Simulated Annealing Design of Time-Domain Coding for FMCW Radar Coexistence

Vilgot Hillbom
vi3085hi-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Daniel Sjöberg

Examiner: Mats Gustafsson

June 20, 2024

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

For range, velocity and angle measurements, radar is a common technique to use, be it in the automotive industry, defense industry, or for medical purposes. Radars can achieve accurate measurements while also being unaffected by weather conditions such as rain, fog, or backlight. In many use cases, it might be necessary to use several radar units, which in turn allows for the possibility of causing mutual coherent interference. To avoid this interference, time division multiplexing can be utilized, which degrades the rate of the Coherent Processing Intervals (CPIs). Another possible solution could be to shorten the CPIs to avoid interference while preserving the CPI rate, but this would worsen the velocity resolution. In this work, sparse CPIs that can be transmitted simultaneously without interfering have been designed. This allows for simultaneous transmission without decreasing the CPI rate or degrading the velocity resolution. To preserve as much as possible of the radar units' measurement qualities, finding these sparse CPIs were posed as an optimization problem where the spectral properties of the CPIs were also taken into account. To solve this non-linear and combinatorial optimization problem the probabilistic algorithm Simulated Annealing was used. Naive CPIs which would not cause interference were used to compare with the CPIs resulting from the optimization. The algorithm successfully finds CPIs that would not cause interference and possess better spectral qualities than the naive CPIs. The optimal CPIs were also compared with results generated using the coprime method used in the related field of designing sparse arrays. However, whether the solutions found are truly optimal is unknown, since the method approximated the optimal solution rather than finding the exact optimal CPI.

Keywords: Radar, FMCW, Interference, Simulated Annealing, Coexistence, Non-uniform zero-augmented sampling

Acknowledgements

I want to express my sincerest appreciation and gratitude to my supervisors, Anders Mannesson, Daniel Sjöberg, Santosh Nadig, and Sebastian Heunisch, for their unwavering support, encouragement, and knowledge sharing, without which this thesis would not have been possible. Many questions have been asked, usually more than once, and each has been patiently answered in the most caring and educating way. I am incredibly lucky and grateful to have such wonderful family and friends who have constantly supported me, not only throughout this project but also every day leading up to it.

Popular Science Summary

Most people are familiar with bats and how they use echolocation to navigate and locate prey when hunting, a technique based on emitting high-frequency sounds and waiting for their reflection. The bat can perceive from what direction the reflected sound comes by comparing when its right versus left ear hears the sound, and by measuring the time between making the sound and hearing it again the bat can estimate the distance to some reflective object. Amazingly using this technique the bat can hunt prey the size of a mosquito in pitch-black environments.

However, what most people probably not have thought about is how a bat can make out the difference between a reflected sound emitted by itself, and a sound emitted by another bat.

Many range and location measuring techniques such as sonar, radar, and LiDAR work similarly, by transmitting some signal and waiting for its return. This means that, as in the case of this thesis, a radar will also face the same challenge of differing between a reflection and receiving the signal emitted from another radar transmitting the same signal. Both bats and radars can make out the difference by using slightly different frequencies and paces of emitting the signals. However, varying the transmitted signal in the radar case is not always a possible option. Another way of handling the problem, which is to make the bats or radars talk one at a time. In the radar case this means that the units takes turns of transmitting their signals. This means that some information will be lost when the radars are silent, and how much information is lost is governed by the order in which the radar take their turns of transmitting their signals. In essence this is a scheduling problem, which this thesis tries to solve by using an optimization method called Simulated Annealing. The method finds transmit codes according to which the radars do not interfere with each other while keeping as much information as possible.

Table of Contents

Acronyms

FMCW *Frequency Modulated Continous Wave*

CPI *Coherent Processing Interval*

AoA *Angle of Arrival*

EM *Electromagnetic*

RCS *Radar Cross Section*

TX *Transmitting Antenna*

RX *Receiving Antenna*

IF *Intermediate Frequency*

CW *Continous Wave*

ADC *Analog to Digital Converter*

LPF *Low Pass Filter*

SNR *Signal to Noise Ratio*

CFAR *Constant False Alarm Rate*

NUS *Non-Uniformly Sampled*

TSP *Traveling Salesman Problem*

SA *Simulated Annealing*

ULA *Uniform Linear Array*

NULA *Non-Uniform Linear Array*

CS *Correlation Sum*

PSR *Peak Sidelobe Ratio*

FWHM *Full Width at Half Maximum*

Chapter 1

Introduction

1.1 Background

Radar is a technique used for range, angle, and velocity measurements of targets, where the measurements are based on transmitting electromagnetic waves in the radio frequency spectrum and analyzing the reflected waves' properties. Radar is an acronym for "radio detection and ranging" which refers back to the basic properties of the technique, although radar has developed tremendously since the idea was first patented in 1903 [1]. Radar systems can be designed in many different ways depending on the purpose of the system, where different properties such as maximum measurable range and velocity can differ. For example, the detection range of a radar can vary from meters to kilometers [2]. Radars have historically mostly been used in military applications, with some exceptions such as navigation, sea, and air traffic control, and in meteorological applications [3][4]. Today radar is also used frequently in civil applications, such as automotive radar [5], measurements of filling levels in storage tanks [6], and medical applications [7]. Compared to other surveillance techniques such as cameras, radars can have the advantage of being unaffected by weather and light conditions, such as fog [8]. Additionally, radars can be used as a means of monitoring sensitive environments such as schools, hospitals, assisted living facilities, and public areas without intruding on people's privacy, since only location and velocity are measured [9]. Due to the increased usage of radar, the risk of mutual interference between radars operating at the same frequencies has increased as well. Mutual interference between radar sensors is classified as either coherent or incoherent, where the incoherent interference appears as noise, which degrades the detection ability of the radar. Coherent interference on the other hand expresses itself as appearing ghost-targets, which can be very difficult to differentiate from real targets. Hence, coherent interference, which occurs when radars transmit and receive identical signals at the same time, poses a larger challenge than incoherent interference [10][11]. The increased risk for interference has caused a rise in the demand for interference mitigation techniques and coexistence solutions [12].

One radar type commonly used in the automotive industry is the *Frequency Modulated Continuous Wave* (FMCW) radar, which is also used in this thesis [11] [10]. An FMCW radar operates by transmitting continuous frequency-modulated signals and receives the waves after reflection on some targets. FMCW radars only sample incoming signals in conjunction to also transmitting signals. One transmitted signal in the FMCW case is called a chirp, and a series of transmitted signals is called a chirp burst or a *coherent processing interval* (CPI). When the radar receives the reflected waves these are signal processed with the end goal of finding some spectral estimates from which range, velocity and *angle of arrival* (AoA) of some targets can be derived [13]. There are different methods of conducting these spectral estimates, where one of the most common methods to use is the Discrete Fourier Transform (DFT).

1.2 Research Problem

Assume two FMCW radars transmitting on the same frequencies are facing each other. The radars can cause mutual interference, even if they are placed well outside of each other's field of view, and hence degrade each other's detection abilities due to the possibility of appearing ghost targets resulting from coherent interference. One solution to this problem can be to allow for coexistence by letting the two radars consecutively transmit their CPIs, such that the radars never transmit their chirp bursts at the same time. This kind of solution is called time division multiplexing (TDM). This however might cause the time between the chirp bursts to increase, and hence impair the overall radar performance. One analogy to increasing the inter-CPI time is for example to decrease the frame rate in a film camera. Another possible solution to allow for coexistence without affecting the inter-chirp burst times is to make the chirp bursts sparse. This means that some of the transmitted chirps in a CPI are removed, rendering the CPI sparse. The CPIs can then be transmitted during the same times, given that two equal signals from the radars never are transmitted at the same time. In other words, the CPIs of the two radars are allowed to overlap in time and hence transmit CPIs during the same time interval, as long as two individual signals are never transmitted simultaneously. Because the FMCW radars only sample incoming signals when also transmitting, this design would not cause interference. Removing chirps from a CPI will however affect the performance of the radars as the quality of the radar measurements is worsened.

