be described by [38]

$$x(t) = \begin{bmatrix}(p_k(t))^T & (v_k(t))^T & (a_k(t))^T & (\omega_k(t))^T\end{bmatrix}^T \tag{49}$$

4.3.2 Probabilistic models and Extended Kalman filters

We have now derived the accelerometer and gyroscope models as:

$$y_{a,k}(t) = \mathbf{r}_k(t)(a_k(t) - g) + \beta_{a,k}(t) + e_{a,k}(t) \tag{50}$$

$$y_{\omega,k}(t) = w_k(t) + \beta_{\omega,k}(t) + e_{\omega,k}(t) \tag{51}$$

With noise and bias as previously defined. For position estimation, this leads to a state space model of [38]:

$$\begin{bmatrix}p_k(t+1)\\ v_k(t+1)\end{bmatrix} = \begin{bmatrix}p_k(t) + \Delta t(v_k(t-1) + a_k(t-1)\Delta t) + \frac{(\Delta t)^2}{2}\{\mathbf{r}_k(t)[y_a(t) - \beta_a(t)] + g + e_{p,a}(t)\}\\ v_k(t) + \Delta t[\mathbf{r}_k(t)(y_{a,k}(t) - \beta_{a,k}(t)) + g + e_{v,a}(t)]\end{bmatrix} \tag{52}$$

These questions show the non-linear nature of the system that is being modelled. Traditional filters, such as the traditional Kalman filter, rely on linearization of the system, thus impacting the performance [39] [38]. However, there is a modified version of the traditional KF, that extends its performance to non-linear systems by computing estimates of the conditional probability distribution [39] [40]. An Extended Kalman Filter (EKF) uses a model (non-linear) to embed information regarding the relationship between the state space and measurement space. For this, it is important to note that the process and measurement noise are both zero-mean white Gaussian, with the measurement noise being additive [38]. The state space is:

$$x(t+1) = f(x(t), w(t)) \tag{53}$$

$$y(t) = h(t) + e(t) \tag{54}$$

With $x(t)$ being the state space, $w_t \sim \mathcal{N}(0, Q)$ and $e_t \sim \mathcal{N}(0, R)$. $h(t)$ is a transformation matrix for conversion from state to measurement space. An EKF uses the non-linear system model, and proceeds to linearize that at the current estimated point. This keeps the Gaussian nature of the probability density function.

The process of linearizing differential equations (or a system of them), is by calculating (for each) partial derivatives (Jacobians). This is done in order to estimate matrices $\mathbf{F}$ (state transition matrix [39]) and $\mathbf{H}$ from the functions f and h from above. These result in:

$$\mathbf{F} = \frac{\partial f(x,u)}{\partial x}\Big|_{x,u} \tag{55}$$

$$\mathbf{H} = \frac{\partial h(\hat{x})}{\partial \hat{x}}\Big|_{\hat{x}} \tag{56}$$

The remaining part of the EKF follows the same steps as traditional Kalman Filters. There have been many implementations and publications for the use of different implementations of the EKF for the estimation of attitude (direction) of a moving system using an IMU. Kok et al [38] analyze and compare the performance of multiple apporaches for estimation of both attitude and positional information from a similar setup. In their evaluation of the EKF implementation, they do clarify that for the position estimation, it is beneficial to include positional measurement information (for example from a GNSS system) in order to provide both initial conditions, as well as an additional source of cancelling the effect of the sensor bias. In addition, R. Labbe [39] shows implementations (in simulations) of the Unscented Kalman Filter (UKF), a further modified

version of the EKF that introduces the so called Unscented Transform - an alternative approach to the EKF for approximation of the non-linear transfer function being applied to a Gaussian. An example implementation is presented by C. Liu et al [41] even shows an UKF implementation that uses a different IMU setup - instead of adding 3 gyroscope axis measurements, it uses 4 triaxial accelerometers. While this increases the complexity of the process (higher dimension problem due to larger number of variables), the reported performance the UKF with respect to the estimation of angular velocity (also providing information about orientation) is presented as acceptable. Yet another implementation is presented by J. Sola [42], in which both orientations and positional data can be estimated via sensor fusion (addition of GNSS or other sources of positional information).

5 Results

The suggested set of AoA estimation approaches was implemented in Matlab, alongside a modified version of the SoftGNSS [7] implementation of a GPS receiver. The planned tests on the theoretical applications were split into two sections: test on the implementation with simulated data, as well as a test dataset gathered using a Software Defined Radio platform, allowing for wide variety of front-end modifications.

5.1 Simulation

The GPS signal modification was built based on the specifications of the L1 GPS signal structure, only omitting the addition of navigation data bits. Overall, the simulation starts by generating a signal following the form given in equation (10). First, for simplicity, the initial delay τ_0 is set to 0. This can be done without loss of generality, as this distance (based on the primary assumptions made in the previous sections) should not change over small periods of time. By following the Gold Code sequence for known GPS PRNs, the C/A code is generated and used to simulate the combination of a carrier and spreading code coming from a single source (simulated SV). The IF that is to be generated by the simulator is defined, and potential doppler shifts caused by receiver motion are calculated. This follows the reception signal model at the GPS receiver, defined in (17) with the difference that no data bits are transmitted. Once the basic simulation was established, the complexity is further increased by introduction of noise (at various SNR levels), increasing the number of sources (each with a different angle of arrival) and exploring the effects of different Doppler shifts on each satellite due to SV movement.

A validation dataset was generated, simulating 6 visible satellites (parameters are given in the table below). Upon execution of the acquisition stage, the receiver's output confirmed that the included PRN sequences were indeed of valid structure. In addition, the randomly selected Phase Shifts placed within the data were correctly estimated during the Acquisition stage for each of the 6 embedded signals. As a final check, it was observed that the Phase Shifts would gradually drift (with a different direction and speed) for each of the satellites - an expected behaviour.

Velocity	100m/s
Sampling rate	38.192MHz
Intermediate frequency	9.548MHz
Angles	[30, -45, 60, -10, 0, 80]
Phase Shifts	[123, 521, 1063, 4261, 2130, 302]

5.1.1 Test number of sources

The first goal is to observe the performance of the different algorithms when the simulated data consists of an increasing number of signals. First, a baseline performance is established by selecting a random PRN sequence (in this case PRN 5), and generating 1 second sample data. The SNR was set at -10db, sampling rate was set at $f_s = 20$ MHz, with a velocity of receiver being 157m/s. This velocity was selected as it allows for a virtual antenna spacing of 0.5λ to be achieved each 100 ms (at the simulated IF). This allowed for the construction of a synthetic virtual array with 8 antenna elements. The angle of arrival for the simulated signal was set at 37 degrees. Given below are the results of runnning the simulated data through the system.

MUSIC	BeamScan	MVDR	Code Phase AoA
36.0	37.0	37.0	33.1

The baseline test confirms that the angle estimation using post-correlation data yields correct angle estimation, based on the phase information of the embedded IF carrier signal. The error of the estimation for all 4 spatial algorithms is within a degree, which is expected. The final approached tested with this simulated data was the Code Phase AoA approach. In this case the estimated angle of arrival is 33.1 degrees. To visualize the effect of sampling speed on the angular resolution, the same experiment was run at $f_s = 40$ MHz. The estimated angle in this case is 37.9 degrees. Finally, the performance of the MUSIC, MVDR and BeamScan implementations can be seen on Fig. 17 below.

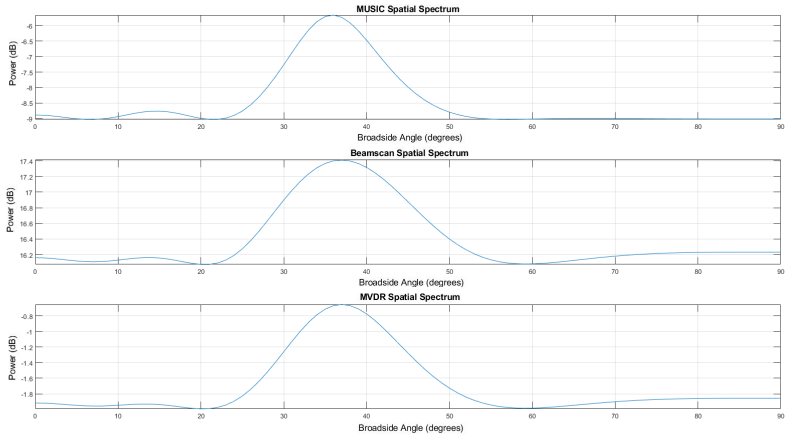

Figure 17: Angle estimation plots for MUSIC, Beamscan and MVDR. The simulated angle is 37 degrees.