The problem of interest in this thesis is to investigate how coexistence between facing radars can be achieved when assuming a certain overlap in time of the CPIs and organizing the individual signals in the CPI in an optimal manner, such that interference is avoided and the quality of the measurements is affected negatively as little as possible. This problem will be investigated using standard techniques for spectral estimation, in essence the DFT.

1.3 Aim and Scope

This thesis aims at organizing radar transmit patterns such that radars operating at the same frequencies can coexist while transmitting at the same times, even if there is some time drift between the CPIs. This problem essentially consists of two parts, which are to both determine the optimal number of chirps to transmit in a chirp burst, and how to organize these such that the radar performance is optimal while also not causing interference.

To solve this problem the Simulated Annealing optimization algorithm is investigated and evaluated. The transmit patterns found are evaluated along with their implications on the measurement qualities. The thesis also compares using the Simulated Annealing method with using an existing similar method, namely the coprime method used to generate sparse arrays.

Chapter 2
Theory

In this section, the theory needed in this thesis is presented. The section begins with formulating a radar signal model, which is then used to explain some properties of the radar which depends on the design of the radar system. Then some theory on how the radar classifies a target from its measurements, that is detection, is presented. The chapter then continues with some information on non-uniformly sampled data and interference.

2.1 Introduction to Radar

Radar is, in its simplest form, a method of detecting objects and measuring the range of these by transmitting electromagnetic (EM) waves in the radio frequency (RF) spectrum and studying their reflections. Today, there are many radar implementations with different abilities depending on the purpose of the system, but they are all built on the same fundamental idea and subsystem. The fundamental idea is to transmit a signal x_T towards an area of interest and wait for its reflection x_R to return. For example, by using the time difference between transmitting and receiving the signal, the range to the target can be estimated. By analyzing changes in the waveform between the transmitted and received signal other properties of the target can be estimated, such as velocity and angle of arrival [14, p.4].

The range of a radar can vary from centimeters to hundreds of kilometers and is partially governed by the amount of power transmitted and later on received by the radar, which can be calculated using the radar range equation. The power density Q_1 of a transmitted wave at some range R away from the radar transmitting with power P_t is found as

$$Q_1 = \frac{P_t}{4\pi R^2} \tag{2.1}$$

Here it is assumed that the antenna is isotropic, meaning that the wave is dispersed in a spherical pattern around the radar [14, pp.61-63]. If the antenna is directional, meaning it has a transmitter gain G_t, the power is concentrated to a certain direction which reduces the dispersion of the power density, and the power density instead becomes

$$Q_1 = \frac{P_t G_t}{4\pi R^2} \tag{2.2}$$

Note that this is the power at range R without any reflections. Now introduce a target at range R, which will be exposed to this transmitted signal with power density Q_1. The signal is again dispersed upon reflection on the target, and the power reflected depends on the target's radar cross section (RCS) σ. The power received by a radar with receiver gain G_r after reflection is found as

$$P_r = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 R^4} \tag{2.3}$$

This formula is called the Radar Range Equation [14, pp.61-63].

The fundamental subsystem in a radar includes a transmitter, receiver, antenna and signal processor. The transmitter provides the antenna with an RF signal which excites the antenna, making it transmit waves with the same frequency as the current. The basic components of the transmitter is a power supply, whose current is altered with a radio frequency by a modulator. This current is then amplified by an RF amplifier before it reaches the transmitting antenna (TX). The resulting RF wave travels to the target where it is reflected and eventually reaches the receiving antennas (RX) where the wave is again converted into a current which is then passed on to the receiver, where it is down-converted to an intermediate frequency (IF) signal by a component called mixer. This signal is then converted from analog to digital and can then be processed to find for example the range and velocity of a target [14, p.4].

The transmitted wave can be either pulsed or continuous, where the pulse radar transmits wave pulses periodically and determines the range to the target by measuring the elapsed time between transmitting and receiving the signal. A continuous wave (CW) radar transmits a signal continuously. An ordinary non-modulated CW radar can be used for target detection, but not range estimation. To enable the CW radar to measure range, the wave has to be modulated, either by frequency or phase, yielding the acronyms FMCW-Frequency Modulated Continuous Wave, and PMCW-Phase Modulated Continuous Wave [14, p.20].

The inner workings of a radar are of course far more complex than described here, but it suffices for a short introduction. Most components of a radar will not be discussed further since they are out of the scope of the thesis, however, some parts of the receiver will be discussed in more detail.

2.2 CW Radar Signal Model

Consider a continuous wave radar with one RX and TX antenna, and assume this radar is transmitting a signal parametrized by frequency f_0, time t, phase ϕ_0 and amplitude A_1. This transmitted signal can be described as

$$x_{\mathrm{T}}(t) = A_1 \cos(2\pi f_0 t + \phi_0) \tag{2.4}$$

Now introducing a stationary reflective target at a range R a received signal $x_R(t)$ with the same shape as the transmitted signal is expected but with a time delay

$\tau = \frac{2R}{c}$, due to signal propagation, and amplitude A_2, where $A_1 > A_2$ due to the energy loss at reflection and dispersion [14, p.287].

$$x_{\mathrm{R}}(t) = A_2 \cos(2\pi f_0(t - \tau) + \phi_0) \tag{2.5}$$

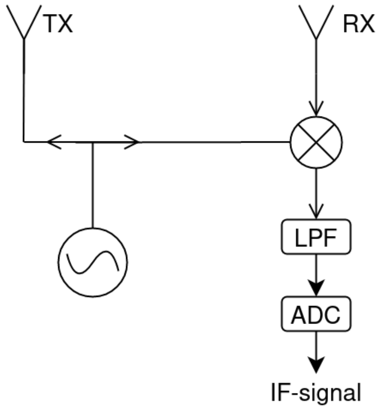


Figure 2.1: Schematics of the components in a generic receiver where the received signal is mixed with the transmitted signal. Here, LPF is an acronym for Low Pass Filter, and ADC for Analog to Digital Converter. The wave symbolizes the frequency chirp generator present in the FMCW case [10].

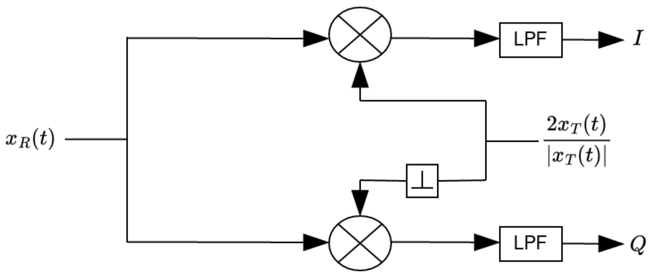


Figure 2.2: Explanatory schematics of a Coherent I/Q Receiver [15, p.14].

After reflection, the wave reaches the RX antennas and propagates to the receiver as an RF wave. The main purpose of the receiver is to down-convert the received signal into an IF signal, which is also needed to enable for an analog-to-digital converter (ADC) to sample the signal. The IF-signal can denote a signal located in frequency between the baseband and carrier frequencies, found by mixing the received signal with some Local Oscillating (LO) signal. However, in this work a homodyne receiver is used, meaning that the IF signal is found by mixing the received signal directly with the transmitted signal. Hence, the IF signal is identical with the baseband signal and the terms are used synonymously in this work. The receiver also applies filtering to clean the signal from noise, as well as a low-noise amplifier and a low-pass filter which excludes signals not belonging to the baseband [14, p.391] [16, pp.27-28].

A receiver can be implemented in many ways, for example as shown in Figure 2.2, which shows a simple schematic of a coherent In-phase/Quadrature (I/Q) receiver. The advantage of this design is its ability to preserve phase information in the down-converted signal [14, p.288]. As the received and propagated signal x_R enters the I/Q receiver, it is mixed with both the transmitted signal and the transmitted signal's orthogonal counterpart, and then passed to a low pass filter (LPF), forming the I and Q pair. According to this receiver design $x_R(t)$ is mixed with $x_T(t)$ and the orthogonal $x_{T\perp}(t) = -A_1 \sin(2\pi f_0 t + \phi_0)$, which gives

$$\begin{aligned} x_{\mathrm{R}}(t) \cdot \frac{2x_T(t)}{A_1} &= A_2 \cos(-2\pi f_0 \tau) + A_2 \cos(2\pi f_0(2t - \tau) + 2\phi_0) \\ x_{\mathrm{R}}(t) \cdot \frac{2x_{T\perp}(t)}{A_1} &= -A_2 \sin(-2\pi f_0 \tau) - A_2 \sin(2\pi f_0(2t + \tau) + 2\phi_0) \end{aligned}$$

Filtering of these signals through the LPF yields

$$\begin{aligned} I(t) &= A_2 \cos(2\pi f_0 \tau) \\ Q(t) &= A_2 \sin(2\pi f_0 \tau) \end{aligned}$$

It is seen that the design of the radar wave receiver gives two orthogonal waves which then can be used to construct the analytic signal

$$a(t) = I(t) + \mathrm{j}Q(t) = A_2 \mathrm{e}^{-\mathrm{j}2\pi f_0 \tau} = A_2 \mathrm{e}^{-\mathrm{j}4\pi f_0 \frac{R}{c}} \tag{2.6}$$

where $\tau = 2R/c$ is used in the last step.

2.3 FMCW Radar Signal Model

Having the CW Radar model, the next step is to find an expression for the Frequency Modulated CW (FMCW) radar signal. This can be achieved by sweeping the frequency linearly during a time interval T_c from f_0 to f_1. The instantaneous frequency $f(t)$ can be defined using the bandwidth $B = f_1 - f_0$ as

$$f(t) = f_0 + \frac{B}{T_c} t \quad , 0 \leq t \leq T_c \tag{2.7}$$

To express the wave as a cosine the instantaneous phase is used, which is found by integrating the angular frequency as

$$\phi(t_s) = 2\pi \int_0^{t_s} f(t) \mathrm{d}t = 2\pi \left[f_0 t + \frac{B}{2T_c} t^2 \right]_0^{t_s} + \phi_0 = 2\pi (f_0 t_s + \frac{B}{2T_c} t_s^2) + \phi_0 \tag{2.8}$$

Using this instantaneous phase $\phi(t_s)$ in the cosine function gives the frequency-modulated signal called a chirp, of which an example can be seen in Figure 2.3 along with the instantaneous frequency. As such, the transmitted and the received signal in the FMCW case can be expressed by substituting $2\pi f_0 t$ with $\phi(t)$ in (2.4) and (2.5). Inspired by [16, pp. 26-28] and using the previous result the signals can be described as

$$x_T(t) = A_1 \cos\left(2\pi f_0 t_s + \pi \frac{B}{T_c} t_s^2 + \phi_0\right) \tag{2.9}$$

$$x_R(t) = A_1 \cos\left(2\pi f_0 (t_s - \tau) + \pi \frac{B}{T_c} (t_s - \tau)^2 + \phi_0\right) \tag{2.10}$$

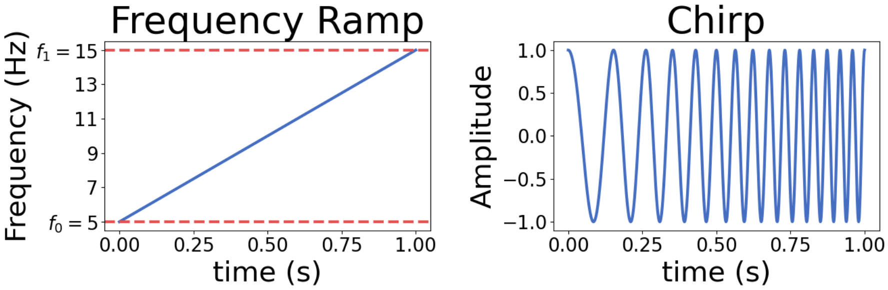


Figure 2.3: Figure displaying an example of a linearly increased instantaneous frequency (left) with $(f_0, f_1) = (5Hz, 15Hz)$ and the resulting chirp (right), similar to the figures in [15, p.30]. The red dashed line emphasizes the limits of the bandwidth B.

The received signal $x_R(t)$ is, just as in the CW case, passed through the coherent I/Q detector, yielding the range-dependent IF signal

$$a(R) = A_2 \mathrm{e}^{-\mathrm{j}4\pi(f_0 + \frac{B}{T_c} t_s)\frac{R}{c}} \tag{2.11}$$

again assuming distance R such that $\tau = \frac{2R}{c}$ [16, p. 29]. In this expression, it is seen that the detector takes two frequency-modulated signals as input, and outputs a constant frequency IF-signal, with a frequency dependent on the range R and hence also τ. Let us denote this constant IF-frequency, also called beat frequency, f_{IF} [13]. An intuitive graphic of this is shown in Figure 2.4.

To allow for digital signal processing the IF-signal is sampled by an ADC. Assuming that the signal is sampled N times, such that $\Delta t_s = \frac{T_c}{N}$ and $f_s = \frac{N}{T_c}$, the discrete time points $t_n = n\Delta t_s$ are acquired, where $n = 0, 1, ..., N-1$ [14, p.409]. Substituting t_s to t_n in (2.11) and expanding gives the discrete chirp frequency

$$f(n) = f_0 + \frac{B}{N} n$$

and the discretized IF signal as

$$a_n(R) = A_2 \mathrm{e}^{-\mathrm{j}4\pi(f_0 + \frac{B}{N} n)\frac{R}{c}} \tag{2.12}$$

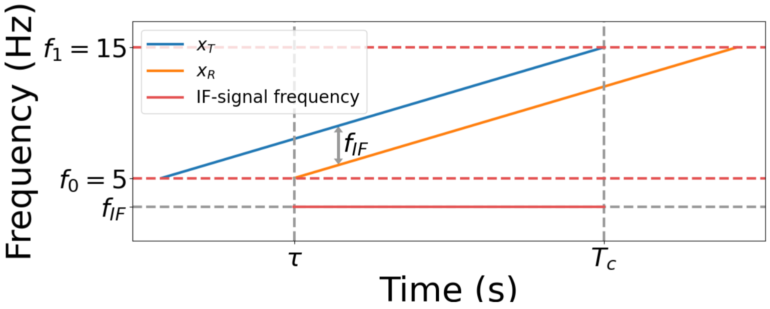


Figure 2.4: Figure displaying an example of the frequency components of a transmitted chirp x_T and its time-delayed ditto x_R. The dashed lines again show the limits of B. The gray lines show the time delay τ, chirp time T_c and the frequency of the IF-signal (solid red line) f_{IF}, which is equal to the vertical distance between the chirps. The figure is inspired by the figures in [13].

In essence this means that the IF signal, a constant frequency sinusoid, is sampled with sampling frequency f_s. According to the Nyquist-Shannon sampling theorem, any signal containing frequencies below the Nyquist frequency $f_n = \frac{f_s}{2}$ can be reconstructed, while signals including frequencies equal to or larger than the Nyquist frequency will be aliased [17, p.30]. To avoid aliasing, a lowpass filter with a cutoff frequency equal to $\pm f_n$ is applied to the analog signal before sampling, to exclude any frequencies $f \geq |f_n|$. This filter is displayed in the receiver schematic in Figure 2.2. This means that the Nyquist frequency f_n, governed by the ADC components sampling frequency, will put a bound on the IF signal's frequency to a frequency band $f_B = [-f_n, f_n]$. The IF signal's frequency on the other hand is governed by the time difference τ between transmitting and receiving the chirp. Hence, the frequency band limiting the IF signal implies a time interval τ_l in which the received signal is sampled. In essence, this gives a time-frequency sampling region B_s, parallel to the transmitted chirp, described in the frequency domain as $B_s = [f_0 + \frac{B}{T_c}(t + \tau_l), f_0 + \frac{B}{T_c}(t - \tau_l)]$ in which received signals are sampled. Some explanatory figures can be seen in Figure 2.5.