The second test introduces a second PRN signal (PRN 10) at 63 degrees. In this case the signal model was built using the form of (3) for the received signal and (16) for the constructed synthetic array, using M = 8. The higher sampling rate was kept for the rest of the experiments, the angular resolution of the Code Phase AoA approach is significantly higher.

PRN	MUSIC	BeamScan	MVDR	Code Phase AoA
5	37.0	37.0	37.0	33.1
10	37.0	72.0	69.0	62.5

The second tests shows a quick degradation of the 4 spatial AoA estimation approaches. Using the Spatial spectrums (given in Fig. 18) it is possible to better understand the causes of the degraded estimation. For example looking at the plots for PRN 5 (left), it is expected that if the post-convolution data was ideal, all components from other signals inside the sample would be suppressed. While it is known that during demodulation of a composite signal there are residual components that could degrade carrier clarity [7], it was not expected that the effect would be that big. Another factor that could contribute to the performance degradation is the fact that the IF carrier signals of the internal signals are separated by approx. 1Hz apart - due to the Doppler shift of the receiver. In a real-life data sample, the motion of the satellites will introduce much more distinct frequency separation (due to the significantly higher velocity of satellite motion). On the positive note, the Code Phase AoA approach continues to perform within the same comparable error margin as in the baseline (provided that the higher sampling rate is used).

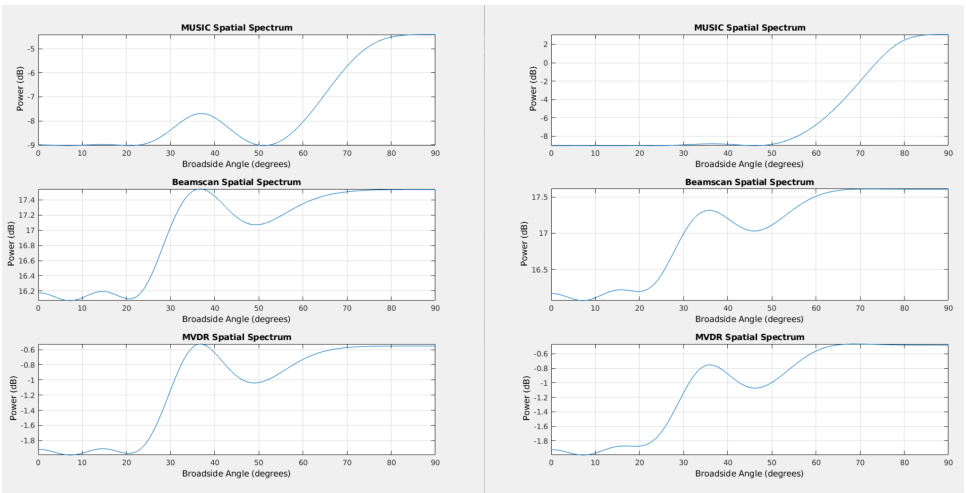

Figure 18: Angle estimation plots for MUSIC, Beamscan and MVDR. The simulated angles are 37 degrees for PRN 5 (left) and 63 degrees for PRN 10 (right) .

The third test consisted of a simulated signal at -10dB with a total of 6 included PRN signals.

PRN	Actual angle	MUSIC	BeamScan	MVDR	Code Phase AoA
5	37.0	35.0	38.0	39.0	33.1
10	63.0	35.0	39.0	39.0	62.5
15	15.0	35.0	39.0	39.0	15.8
20	0.0	36.0	39.0	39.0	0.0
25	41.0	36.0	39.0	40.0	43.0
27	79.0	36.0	39.0	39.0	72.7

As evident from the obtained results, the more signals are inserted at the same SNR, the lower the overall performance of the AoA approaches becomes. After looking at the spacial spectrum, it becomes more apparent where the cause of the issue is (Fig. 19). Given below are the spectrum using PRN 10 (left) and PRN 20 (right). Based on the overall power levels, it appears that the demodulation phase is not capable of clearly removing other carrier fragments. Another possibility is the low selected SNR. This will be investigated in the next subsection. However, the 6 channel simulation is good test-case for the Code Phase AoA. While the accuracy of the approach appears to decrease at larger angles. This can be explained by the linear spatial resolution (uniform sampling intervals and constant speed), which is converted to a linear resolution on the sin, not directly to the angle. This means that upon conversion to an angle, the resolution of estimation will become more and more coarse the larger the angle becomes (as observed). However, if applied in a system where accuracy is less important than reliability in determining whether satellite positions appear to overlap (spoofing detectior), this approach could prove to be quite effective.

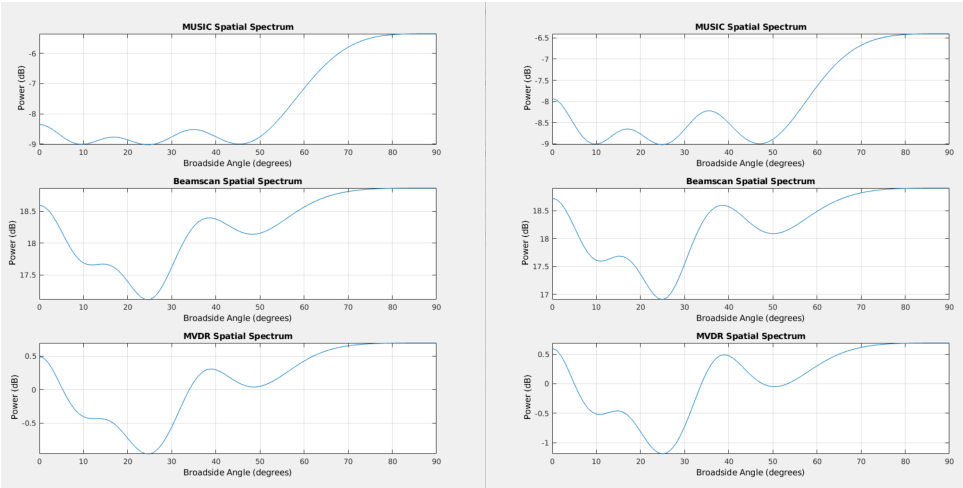

Figure 19: Angle estimation plots for MUSIC, Beamscan and MVDR. The simulated angles are 63 degrees for PRN 10 (left) and 0 degrees for PRN 20 (right).

5.1.2 Test effects of SNR

The second part of the performace analysis focuses on the effects of higher and lower SNR on the system. First, a signal was generated to investigate if the performance in the 2 signal case can be improved by bringing the SNR from -10dB to 0dB. All other parameters are kept the same.

PRN	MUSIC	BeamScan	MVDR	Code Phase AoA
5	37.0	38.0	37.0	33.1
10	36.0	78.0	65.0	62.5

As clearly visible, increasing the SNR to 0dB does not seem to significantly improve performance. The next step would be to observe behaviour at positive SNR. The last simulation test with 2 sources is at SNR of 20dB. Fig. 20 shows that in this case, MVDR is capable of splitting between the 2 comprising signals much better than the other two algorithms. It should be noted that as the working theory of these experiments is that the post-correlation data should contain coherent information only about the source with that specific PRN. In this case, a second run of the PRN 5 demodulated data is used in a run where the expectation is that 2 signals are present within the data sample. As visible in Fig. 21, while the improvement is not dramatic, it does seem to allow for the recognition of the PRN 5 angle of arrival in the case of MUSIC.

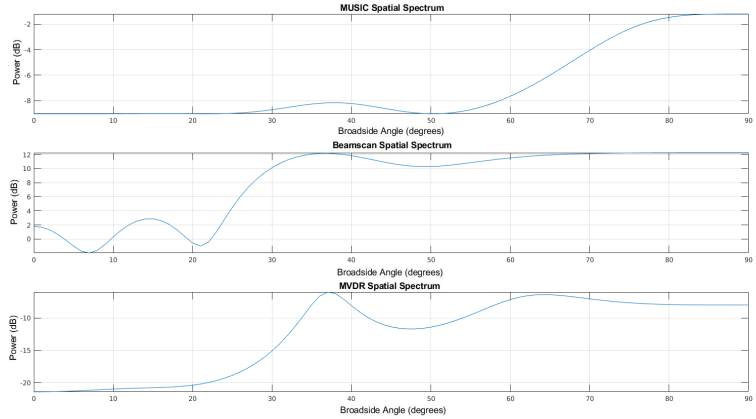

Figure 20: Angle estimation plots for MUSIC, Beamscan and MVDR at 0dB. The simulated angles are 37 degrees for PRN 5 (left) and 0 degrees for PRN 10 (right).

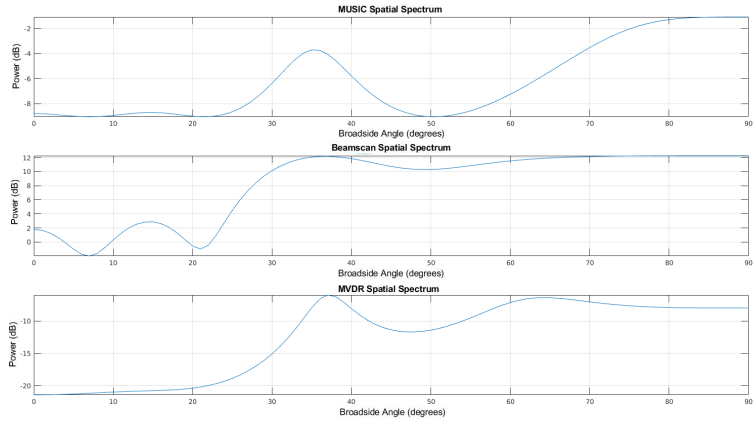

Figure 21: Angle estimation plots for MUSIC, Beamscan and MVDR at 20dB. The simulated angles are 37 degrees for PRN 5 (left) and 0 degrees for PRN 10 (right).

5.1.3 Frequency separation

A possible explanation to the diminishing performance when introducing more signals is the frequency separation of the different simulated signals. While the Doppler shift due to the motion of the receiver, none of the above took into consideration the frequency shift to the IF, caused by the orbital motion of the satellites (in a genuine signal) with respect to the receiver. In order to investigate the effects, the simulation was modified to incorporate slight changes to the IF frequency of each part of the simulated signal. To do so, a dataset from using a static receiver, tuned to the same IF via its down-converter, all IFs for all visible PRNs were obtained. The same were then introduced to the modified simulation. The frequencies used are listed below:

PRNs	[5, 6, 12, 18, 24, 29]
Angles	[30, 45, 60, 10, 0, 80]
Phase Shifts	[123, 521, 1063, 4261, 2130, 302]
IF	[9.5476 MHz, 9.54967 MHz, 9.54867 MHz, 9.5469 MHz, 9.5503 MHz, 9.54867 MHz]

First, in the 2 source case Fig. 22, there is a major improvement over what was previously observed, especially for the case of the MUSIC algorithm. While the particular angles are not completely accurate (34 degrees estimated versus 30 degrees simulated for PRN 5 and 43 degrees estimated versus 45 degrees simulated for PRN 6), the post-convolution signal does a much better job of separating the signals. In addition, there is also some improvement in the performance of beamscan and MVDR, even if not as strong. The rather small angle separation (15 degrees) makes it so that the maxima in both approaches is not as clearly distinguishable.

Nonetheless, the estimated angles are 33 degrees and 42 degrees for Beamscan and 31 degrees and 43 degrees for MVDR respectively. It should also be noted that the results were obtained at a signal strength of 0dB.

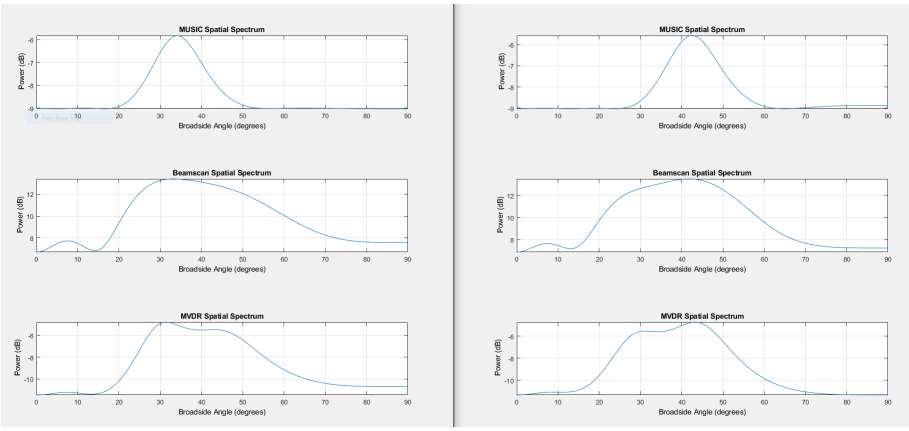

Figure 22: Angle estimation plots for MUSIC, Beamscan and MVDR at 0dB. The simulated angles are 30 degrees for PRN 5 (left) and 45 degrees for PRN 6 (right).

Following this experiment, the number of sources was increased to 6. In this the performance degrades once again, in a similar fashion to the previous experiment. While some of the peaks visible across the different approaches match the simulated angles, not all are identifiable. In addition, this does not match the ideal case expectations. The expected interaction in the post-correlation step is that the convolution process will null the effects of all sources, except for the precise match. The fact that there is little difference between the spatial power spectrum for each of the simulated PRNs (example given in Fig. 23) means that certain components are not being cleared out well. As what remains after correlation is mainly the carrier (at the IF), it is also likely that the similar structure of the signals (also purely repetitive) degrades the behaviour even more.

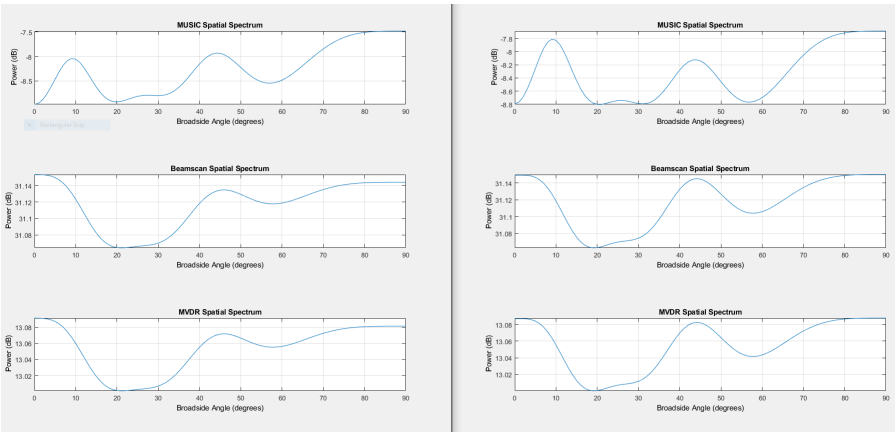

Figure 23: Angle estimation plots for MUSIC, Beamscan and MVDR at 0dB. The simulated angles are 30 degrees for PRN 5 (left) and 45 degrees for PRN 6 (right).

To ensure that the correlation was performed correctly for the same data, a comparison of the power spectrum is given in Fig. 24. The frequencies estimated for PRN 05 and PRN 06 (carrier at IF) are 9.5476 MHz and 9.5495 MHz respectively. The resolution of the spectrum is increased by using a longer signal sample. The duration of correlation used for this estimation is 10 ms instead of the 1 ms used for the angle estimation. Further check at the rest of the simulated PRNs show that, indeed, the carrier at IF (at the correct IF) can be isolated via the correlation procedure.