This far one stationary target at some range R has been considered. This model is now extended to allow the target to have a velocity v. Define the radial velocity v as positive if the target moves radially away from the radar. Redefining the distance R and substituting into (2.12) gives the new expression for the analytic IF signal as

$$R_n = R_0 + vn\Delta t_s$$
$$a_n(R_0, v) = A_2 \mathrm{e}^{-\mathrm{j}4\pi(f_0 + \frac{B}{N}n)\frac{R_0 + vn\Delta t_s}{c}}$$

When using a saw tooth chirping pattern the velocity of the target cannot be determined by only one chirp, but has to be found as the phase difference between

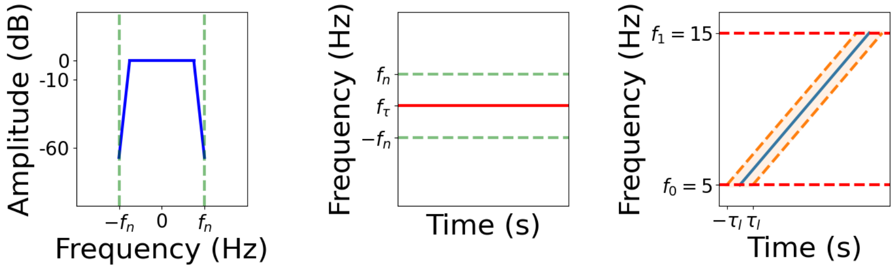


Figure 2.5: Leftmost figure displaying an example of a gain-magnitude frequency response (blue line) of the lowpass filter suppressing frequencies larger than f_n. The middle figure displays the constant IF-frequency (red line) along with the filter implied bound (green lines). Rightmost figure displays the frequency band B_s (orange area) around the frequency content of the transmitted signal x_T (blue line) in which received signals are sampled.

the reflection of multiple chirps [13].

A Coherent Processing Interval (CPI), also called a chirp burst, is a series of M consecutive chirps, each with sweep time T_c during the time interval $[0, T_f]$ where the total time of the CPI is $T_f = M \cdot T_c$ [14, p.548]. Assuming the chirps are placed uniformly without any idle time in between, the time between two chirps' start times will equal T_c, and the start time for some chirp m will be $m \cdot T_c$, where $m = 0, 1, ..., M - 1$.

The range to the target R can be redefined as

$$R_{nm} = R_0 + (mT_c + n\Delta t_s)v$$

and the resulting IF-signal from chirp m at sample n reflected from a target at range R_0 with velocity v can be described as

$$\begin{gathered} a_{nm}(R_0, v) = \\ A_2 \mathrm{e}^{-\mathrm{j}4\pi f_0 \frac{R_0}{c}} \mathrm{e}^{-\mathrm{j}mT_c 2\pi(\frac{2vf_0}{c})} \mathrm{e}^{-\mathrm{j}n2\pi(\frac{2vBmT_c}{Nc} + \frac{2BR_0}{Nc} + \frac{2vf_0\Delta t_s}{c})} \mathrm{e}^{\mathrm{j}2\pi \frac{n^2 2Bv\Delta t_s}{Nc}} \end{gathered} \tag{2.13}$$

where the definition $t_n = n\Delta t_s$ have been used. The last term is a chirping effect which can be neglected due to the assumed relatively slow speed of the target.

In the case of having multiple objects, the transmitted signal will be reflected on each target and the received and processed signal can be described as the superposition of (2.13). In the frequency domain, this corresponds to multiple IF-tones representing the ranges of the targets [13].

2.3.1 Radar Data Cube

From the previous section and (2.13) it is known how to describe the IF-signal resulting from the reflection on a target at an initial range R_0 and with velocity v at any sample n in any chirp m. A radar usually transmits a chirp burst or CPI, consisting of M chirps where each chirp is sampled N times. The data from one CPI is stored in a matrix structure with the samples along the row indices, and the chirp number along the column indices. The structure of the matrix is shown below in Figure 2.6. Let us denote this matrix $\mathbf{A}(R_0, v)$, since the matrix still is a function of a target's range and velocity. Now let us introduce G targets, each with initial radial range R_i and velocity v_i where $1 \leq i \leq G$. The IF-signal resulting from M chirps being reflected at G targets is found as the superposition of the received signals, this is

$$\mathbf{C} = \sum_{i=1}^{G} \mathbf{A}(R_i, v_i) \tag{2.14}$$

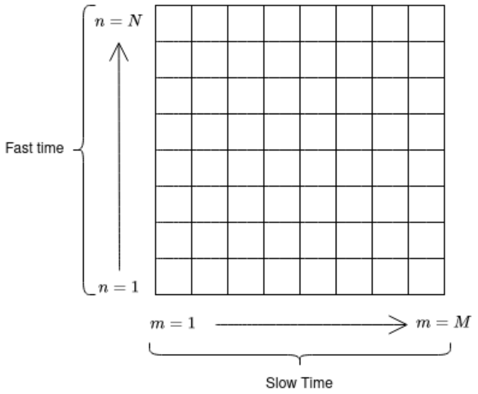


Figure 2.6: Example of small radar data matrix.

Now, $\mathbf{C}$ denotes the matrix found as the superposition of all radar data matrices resulting from each single target, for an arbitrary number of targets. To this point, the model used has only included one TX and one RX antenna, which yields this matrix when transmitting a chirp burst. To enable for estimation of the AoA of the targets more antennas have to be included. When including more antennas the matrix expands to a cube where the added dimension is commonly referred to as the channel dimension. This cube is called the Radar Data Cube [14, pp.291,502-503], and from it the range, velocity and angle of arrival (AoA) of a target can be found [18, pp. 16-17].

2.4 Spectral Estimates

By analyzing the spectral properties of the radar data cube in different dimensions the range, velocity and angle to the targets can be estimated. In the following, the range and velocity estimates will be explained, while assuming for simplicity that the radar only illuminates one target at an initial range R_0 with constant velocity v. The spectral analysis will be conducted using the Fourier transform. The transform will be applied to the discretized signal given by (2.13), and hence the Discrete Time Fourier Transform (DTFT) is used. Applying the DTFT to a sequence x_n will in the following be denoted $\mathcal{F}[x_n]$. In practice, the DTFT is implemented using the Fast Fourier Transform (FFT) implementation of the Discrete Fourier Transform (DFT) [17, pp.129,153].

2.4.1 Windowing

When using the DFT, spectral leakage will occur, which is a smearing of the energy content in one frequency bin over more frequency bins. To compensate for this leakage, windowing can be used, which is a technique performed by multiplying the signal with a window function in the time domain before applying the DFT to it. Many windows with different properties exist, where Hann window is a common choice [17, pp71-85.]. The implication of applying a Hann window to a signal, compared to using the rectangular window, in the time domain is a slightly wider mainlobe in the frequency domain, while also getting less spectral leakage and smaller sidelobes [15, p.183] [19, pp.363-366]. This is equivalent to getting a larger amplitude difference between the mainlobe and the highest sidelobe peak, also called Peak Sidelobe Level (PSL).