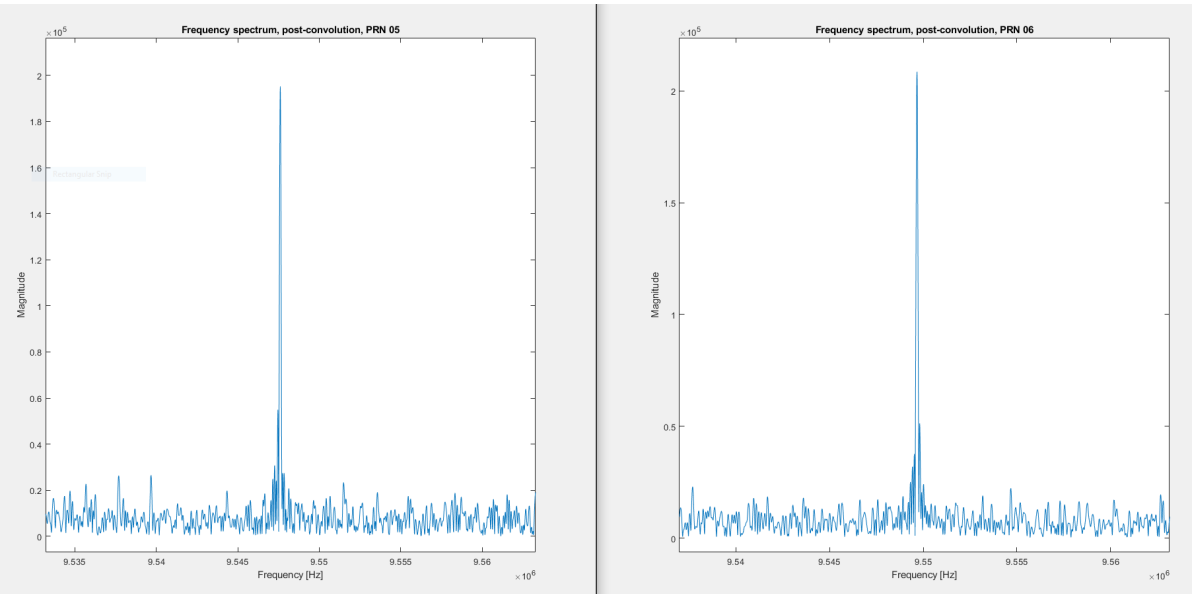

Figure 24: Frequency spectrum comparison for post-correlation data of PRN 05 and PRN 06

5.2 Field test

The field test took place on 21st of June 2019 on a straight road in the vicinity of Delft. The sky plot, previously given in Fig. 10 shows the distribution of the visible GPS satellites. In order to replicate a moving platform at a relatively constant speed, the antenna active GPS antenna of a Software Defined Radio (Ettus N200 [32]) was mounted on the hood of a car. The location of the road, as long as its orientation can be seen on Fig. 25. Once the experiment started, the car accelerated to 36kmph, and maintained a stable speed for at least 10 seconds. In that period (once the velocity reached "stable state"), a dataset was recorder. This was repeated several times, gathering multiple datasets at different sampling speeds (20 Msps, 33 Msps). Once recorded, the data was labeled (for time and direction of movement), which allowed post-processing. For more details on the equipment and control software, please refer to Appendix A.

Figure 25: Location data sampling experiment. Orientation is NE to SW along a straight road.

5.2.1 Validation of the captured signal

Once acquisition was complete, the signal validity had to be ensured. After running several samples through signal acquisition, it was confirmed that the expected PRN codes were visible. In addition, tracking was executed for multiple samples, and confirmed to be performing correctly. An example is given on Fig. 26. On the bottom, one can see the correlation results of the three branches described in Section 3.2.5. The metric in this particular example is not only the correlation of the I component, but can be treated in a similar manner. At the start of the tracking process, we can observe correlation values for the *late* and *prompt* branches to be rather similar. However, as the tracking loop adjusts for this, the *prompt* branch has consistently higher performance metrics than the other two (thus being in the "synchronized" mode, shown earlier in Fig. 16). In addition, in a way to show the navigation bit extraction given in eq. (20), the top graph shows the embedded navigation data for the entire 7 seconds of processed sampled data.

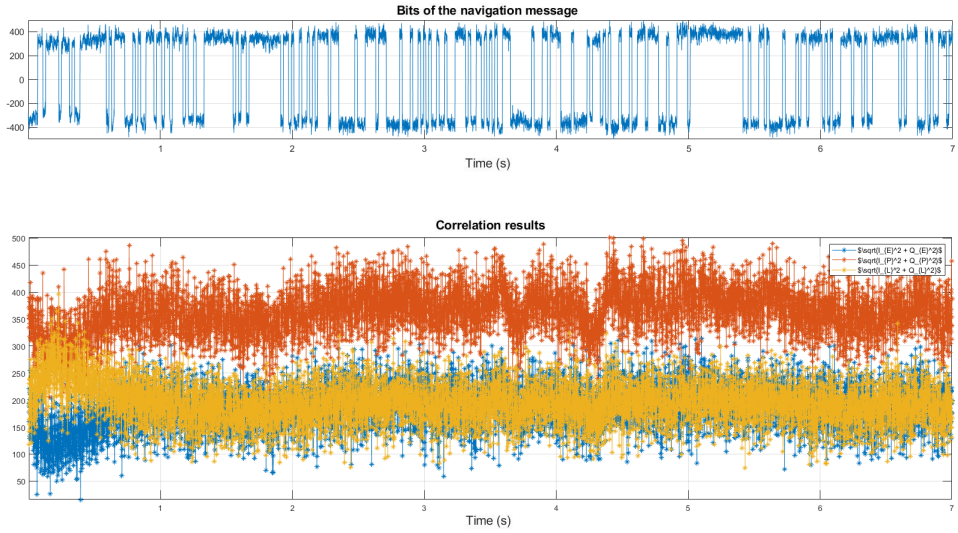

Figure 26: Tracking loop of PRN 18 for 16:22 at 21.06.2019. Given in blue is the early, orange is the prompt and yellow is the late correlators.

5.2.2 Post-correlation test

The second part of the performance analysis replicates the approach used during the simulation analysis. As the sampled data will always include the sum of all available signals (following the GPS signal model given in (2)), it is impossible to test the model for the "single source" case. For the first test, a sampling rate of 33.3 Msps was used. As the system was moving at approximately 10 m/s, a $\frac{1}{2}\lambda$ element spacing was achieved by taking sections of the sampled data of 10ms apart (distance between the elements becomes 0.1 m, while $\frac{1}{2}\lambda$ at L1 carrier is 0.095 m. In this way, using data collected over 80 ms, allows to construct an 8-element synthetic array. After applying the PRN correlation for each of the available satellites, the MVDR, MUSIC and Beamscan methods were applied (Fig. 27):

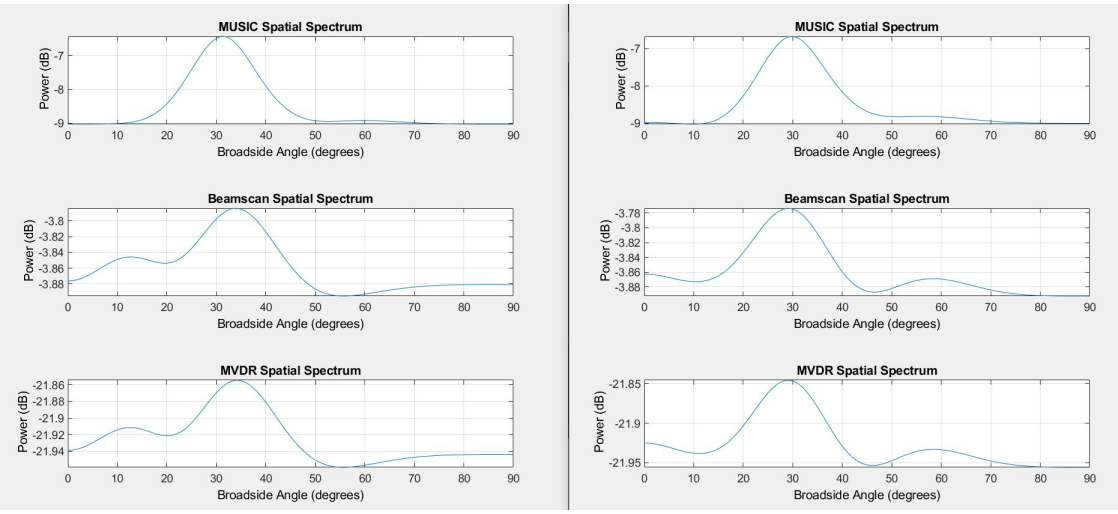

Figure 27: Post-correlation AoA run of MUSIC, MVDR and Beamscan for PRN 07 (left) and PRN 08 (right).