2.4.2 Range estimate

The range estimation is done by examining the frequency difference between the transmitted and received signals. Since the signal is mixed, it suffices to find the frequency content of the IF signal, this is the beat frequencies, stored in the matrix $\mathbf{C}$. Returning to the radar data matrix, it is known that the IF-signals from each chirp are stored as columns in $\mathbf{C}$, and hence by finding the frequency content of each column, the range can be estimated. For simplicity only one target is concerned and hence $\mathbf{C} = \mathbf{A}(R_0, v)$. The DTFT is applied to each column, and hence only one chirp in fast time is considered. Returning to (2.13) this means that m can be regarded as constant, and this expression for the IF-signal can be used to describe the signal values along one column in the radar data matrix. Since m now is constant the expression of the signal can be divided into one constant part $\gamma_m(R_0, v)$ and one variable part, where the constant part of (2.13) is

$$\gamma_m(R_0, v) = A_2 \mathrm{e}^{-\mathrm{j}4\pi f_0 \frac{R_0}{c}} \mathrm{e}^{-\mathrm{j}mT_c 2\pi(\frac{2vf_0}{c})}$$

and neglecting the last term in (2.13) gives

$$a_{nm}(R_0, v) = \gamma_m(R_0, v) \mathrm{e}^{-\mathrm{j}n2\pi(\frac{2vBmT_c}{Nc} + \frac{2BR_0}{Nc} + \frac{2vf_0\Delta t_s}{c})}$$

This expression then corresponds to the IF signal of one chirp, or one fast time column in the radar data cube. To find the frequency of the signal, a partial Discrete Time Fourier Transform along the fast time n indices is used. The partial DTFT with respect to the sample number n, that is $\mathcal{F}_n[a_{nm}(R_0, v)]$, will then result in a peak at frequency bin $\hat{f}_n$ corresponding to the frequency

$$\hat{f}_n = \frac{2vBmT_c}{Nc} + \frac{2BR_0}{Nc} + \frac{2vf_0\Delta t_s}{c}$$

The last term in this expression will be negligible compared with the first two, meaning that the chirp-dependent range R_m can be estimated as

$$R_m = R_0 + vmT_c = \frac{\hat{f}_n Nc}{2B}$$

This process is then used to find the radial range information of each chirp, along each column in the radar data cube [13, 14, p.4, p.645].

2.4.3 Velocity Estimate

For the velocity estimate the partial DTFT of the rows in the radar data matrix is considered instead, thus regarding the sample number n as constant. When the target moves the range will change, but due to the fast chirping the range difference between two chirps will be too small to detect. The velocity on the other hand will have a significant effect on the phase of the incoming wave due to the short wave length. Hence, by measuring the phase difference between each chirp the velocity of a target can be found, and to this end the partial DTFT along the slow time axis is used. Again returning to (2.13), regarding n as constant and ignoring the last term allows us to write

$$\gamma_n(R_0, v) = A_2 \mathrm{e}^{-\mathrm{j}4\pi f_0 \frac{R_0}{c}} \mathrm{e}^{-\mathrm{j}n2\pi(\frac{2BR_0}{Nc} + \frac{2vf_0\Delta t_s}{c})}$$

$$a_{nm}(R_0, v) = \gamma_n(R_0, v)\mathrm{e}^{-\mathrm{j}2\pi m(\frac{2vf_0T_c}{c} + \frac{2vBnT_c}{Nc})}$$

Again applying the partial DTFT but this time along the slow time axis, $\mathcal{F}_m[a_{nm}(R_0, v)]$, gives a frequency peak located at the frequency bin $\hat{f}_m$

$$\hat{f}_m = \frac{2vf_0T_c}{c} + \frac{2vBnT_c}{Nc}$$

The second term will be comparably small and the frequency can be estimated as

$$\hat{f}_m = \frac{2vf_0T_c}{c} \tag{2.15}$$

and find the velocity as

$$v = \frac{\hat{f}_m \lambda}{2T_c} \tag{2.16}$$

where $\lambda = \frac{c}{f_0}$.

The partial DTFT in slow time gives the radial velocity information along one fast time sample, and thus this process is again carried out along the slow time axis at each fast time index in the radar data cube [13, p.5].

As a final note on the estimates, observe that if several targets would be regarded the linearity property of the DTFT would be used, which would give one spectral estimate per target. The simplification of only using one target in the previous sections implies no loss of generality since if several targets would be introduced the sequences analyzed would be the superposition of many IF-signals, and the DFT would instead become a sum of DFTs. The end estimate would be a sum of these frequency-valued sequences, in which we would get one peak per target, regardless of the dimension analyzed.

2.4.4 Range Doppler Map

From the fast-time and slow-time DTFTs the range and velocity of some targets can be estimated, along with the AoA estimated from the DTFT along the channel indices. Hence a DTFT is applied to each dimension of the radar data cube resulting in a similar cube in which instead of chirp, sample and channel index are partitioned into frequency bins. The fast time and slow time indices are replaced with range bin and velocity bin indices respectively.

The range and velocity information is commonly combined into a Range-Doppler Map. This map is a matrix with the previously mentioned range bins as rows and velocity bins as columns. A target will in the Range-Doppler map appear as a dot where its distance from the center line shows its velocity, and its distance from the bottom of the map shows its range. An example of a Range-Doppler map of one target is shown in Figure 2.7.

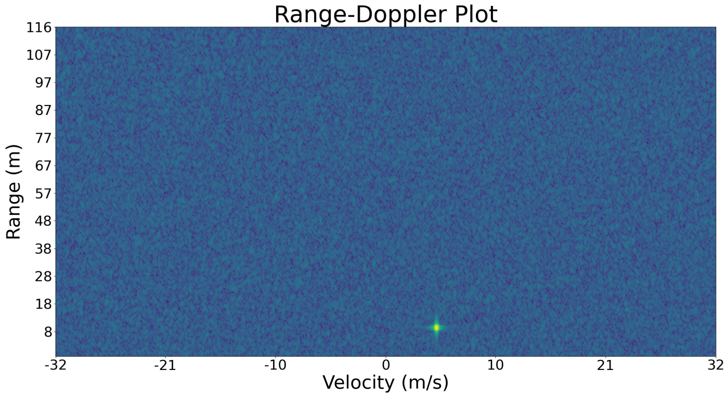


Figure 2.7: Figure displaying an example of a Range Doppler Plot of a simulated target having radial velocity 5 m/s and radial distance 10m. The simulation includes white Gaussian noise.

2.4.5 Maximum Velocity

From the previous sections, it is known that the range can be measured using the DFT along the fast time axis, and the velocity as the DFT along the slow time axis. During the time T_f, in which the CPI is transmitted, the target's position will not change, due to its relatively small velocity. Hence, during a CPI the target's movement will be negligible, and therefore the DFT of each chirp in fast time will map the target to the same range bin in the Range-Doppler map, despite having this velocity. However, each IF-signal stored in the radar data matrix will have a relative phase difference due to the velocity, as explained before. By applying a partial DFT along the slow time axis this phase difference can be measured as the frequency at the resulting spectral peak $\hat{f}_m$.

From the Nyquist criterion, [17, p.30] it is known that aliasing of the frequency spectrum will occur for any frequencies outside the frequency interval $[-f_n, f_n]$ where $f_n = \frac{f_s}{2}$, or more formal $|\hat{f}_m| < f_n$. The Nyquist critical frequency of a signal sampled uniformly with a grid size T_g is given by

$$f_n = \frac{1}{2T_g} \tag{2.17}$$

according to Bretthorst in [20, p.3]. Thus, the normalized frequency range in which $\hat{f}_m$ can appear is in $[-\frac{1}{2}, \frac{1}{2}]$, and hence the maximum frequency measurable is $\frac{1}{2}$. Allowing the radar to measure frequencies outside of this range allows for ambiguities due to aliasing according to the Nyquist criterion [17, p.30].

Inserting the maximum measurable frequency $\hat{f}_m = \frac{1}{2}$ into (2.16) gives the maximum unambigous velocity as

$$v_{max} = \frac{\lambda}{4T_g} \tag{2.18}$$

In the context of this thesis, each chirp is regarded as one sample in slow time, and hence T_g equals the time difference between two chirps' start times. Since it was assumed that there is no idle time between chirps, this means that

$$v_{max} = \frac{\lambda}{4T_c} \tag{2.19}$$

describes the maximum velocity measurable by the radar, where T_c again is the chirp time [13].

2.4.6 Velocity Resolution

A DFT of some discrete signal s_n, where 0 can resolve two separate frequency peaks $\hat{f}_1$ and $\hat{f}_2$ if their frequency difference $\Delta\hat{f} = \hat{f}_1 - \hat{f}_2$ fulfills

$$\Delta\hat{f} > \frac{1}{L} \tag{2.20}$$

with L being the number of sample points in the signal s_n [21, p.31]. Having two targets with velocities v_1 and v_2, and their corresponding frequency peaks at $\hat{f}_1$ and $\hat{f}_2$ respectively, their velocity difference is defined as $\Delta v = v_1 - v_2$. Using equation (2.16) gives

$$\Delta v = v_1 - v_2 = \frac{\hat{f}_1 \lambda}{2T_c} - \frac{\hat{f}_2 \lambda}{2T_c} = \frac{\Delta \hat{f} \lambda}{2T_c} \tag{2.21}$$

from which $\Delta \hat{f}$ can be found as

$$\Delta \hat{f} = \frac{\Delta v \cdot 2T_c}{\lambda} \tag{2.22}$$

In the case of applying the DFT to the data cube along the slow time indices L in (2.20) is equal to M. Insertion of (2.22) into (2.20) gives

$$\begin{aligned} \Delta \hat{f} &> \frac{1}{M} \iff \\ \frac{\Delta v \cdot 2T_c}{\lambda} &> \frac{1}{M} \iff \\ \Delta v &> \frac{\lambda}{2T_c M} \end{aligned}$$

Using the previously defined $T_f = MT_c$ helps us find the expression for the minimum resolvable velocity difference as

$$v_{res} = \frac{\lambda}{2T_f} \tag{2.23}$$

which shows that velocity resolution depends on the length of the CPI exclusively [13, p.5].

2.5 Detection Thresholds

When the signal has been processed into the Range-Doppler map, the issue of separating targets from noise and clutter in the signal remains. To this end a method called Constant False Alarm Rate can be employed, but before presenting this method some additional theory is needed.

2.5.1 Thermal Noise

The signal received by the radar will be affected by various types of noise, amongst others thermal noise. The power of this noise P_n is given by

$$P_n = kT_0FB \tag{2.24}$$

Where k is Boltzmann's constant, $T_0 = 290\,K$ which is the standard room temperature in Kelvin, F is the receiver noise figure and B is the instantaneous receiver bandwidth [14, p.65].

2.5.2 Signal to Noise Ratio

From the introduction section, it is known that the power received from a reflected signal is described by the radar range equation and denoted P_r. The ratio of the received signal power and the noise power is called *Signal to Noise Ratio* (SNR). This ratio is defined as

$$\mathrm{SNR} = \frac{P_r}{P_n} = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 F B} \tag{2.25}$$

Where k, T_0, F and B are defined as in the thermal noise case. G_t and G_r denote the transmitter and receiver gain respectively, λ and σ denote the carrier wavelength and mean RCS of a target, and P_t is the peak transmitted power [14, pp. 64-66].

2.5.3 Dynamic Range

The Dynamic Range of some quantity Q is defined as

$$\text{Dynamic Range} = 20 \cdot \log_{10}\left(\frac{Q_{max}}{Q_{min}}\right) \tag{2.26}$$

In the radar setting the quantity Q can for example be the velocity spectrum resulting from the slow time partial DFT, and then Q_{max} corresponds to the peak value of the main lobe. To avoid ambiguities in the measurements only amplitudes as low as the value of the largest side lobe can be measured, this would be Q_{min}. Hence, the dynamic range is the distance between the mainlobe and the largest sidelobe in dB scale [22, p.77].

2.5.4 Constant False Alarm Rate

Generally, detection is carried out by comparing spectral peaks with some threshold, where any peak larger than the threshold is considered a target. The question of how to determine this threshold arises, and one common way is by Ordered Statistics Constant False Alarm Rate (OS-CFAR). This method uses a window that is swept over the signal. Each amplitude in the signal included in the window is ordered by amplitude in ascending order. The threshold is then set by choosing an amplitude in this ordered sequence at a predefined index, and multiplying this amplitude by a scaling factor [23, pp.3-6]. The signal which the OS-CFAR is applied to results from the superposition of noise, clutter and the spectrum given by the reflection on the target. As such the magnitude of this threshold will be affected by the SNR and Dynamic Range. Hence, having two targets with different signal strengths, a small Dynamic Range or SNR will result in a higher probability that the weaker target peak is located below the detection threshold, and will hence not be detected by the radar [15, p.4].

2.5.5 Integration Gain

Coherent integration means that both the amplitude and the phase of some received signals are used in the signal processing, with one example of a coherent integration method being the DFT. Using coherent integration on a CPI of chirps gives rise to *integration gain*, in essence the SNR is increased when more chirps are included in the CPI. Denoting the SNR of one chirp SNR_1, the SNR of the CPI can be described as $\mathrm{SNR}_{\mathrm{CPI}} = m \cdot \mathrm{SNR}_1$, where m is the number of chirps in the CPI [14, p.67].

2.6 Non-Uniform CPIs

In the previous sections, it is assumed that each received chirp is uniformly sampled and that the chirps are uniformly transmitted. Further on it will be useful to be able to chirp non-uniformly.

Having a uniform CPI of chirps, a non-uniform sampled (NUS) zero-augmented chirp burst can be found by multiplying the uniform burst by a sum of boxcar functions, which then replaces some specified chirps with 0's. This idea is depicted in Figure 2.8. With the Radar Data Square in mind, it is easy to realize that setting one chirp to equal zero at each sample is equal to setting the column corresponding to the chirp to zero. Defining a set S containing the slow time start-indices of the chirps to be transmitted, this is $S \in [1, M]$, a row vector with 1's at the indices specified in S can be described as

$$\Omega = \begin{pmatrix} \delta_{1,S} & \delta_{2,S} & \delta_{3,S} & \dots & \delta_{M,S} \end{pmatrix} \tag{2.27}$$

where δ denotes the Kronecker delta function defined in [24, p.10] as

$$\delta_{t,i} = \begin{cases} 1, & \text{if } t = i \\ 0, & \text{if } t \neq i \end{cases} . \tag{2.28}$$

With the radar data matrix $\mathbf{C}$ found in (2.14), Ω and the $(1 \times N)$ vector $\mathbb{1}_{1,N} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$ the NUS zero-augmented radar data matrix can be found as

$$\mathbf{C}_{NUS} = \mathbf{C} \odot (\mathbb{1}_{1,N}^T \cdot \Omega) \tag{2.29}$$

with $\odot$ denoting the Hadamard (element-wise) product. This would then make each column with index not included in S equal to zero. This is a way of choosing sample points from a uniform grid in a possibly non-uniform way, and as such this is commonly called on-grid sampling [25, p.9].

2.7 Spectral Estimates For Non-Uniform CPIs

Since the chirping pattern no longer has to be uniform, the velocity estimate has to be revised. The range estimate however only depends on the fast time axis of

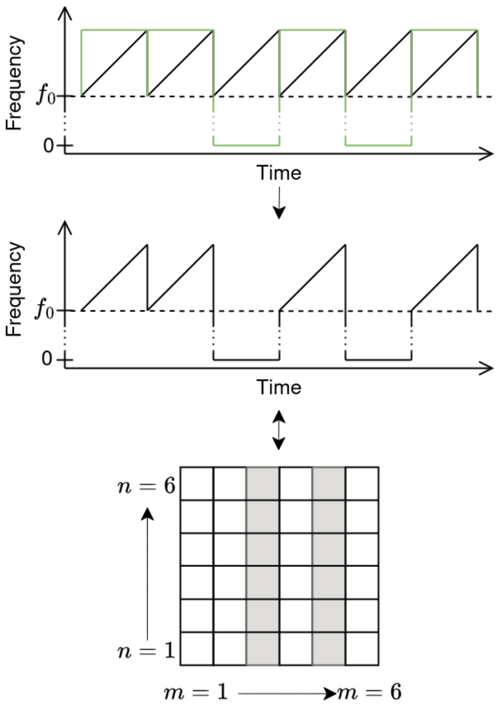


Figure 2.8: Figure depicting an example of the idea of using boxcar functions to cancel out specified chirps. The green line in the upper figure displays the boxcar sum. When this sum is multiplied by the uniform chirp burst the result is a non-uniform chirp burst shown in the middle plot. The bottom plot depicts the NUS zero augmented data matrix after Hadamard multiplication with Ω, where the gray areas represents 0-valued entries.

the radar data matrix, which remains unchanged, and hence the range estimation is still valid.