While in each of the cases (PRN 07 and PRN 08) the estimated angles between the different approaches is rather consistent (difference of approx. 5 degrees between them), these angles are incorrect as both peak at around 30 degrees (PRN 07 is expected to be at 10 degrees and PRN 08 at roughly 50 degrees). This confirms some of the findings from the simulations: the captured signal during the test is a sum of multiple GPS signals, and it appears that during the correlation with the locally generated PRN sequences there are artefacts remaining from the other sources. Looking at the spectrums of the post-correlation results of PRN 18 and PRN 27 shows the same behaviour (Fig. 28). Similar behaviour was observed when repeating the same experiment with other datasets at 20 Msps, as well as a different trajectory of the moving receiver.

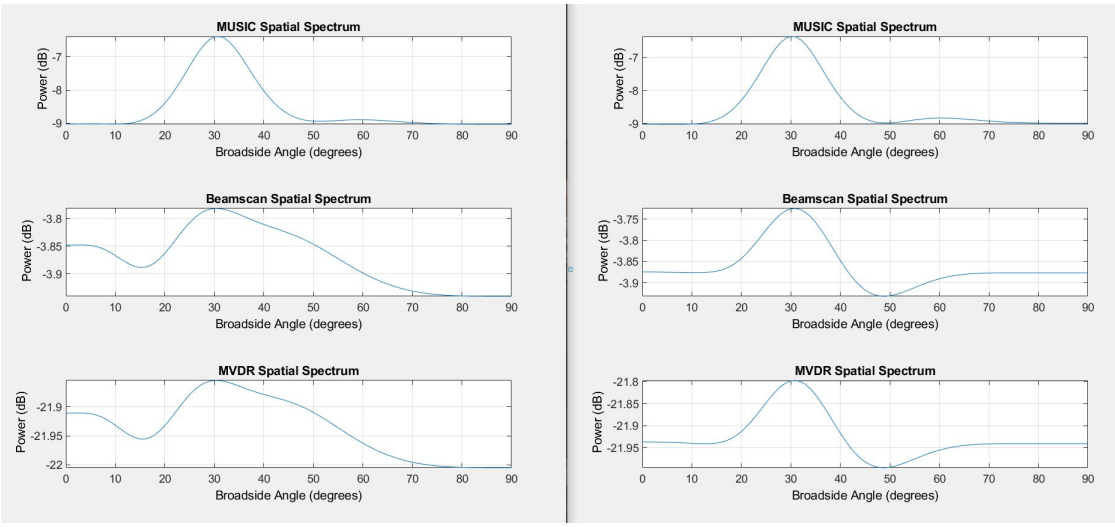

Figure 28: Post-correlation AoA run of MUSIC, MVDR and Beamscan for PRN 18 (left) and PRN 27 (right).

The next test focused on the Code Phase AoA approach. The first attempt included calculating the C/A code offset for PRN 27 at two different points in time 100 ms apart (separated by an exact number of samples equal to 100 ms to preserve the periodicity of the expected signals). For this time, the distance between the two receiver locations becomes $d = 1$ m. Using the same 20 Msps dataset, with this specified time offset, the estimated distance due to Code Phase was estimated to be 44.97 m. Plugging the findings in (45), the cos of the unknown angle θ becomes 44.97, which does not respond to an angle in the range of $[-\pi, \pi]$. Further analysis shows that at the used resolution, the minimum measurable signal path difference becomes $c_0/f_s = 15$ m. At $d = 1$ m, it is obvious that it becomes impossible to measure an angle at the current combination of receiver velocity and sampling speed. However, the large C/A offset measured between 100 ms of data is too large to be contributed to receiver motion. This finding points towards the possibility that one of the assumptions made within the signal model early on, regarding the consistency of the time needed for the signal to travel between the source (SV) and the receiver over brief periods of time, may in fact not hold. To better test this, all C/A offsets for each 1 ms period over a 400 ms period were calculated for two different channels (Fig. 29)

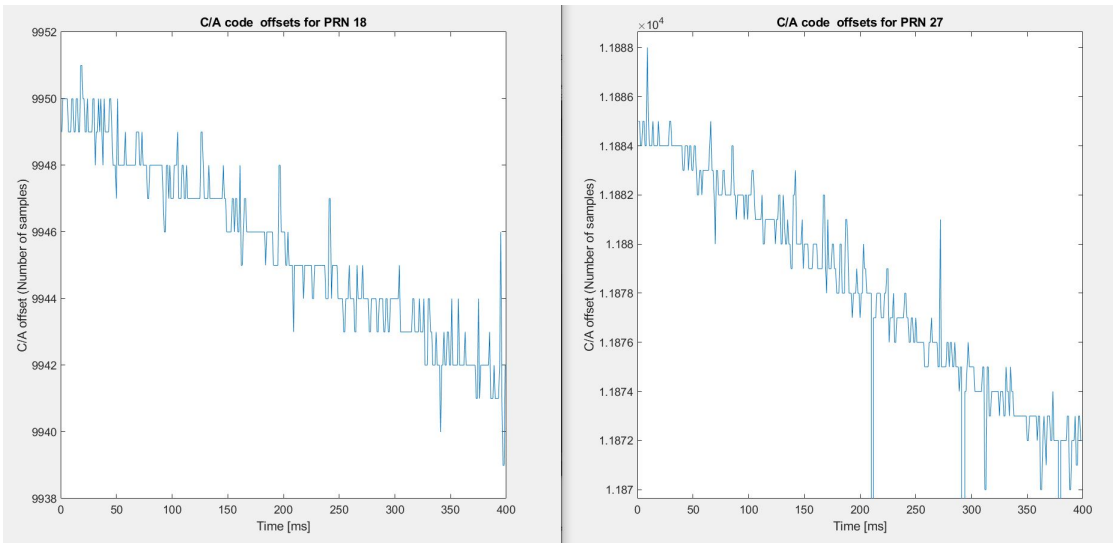

Figure 29: C/A codes for PRN 18 (left) and PRN 27 (right).

There are two important findings from the analysis. First of all, it is visually confirmed that the combined motion of the system and the sources results in a gradual, and in the case of a consistent velocity and trajectory relatively consistent slope of the C/A offset values. This confirms the basic principal behind the C/A code offset approach, but also shows the large C/A code offset change over relatively short periods of time. It should be noted that SV 18 and 27 for that period of time were at two very different positions: at approx 40 degrees for

SV 18, and directly "above" the system for SV 28. However, the more important finding is the instability of the PRN offset for consecutive 1 ms periods: the overall shape of the plot resembles a noisy stair-case function. As the resolution the C/A code is measured in number of samples, in this particular case the time resolution becomes $1/f_s = 0.00000005$ s. There are multiple possible reasons for this behaviour, some of which are oscillator instability at the receiver's end [43] or even ionospheric effects on the GPS signals.

To better understand the contributions of the different factors to this situation, a reference data file was used to repeat the same C/A calculations. The dataset (provided by [39]) contains a high-gain static receiver, at the same IF and comparable sampling rates. As the receiver is stationary, any variation in the observed offsets (which bare resemblance to the ones observed in Fig. 29) can be contributed to external factors. Fig. 30 shows the observed C/A offsets for two of the channels (PRN 15 and PRN 21) over 400 ms.

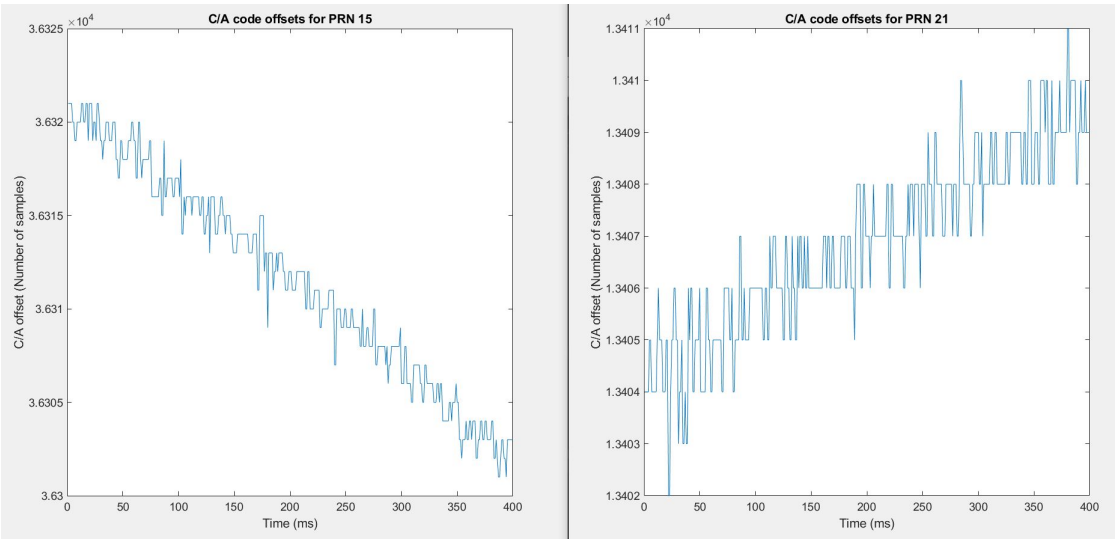

Figure 30: C/A codes for PRN 15 (left) and PRN 21 (right).