As found before, the NUS zero-augmented radar data matrix is described as

$$\mathbf{C}_{NUS} = \mathbf{C} \odot (\mathbb{1}_{1,N}^{T} \cdot \Omega)$$

and regarding only one target at initial range R_0 and with velocity v this reduces to

$$\mathbf{C}_{NUS} = \mathbf{A}(R_0, v) \odot (\mathbb{1}_{1,N}^{T} \cdot \Omega) \tag{2.30}$$

which we want to Fourier transform along the slow time m axis, following the method of finding the velocity when having the ordinary uniformly sampled radar data cube. The DTFT is performed along the m axis row-wise, and hence letting the row index be constant, $n = k$, the radar data matrix in (2.30) corresponding to one target reduces to a row vector $\overline{\mathbf{c}}_{NUS} = \mathbf{A}_{k,m}(R_0, v) \odot \Omega$.

Applying the DTFT to this row vector yields

$$\mathcal{F}_M[\overline{\mathbf{c}}_{NUS}] = \mathcal{F}[\mathbf{A}_{k,m}(R_0, v) \odot \Omega] = \frac{1}{M}\mathcal{F}[\mathbf{A}_{k,m}(R_0, v)] \circledast \mathcal{F}[\Omega]$$

where the convolution theorem has been used and ($\circledast$) denotes the circular convolution. The DTFT of Ω is commonly called the point-spread function and Ω is called the sampling vector [25].

Using the same notation as in the uniform chirping case, the expected frequency peak of the NUS data will remain at the same frequency bin as in the uniform sampled data case $\hat{f}_m$. If the set S does not contain the indices of all possible chirps, that is the sequence contains some 0 elements, the convolution will result in an increased amplitude of the side lobes. Due to the Hadamard multiplication of the sampling vector Ω and the uniformly sampled row vector $\mathbf{A}_{k,m}(R_0, v)$, the actual time the signal is sampled will depend on the placings of the elements in Ω. A sampling vector Ω with 0's at the end and start position will effectively yield fewer samples, where the first and last samples are sampled with a shorter time in between. Hence such a sampling vector would effectively decrease the sampling time.

2.7.1 NUS Velocity Resolution

From (2.23) it is known that the velocity resolution is dependent on the CPI time T_f. When introducing the NUS zero-augmented signal the CPI time might be shortened, for example, if the first or last chirp is canceled out. This way, the velocity resolution will instead depend on the indices of the chirps stored in S. The time difference between the first and last chirp can be found as

$$T_{NUS} = \Delta t_c \cdot (\max(S) - \min(S)) \tag{2.31}$$

and insert this into the velocity resolution expression (2.23), and hence the NUS velocity resolution, not too different from the uniform velocity resolution, is found as

$$v_{res,NUS} = \frac{\lambda}{2T_{NUS}} \tag{2.32}$$

The effect a shorter sampling time has on the frequency spectrum is known as reciprocal spreading, where a signal compressed in the time domain results in a widened peak in the frequency domain [15, p.25].

2.7.2 NUS Maximum Velocity

In the uniform CPI case, it was shown that the necessary bound for the maximum velocity to avoid aliasing was given by $|\hat{f}_m| < f_n$ and normalizing by the Nyquist critical frequency f_n gives $|2\pi\hat{f}_m| < \pi$ where $\hat{f}_m$ is the spectral peak resulting from the partial slow-time DTFT along the slow time axis. For reference see the section “2.4.5 Maximum Velocity”.

Now in the non-uniform CPI case, this has to be revised since the Nyquist critical frequency f_n is no longer valid. Consider a sampled complex signal, where the Imaginary and Real parts of the complex signal do not have to be sampled at the same times (non-simultaneous) and the samples are non-uniform, then according to Bretthorst the Nyquist critical frequency f_{NUS} of a non-uniform, non simultaneously sampled signal is instead given by

$$f_{NUS} = \frac{1}{2\Delta t_{NUS}} \tag{2.33}$$

where Δt_{NUS} is the largest possible grid size such that each sample (both Imaginary and Real) is located at an integer multiple of Δt_{NUS} [20, pp.7, 16-18]. In this context, we do not have non-simultaneous sampled data, but this does not make any difference to the expression in (2.33), which still is valid.

Now considering the spectrum resulting from a DTFT of this NUS signal, aliasing will occur for frequencies $\hat{f}_m$ outside the bound $|\hat{f}_m| < f_{NUS}$. Again dividing by f_n found in (2.17) and multiplying by π gives the normalized spectrum. This yields

$$|2\pi\hat{f}_m| < \frac{f_{NUS}}{f_n} \cdot \pi = \frac{\Delta t_{NUS}}{\Delta t_c} \cdot \pi \tag{2.34}$$

which also agrees with the result presented in [26, pp.34-35].

Since the frequency peak of the DTFT on the NUS sequence in slow time will be located at the same frequency bin as in the uniform case, we can proceed as in the uniform example and insert (2.15) into (2.34):

$$|2\pi \cdot \frac{2vf_0T_c}{c}| < \frac{\Delta t_{NUS}}{\Delta t_c} \cdot \pi \iff \tag{2.35}$$

$$v < \frac{\lambda}{4T_c} \cdot \frac{\Delta t_{NUS}}{\Delta t_c} \tag{2.36}$$

and hence

$$v_{max,NUS} = \frac{\lambda}{4T_c} \cdot \frac{\Delta t_{NUS}}{\Delta t_c} \tag{2.37}$$

Since Δt_c is the grid size in the uniform case there is no larger possible grid size of which each sample is an integer multiple. Hence $\Delta t_c \geq \Delta t_{NUS}$ and $v_{max} \geq v_{max,NUS}$.

Note that this is only applicable if the sample points can be placed at integer multiples of the grid size, that is on-grid sampling as opposed to off-grid sampling [25, p.9]. If the samples are randomly chosen there is no well-accepted definition of the Nyquist critical frequency [27, p.31].

2.8 Interference

Radar interference can have different definitions and sources, such as reflections from unwanted targets or surroundings, known as clutter, or it can be due to jamming as a result of electronic warfare countermeasures [14]. In the context of this thesis, interference is defined as the reception of foreign signals. The radar being affected by the interference is called *victim radar*, while the radar transmitting the signals causing the interference is called *aggressor radar* [28]. Interference can be divided into three categories, incoherent interference, partially coherent interference, and coherent interference. Each of these interference types is due to radar-to-radar interference but is classified differently depending on the similarity of the signals being transmitted and received. Coherent interference occurs when the received foreign signal is no different from the transmitted signal, $x_R(t) = x_T(t-\tau)$. The mixing of the foreign signal in the receiving radar will then result in a clean sinusoid and the interfering signal will appear as a proper target in the range-doppler image, thus rendering a *ghost target* which causes false detections [11, p.7].