A similar behaviour can be observed for the static receiver as well - the variation between consecutive C/A code phases is present in the static receiver case as well. In addition, the rapid change of the C/A code with time, with a stable slope through the short period of observation, shows that the assumption of consistent C/A offset over brief time periods is not correct in the genuine GPS signal case. Looking at the higher slope of PRN 15 it becomes apparent that the relative velocity between the receiver and GPS 15 is higher than the relative velocity with respect to GPS 21. For the case of PRN 15, we can observe periods of approximately 30 ms where the average of the C/A offset is consistent. Using this knowledge it is possible to expect that the C/A phase AoA approach may be applicable for synthetic arrays, given that the the time difference Δ_ϕ from eq. (45) is smaller than this 30 ms. This also provides the largest drawback of the C/A offset AoA approach - it is strongly bound to the ratio between the speed of the moving receiver (providing the distance d) and the sampling frequency (providing the resolution for the path difference component).

5.2.3 CPDE

The final approach to be tested is the CPDE algorithm. Following its implementation steps, Fig. 31 shows the results of a tracking loop on one of the test experiments with a moving receiver (velocity is 10 m/s, sampling frequency is 33 Msps, constellation map is as shown on Fig. 10). As seen, the tracking loop successfully keeps the *prompt* channel metric above the *early* and *late* correlators (required).

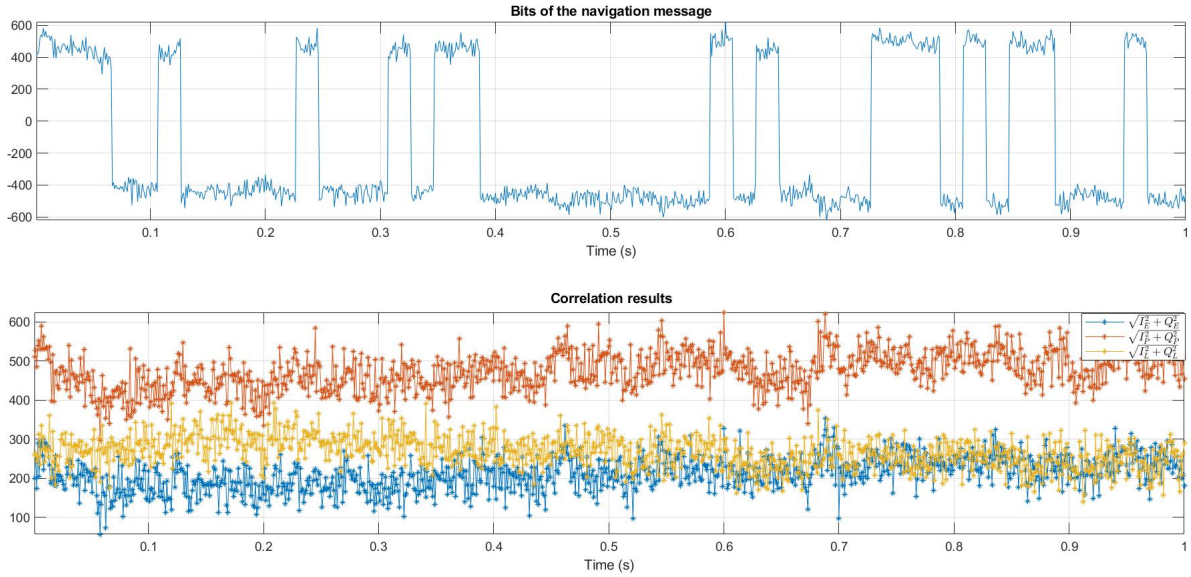

Figure 31: Phase between the reference and secondary channel using CPDE for PRN 8.

As this provides a synchronized local carrier and PRN sequences, it is possible to calculate the phase offset from equation (39). Given in Fig. 32 is the performance of this measure over 1 second, with a distance between the synthetic antenna elements of 10 cm (or time delay of 10 ms) for one of the available SVs (PRN 8). It becomes immediately visible that there is strong variation throughout the test. As the phase is used to calculate the path difference $\Delta\tau$, the estimated angle of arrival will be impossible to calculate (as the arccos function will yield results across the entire $[-\pi, \pi]$ range). The same observation was made for other SVs of multiple datasets, confirming that the approach is not functional with live data.

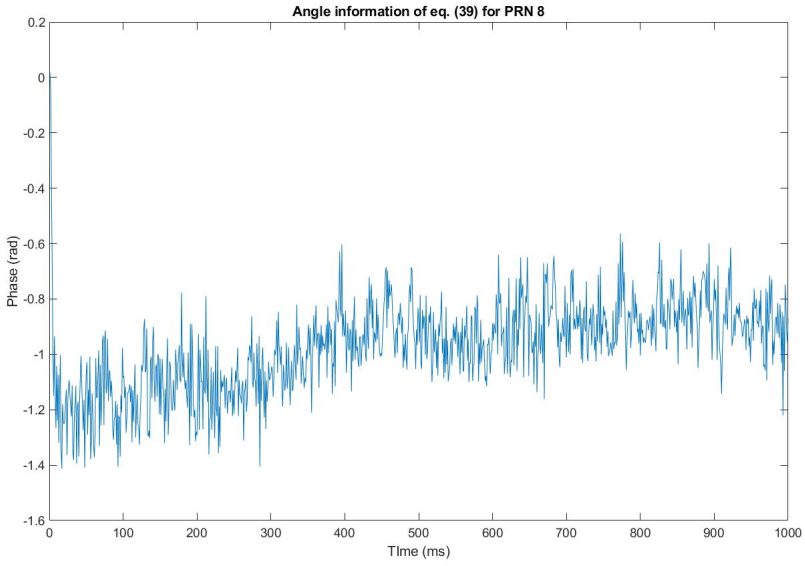

Figure 32: Phase between the reference and secondary channel using CPDE for PRN 8.

In an attempt to find the causes of this behaviour, the same reference dataset (static receiver provided with [39]) was used in an attempt to establish a verification of the approach. As the receiver is not in motion, the estimated phase from (39) should be equal for all available channels - in fact, it should be 0 as there should be no additional time delays that induces additional shift of the received carrier (according to the signal model (16)). The results for two of the SVs (PRN 21 and PRN 22) show more stable behaviour in terms of overall average level, but also have similar variance during a 1 s test (Fig. 33). Comparing the moving versus static receiver,

it is possible to identify the effects of slightly changing velocity of the receiver in the first case. However, the more important effect is the variation itself. As for both of the tested PRNs the values differ within a range of ≈ 0.4 rad, the resulting angles will also have very high variance, making it unreliable.

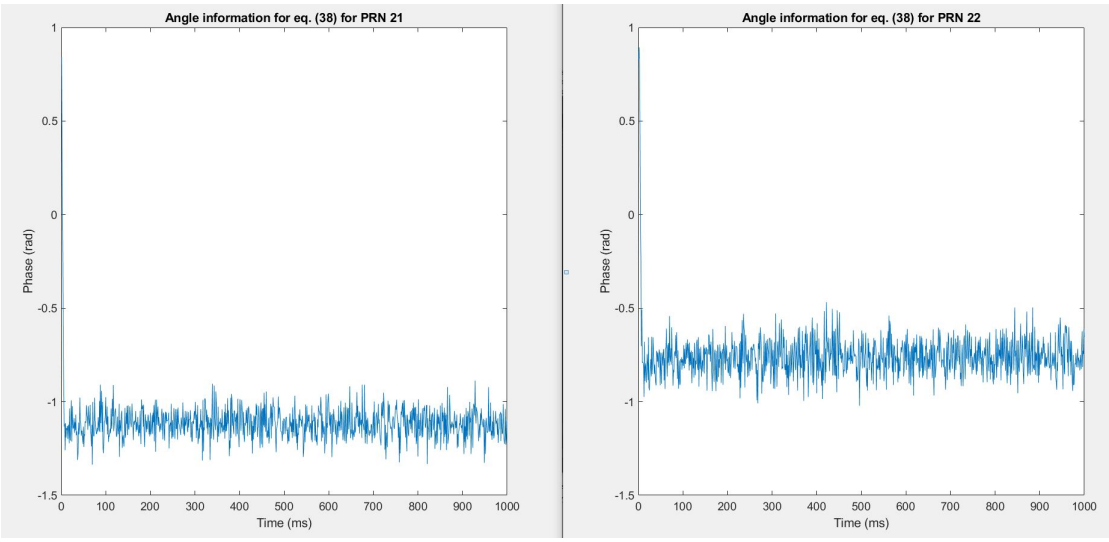

Figure 33: Phase between the reference and secondary channel using CPDE for stationary dataset for PRN 21 (left) and PRN 22 (right)

Looking back at the operation of the CPDE algorithm, its original form was proposed for conventional phased arrays. From Fig. 30 it became clear that the received signal is not as stable over time as it is during simulations. Regardless of the cause (whether due to equipment imperfections or effects due to medium of propagation), it becomes apparent that the expected accuracy and consistency of the periodicity of the incoming signal is not maintained in the real-life application. Looking at equations (21) through (26), the most important parts of the correlators and integrators are the precision in the synchronization of both the PRN sequence, as well as the carrier. From the static C/A code experiment it becomes apparent that during propagation, even when accounted for Doppler shift, the duration of a single period is altered more than expected. This means that while in a traditional phased array the measured signals observe the same incoming wave (thus ignoring any potential modifications of the signal due to, for example, dynamic effects in the ionosphere), in the synthetic array the differences experienced even in consecutive executions of the periodic part of the incoming signal are not consistent enough. In fact, Fig. 33 also confirms that the motion of the satellite over 10 ms is significant enough to cause measurable phase shift.

6 Discussion and Conclusion

The aim of this work is to explore the known issue of spoofing in the field of GNSS signals. Following a research into the different approaches of detection of spoofing (such as Automatic Gain Control (AGC), Signal Quality Monitoring (SQM), Doppler monitoring, etc.), a decision was taken to focus on the Angle of Arrival estimation approach. As a well known method, utilized in a variety of industries and research fields, angle of arrival via phased arrays is still one of the more reliable approaches to spoofing detection, as it can allow for the user to gain spatial understanding of the incoming signals. More specifically, this approach was chosen due to the fact that it relies on the properties of the signal due to its physical propagation rather than the embedded information, as it is well known and easily replicable by a malicious third party source. Focusing on the drawbacks of traditional phased arrays - the physical requirement of multiple receivers with pre-determined spacing, as well as hardware to handle synchronous sampling, the decision was made to address the possibility of using synthetic arrays as a way of utilizing existing degree of arrival algorithms in a more simple hardware set-up. In their essence, synthetic arrays utilize a single moving antenna to provide the spatial sampling required to reconstruct the data matrices.

To provide the position data needed for the tracking of the antenna, the possibility of using Inertial measurement units (IMUs) was explored. As they largely comprise of multi-axis accelerometers and gyroscopes, traditional IMUs often suffer from bias build-up and sensor drift over prolonged use. By introducing an additional layer - in the proposed system a non-linear Extended Kalman Filter (EKF), one can counter these effects by providing a basic model of the dynamics of a moving system. Analysing implementations of EKF and Unscented Kalman filter (a variation of the EKF), it is possible to track the position of a system using IMUs of different complexities.

Using the well-established structure of GNSS (or in the majority of this work GPS as an example) and known properties of its spreading codes (PRN sequences), it was shown that there is periodicity in the signal that is open to the general public. This allows for the reconstruction of a single time series dataset into a data matrix, and essentially providing the same spatial information available to a stationary phased array. Once this is available, it allows for the use of known techniques such as MVDR, MUSIC, classical Beamscan, as well as more specific approaches such as the Carrier Phase Difference Extraction [25] and the Code Phase Direction of Arrival. A major setback to all of the above would be the signal power at a typical GNSS receiver, as it is well below the thermal noise floor. In order to avoid adding complexity to the system, and to make use of existing mechanisms within the GPS processing chain, it was decided to make use of the PRN convolution properties, in order to boost the incoming signal quality, as well as allow for the identification of each of the sources.

A series of simulations were used to analyse the performance of the different estimation approaches using the synthetic array data model. Following the simplistic case of a single receiver gave a promising start, showing that angle extractions are possible. However, it was quickly shown that apart from the Code Phase approach, by increasing the complexity of the simulation (introducing multiple sources of signals with similar structures), all techniques showed a drop in performance. This is likely caused by artefacts produced during the correlation stage - while ideally designed to have close to 0 cross-correlation, it has been previously shown that during demodulation there are remaining parts of the signals coming from other sources within the constellation. The degrading behaviour makes it impossible to locate all sources when the number of simulated SVs was increased above 4. The only approach that remained reliable regardless of the number of sources was shown to be the Code Phase approach. Despite its results, it was also shown that the method itself is highly limited by the physical properties of the receiving system - in its essence, the approach is bound by two factors: the velocity of the receiver and the sampling speed. In order to provide high resolution estimation, the approach requires large distance between two synthetic elements, as well as high sampling rates.

These results were backed up by live experiments with both stationary and moving systems. As shown in Section 4.2, while it is possible to confirm that the demodulation of the signal is successful, it was impossible to extract the angle of arrival in any of the use cases. After some analysis, it was shown that one of the main causes for this lies in the fact that the signal model was constructed in the idealistic case where the propagation medium was (for the most part) ignored - the signal model does not account for dynamic environments, where delays can be observed. Using both stationary and moving receivers, it was shown that the expected periodicity of the signal is not ideal - the calculated C/A codes in either case were with high variance. The last issue, which also greatly contributes to the fact that none of the approaches was applicable for the real-life scenario is that fact that even at very short time periods it was possible to see the effects of the satellite motion, even on a stationary receiver. One of the main assumptions within the signal model was precisely the fact that over brief periods, the distance that the signal travels to the reference position (starting position of the antenna) remains consistent. This was largely disproved during the data analysis stage of the project.

Overall, the prospect of using synthetic arrays for spoofing detections in GNSS does remain open. Despite the negative outcome of the conducted tests, there are areas to improve the signal model. As observed, the satellite

(source) motion cannot be ignored even within small time periods. However, the position and motion of GPS satellites are well known, and can be deduced once a lock is established. By allowing the system to compensate for the motion of the remote sources, it might be possible to greatly increase the usability of the proposed approach. Another open possibility is to analyse the differences in signal behaviour between the constellation signals and spoofer signals, as in most applications spoofers can be considered to be stationary. However, a problem that is hard to address is the fact that the medium that the genuine GNSS signals have to travel through before reaching the receivers is in fact very dynamic. The main contributor for the observed instability is likely to be a combination of multipath effect and ionospheric effects. While mostly present within signals coming from satellites that are closer to the horizon, the ionospheric effects are known to cause issues within GNSS receivers.

References

[1] [1] Statement of work, *"Techniques for spoofing detection and mitigation in Aeronautical receivers"*, ESA, 19/09/2018

[2] [2] J. A. Volpe, *"Vulnerability assessment of the transportation infrastructure relying on Global Positioning System"*, U.S. Department of Transportation, 29/08/2001

[3] [3] K. C. Zeng et al., *"All your GPS are beling to us: towards stealthy manipulation of road navigation systems"*, Usenix Security, 2018

[4] [4] G. W. Hein et al., *"Authenticating GNSS: Proofs against Spoofs, Part 1"*, InsideGNSS, vol. 2, no. 5, pp. 58-63, 2007

[5] [5] G. W. Hein et al., *"Authenticating GNSS: Proofs against Spoofs, Part 2"*, InsideGNSS, vol. 2, no. 6, pp. 71-78, 2007

[6] [6] J. T. Curan, *"(In)Feasibility of Multi-friquency Spoofing"*, InsideGNSS, 2018

[7] [7] K. Borre, D. Akos, *"A Software Defined GPS and GALILEO receiver: A Single Frequency Approach"*, ISBN 978-0-8176-4540-3

[8] [8] D. B. Goldstein, *"Interface Specification ICD-GPS-200, Revision E"*, Global Positioning System Wing, 08/06/2010

[9] [9] D. M. Akos, *"Who is afraid of the Spoofer? GPS/GNSS Spoofing detection via Automatic Gain Control (AGC)"*, Navigation, vol. 59, no. 4, pp. 281-290, 12/2012

[10] [10] C. Hegarty et al., *"Spoofing detection for airborne GNSS equipment"*, ION GNSS+ 2018, 2018

[11] [11] M. Appel etl. al., *"Robust Spoofing Detection and Mitigation based on Direction of Arrival Estimation"*, ION GNSS+ 2015, 09/2015

[12] [12] J. Magiera, R. Katulski, *"Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing"*, Journal of Applied Research and Technology, vol. 13, issue 1, pp. 45-47, 02/2015

[13] [13] M. Appel etl. al., *"Experimental validation of GNSS repeater detection based on antenna arrays for maritime applications"*, CEAS Space Journal, vol. 11, issue 1, pp. 7-19, 03/2019

[14] [14] M. Meurer, A. Konovaltsev, et al., *"Direction-of-Arrival Assisted Sequential Spoofing Detection and Mitigation"*, 2016 International Technical Meeting, 02/2016

[15] [15] T. Lin, A. Broumandan et al., *"Robust Beamforming for GNSS Synthetic Antenna Arrays"*, ION GNSS 09, 2009

[16] [16] A. Cavaleri, B. Motella, M. Pini, M. Fantino, *"Detection of spoofed GPS signals at code and carrier tracking level"*, 2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 8-10/12/2010

[17] [17] Y. Liu, S. Li, Q. Fu, Z. Liu, *"Impact Assessment of GNSS Spoofing Attacks on INS/GNSS Integrated Navigation System."*, Sensors (Basel), 04/05/2018

[18] [18] M. Grewal, A. Andrews, *"Kalman filtering: theory and practice using MATLAB"*, New York: John Wiley and Sons. 14. 10.1002/9780470377819.

[19] [19] J. T. Curran, A. Broumandan *"On the use of Low-Cost IMUs for GNSS Spoofing Detection in Vehicular Applications"*, International Technical Symposium on Navigation and Timing (ITSNT) 2017, 2017.

[20] [20] Y-H. Chen, S. Io, D.M. Akos, D.S. De Lorenzo, *"Validation of a controlled reception pattern antenna (crpa) receiver built from inexpensive general-purpose elements during several live jamming test campaigns"*, 2013

[21] [21] Y-H. Chen, S. Io, D.M. Akos, D.S. De Lorenzo, *"Validation of a controlled reception pattern antenna (crpa) receiver built from inexpensive general-purpose elements during several live jamming test campaigns"*, 2013

[22] [22] M. Jones, *"GNSS Protection Overview 2017"*, 2017

[23] [23] J. R. Merwe, X. Zubizarreta, et al. , *"Classification of Spoofing Attack Types"*, 2018 European Navigation Conference (ENC), Gothenburg, 2018, pp. 91-99.

[24] [24] R. O. Schmidt, *"A signal subspace approach to multiple emitter location and spectral estimation"*, 1981

[25] [25] B. Wang, D. M. Akos et al., *"A Low Complexity GNSS Array Signal Angle of Arrival (AoA) Estimation Algorithm and Validation"*, ACES Journal, vol. 33, n. 10, 10/2018

[26] [26] R. T. Lacoss, *"Data Adaptive Spectral Analysis Methods"*, Geophysics, vol. 36, n. 4, 8/1971

[27] [27] J.Capon, *"High-Resolution Frequency-Wavenumber Spectrum Analysis"*, Proceedings of IEEE, vol. 57, no. 8, 8/1969

[28] [28] C. Vaidyanathan, K.M. Buckley, *"Performance analysis of the MVDR spatial spectrum estimator"*, Proceedings of IEEE, vol. 57, no. 8, 8/1969

[29] [29] R. 0. Schmidt, *"Multiple Emitter Location and Signal Parameter - Estimation"*, IEEE Transactions on Antennas and Propagation, vol. 34, no. 3, 3/1986

[30] [30] H. K. Hwang, *"Direction of Arrival Estimation using a Root-MUSIC Algorithm"*, Proceedings of the International Multi-Conference of Engineers and Computer Scientist, vol. 2, 3/2008

[31] [31] Keith W. Forsythe, *"Utilizing Waveform Features for Adaptive Beamforming and Direction Finding with Narrow-band Signals"*, LINCOLN LABORATORY JOURNAL, vol. 1, no. 2, 1997

[32] [32] Ettus Research, *"Ettus N200/N210 Knowledge Base"*,

[33] [33] ESA Navipedia, "Front end", [$Availableat : https : //gssc.esa.int/navipedia/index.php/Front_End$]

[34] [34] Anaheim Automation, "Encoder Guide", [$Availableat : https : //www.anaheimautomation.com/manuals/forms/encoder - guide.php$]

[35] [35] R. Takai, O. Barawid, *"Development of Crawler-Type Robot Tractor based on GPS and IMU"*, IFAC Proceedings Volumes, vol. 43, no. 26, 2010

[36] [36] J. Wendel, O. Meister, *"An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter"*, Aerospace Science and Technology, vol. 10, no. 6, 9/2006

[37] [37] ST, LSM6DSOX, [$Availableat : https : //www.st.com/en/mems - and - sensors/lsm6dsox.html$]

[38] [38] Manon Kok, Jeroen D. Hol, *"Using Inertial Sensors for Position and Orientation Estimation"*, Foundations and Trends in Signal Processing, vol. 11, no. 1-2, 2017

[39] [39] R. Labbe, *"Kalman and Bayesian Filters in Python"*, 2015

[40] [40] M. A. Skoglund, G. Hendeby, and D. Axehill, *"Extended Kalman filter modifications based on an optimization view point"*, Proceedings of the 18th International Conference on Information Fusion, 2015

[41] [41] C. Liu, S. Yu, S. Zhang, X. Yuan, *"An Effective Unscented Kalman Filter for State Estimation of a Gyro-Free Inertial Measurement Unit "*, Huazhong University of Science and Technology

[42] [42] J. Sola, *"Quaternion kinematics for the error-state Kalman filter"*, 10/2017

[43] [43] E. Tzoreff, B. Z. Bobrovsky, *"Single Receiver Emitter Geolocation Based on SignalPeriodicity With Oscillator Instability"*, IEEE Transactions on Signal Processing, vol. 62, no. 6, 2014

7 Appendix A

7.1 Hardware list

The hardware used for obtaining the signal samples for post-processing consisted of:

- Active GPS Antenna from Ettus Research - 5V Active antenna, providing 27dB gain around the L1 frequency band
- Ettus Research N200 Software Defined Radio - a customize platform, enabling the sampling, mixing and pre-processing of RF signals via programmable interfaces.
- Ettus Research DBSRX2 800-2300 MHz Rx Daughterboard - an add-on card for the N200, responsible for mixing and providing power to the active antenna front-end

To control this, a Linux PC (Ubuntu 17.10), with pre-installed Python, GNU Radio, UHD and GNU Radio Companion was set up. The communication and data channels were established via a Cat 5e cable between two 1GIGABIT Ethernet cards. This was sufficient to maintain data transfer at up to 50 Msps. However, to achieve this without overflowing data buffers, the sampling mode of the N200 has to be switched from 16bit I/Q to 8-bit I/Q. In order to make it compatible to the rest of the software, the 8-bit I/Q data was converted to 16bit floats, keeping only the in-phase part.

7.2 Experiment setup

The antenna of the receiver was placed on the hood of a car. This provides a large ground-plate, and a movable platform that can be monitored (with respect to speed). After locating a suitable location (Fig. 25). The speed of the vehicle was kept at 36kmps (10m//s) to ensure that each 1ms of motion corresponded to 0.1m of physical displacement. The speed was monitored by a driver (human), so oscillations are inevitable. However, based on observation, the average speed was kept consistent.