2.8.1 Cross-correlation and Interference

In the previous sections, it is explained that a radar transmits CPIs of chirps, which are reflected, received and mixed with the transmitted signal. The radar samples incoming signals only in conjunction with transmitting. Assume now that we have a setting without reflecting objects, but with two identical radars facing each other at a distance R, and having a probability of causing mutual coherent interference. Denote the victim and aggressor radar with the superscript $^{(v)}$ and $^{(a)}$ respectively. Let both radars transmit CPIs with total CPI time T_f. Let the victim and aggressor radars transmit their CPIs at times $t^{(v)}$ and $t^{(a)}$ where $0 \leq t^{(v)} \leq T_f^{(v)}$ and $t_0^{(a)} \leq t^{(a)} \leq T_f^{(a)}$. The victim radar will then receive the aggressor radar's signals at $\tau = \frac{R}{c} + t_0^{(a)}$, such that $x_R^{(v)}(t) = x_T^{(a)}(t-\tau)$. Now assume that the distance between the radars is small enough and the CPI is long enough such that $T_f^{(v)} \geq \frac{R}{c}$. For some transmit start time of the aggressor radar $t_0^{(a)}$ the victim radar will receive $x_T^{(a)}$ while chirping, and will hence sample the received signal.

The number of interfered chirps in each radar, caused by coherent interference, can then be measured as the number of chirps received by the victim radar while the victim radar is also transmitting. Let again the signal transmitted by the victim radar at time t be denoted $x_T^{(v)}(t)$ and the signal transmitted by the aggressor radar and received by the victim radar be denoted $x_T^{(a)}(t-\tau) = x_R^{(v)}(t)$. Assume also that there are no reflective targets within any of the radars visible range, meaning that they will only receive signals transmitted from other radars.

Here, the signal value x itself is not important, only if the signal is present or not, and hence $|sgn(x)|$ will be used, where

$$|\operatorname{sgn}(x)| = \begin{cases} 0 \text{ if } x = 0 \\ 1 \text{ if } x \neq 0 \end{cases}. \tag{2.38}$$

The number of interfered signals can then be found by counting how many signals in $x_T^{(v)}(t)$ and $x_T^{(a)}(t)$ that occur at the same time t, using the $|sgn(x)|$ function.

The cross-correlation is defined as

Definition 1 *Continuous cross correlation of functions f and g*

$$(f \star g)(\tau) = \int_{\infty}^{\infty} \overline{f(t)} g(t+\tau) dt \tag{2.39}$$

at time displacement τ with complex conjugate operation $\overline{\bullet}$

Using the cross-correlation function, the amount of interfered signal for some delay τ can be found as $(|\operatorname{sgn}(x_T^{(v)})| \star |\operatorname{sgn}(x_R^{(v)})|)(\tau)$. In this work, the signals processed are sampled, and hence we will instead have use of the discrete cross-correlation function.

Definition 2 *Deterministic Discrete Cross-correlation of signals f[n] and g[n]*

$$CC_{fg}[l] = \sum_{n=-\infty}^{\infty} f[n]\overline{g[n+l]} \tag{2.40}$$

at correlation lag l with complex conjugate operation $\overline{\bullet}$

Denote the discretized signals $x_{T,n}^{(v)}$ and $x_{R,n}^{(v)}$ at index n. The expression $|\operatorname{sgn}(x_{T,n}^{(v)})|$ indicates when in time the victim radar transmits, and thereby also samples, signals. The expression $|\operatorname{sgn}(x_{T,n}^{(v)})|$ on the other hand represents when in time the victim radar receives the aggressor radar's transmitted signals. The interpretation of using the discrete cross-correlation on the two discrete signals at some lag l is that the number of samples at which we both sample and receive foreign signals are counted.

2.9 Figures of Merit

In order to be able to measure the quality of some frequency spectra and correlations some figures of merit are introduced in the following section. Here, three metrics are introduced which later on will aid when comparing the quality of transmit codes and their spectra.

2.9.1 Peak Sidelobe Ratio

Let x be some signal and $\mathcal{F}(x) = X$ its corresponding frequency spectrum, then the power spectrum in dB is given as $S_{xx}(f) = 20\log_{10}(|X(f)|)$. The Peak Sidelobe

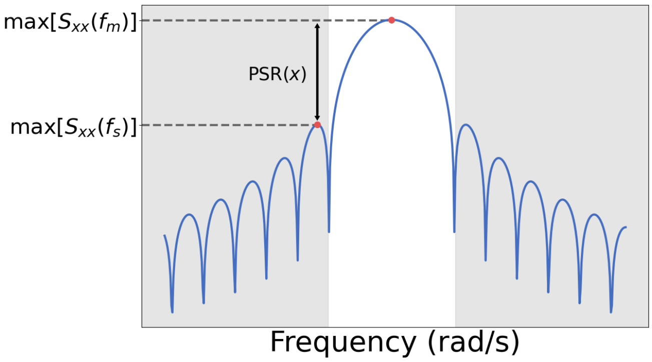


Figure 2.9: Figure showing an example of the PSR in a power spectrum $S_{xx}(f)$. The shadowed area marks the sidelobe region f_s, and the non-shadowed area marks the mainlobe region f_m. The two dots (red) represent the positions of the max mainlobe amplitude and the max sidelobe amplitude which in this specific example is located at the peak of the first side lobe. The PSR is given as the distance between these two. The unit on the y-axis is dB.

Ratio (PSR) is the ratio of the mainlobe peak and the largest sidelobe peak in a spectrum X, which in the logarithmic power spectrum can be written as

$$\mathrm{PSR}(x) = 20\log_{10}(\max[|X(f_s)|]) - 20\log_{10}(\max[|X(f_m)|]) \tag{2.41}$$

where f_s is the indices of the sidelobe region of the spectrum and f_m is the indices of the mainlobe region. Simply put this is the amplitude of the largest sidelobe divided by the amplitude of the mainlobe. These regions are depicted in Figure 2.9.

2.9.2 Full Width at Half Maximum

The Full Width at Half Maximum (FWHM) measures the spectral width of a mainlobe. Let again x be some signal and $X(f)$ its corresponding frequency spectrum at frequency f, and the power spectrum in dB is given as $S_{xx}(f) = 20\log_{10}(|X(f)|)$. Denote by f_a and f_b the frequency indices in the mainlobe region of the spectrum, $S_{xx}(f_m)$, such that $S_{xx}(f_a) = S_{xx}(f_b) = \max[S_{xx}(f_m)] - 3$ and $f_a < f_b$, then

$$\mathrm{FWHM}(x) = f_b - f_a \tag{2.42}$$

which is the spectral width of the mainlobe of the normalized power spectrum at -3 dB. An explanatory graphic of the FWHM measure is shown in Figure 2.10. The mainlobe width affects the ability to resolve two separated peaks, as a narrow mainlobe will make the peaks easier to resolve, as also stated in [29], [30] and [21, p.40].

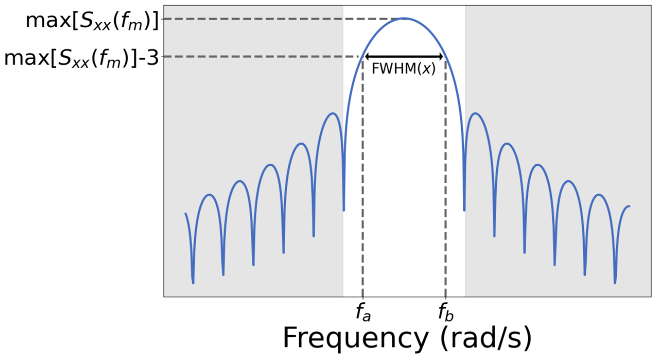


Figure 2.10: Figure showing an example of the Full Width at Half Maximum (FWHM) for a power spectrum S_{xx}, where the shadowed area marks the sidelobe region, and the non-shadowed area marks the mainlobe region f_m. The unit on the y-axis is dB.

2.9.3 Correlation Sum

It will prove to be useful to be able to measure the discrete cross-correlation of two sequences f and g of length N at a lag τ, this is $CC_{f,g}[\tau]$ according to Definition 2, relative to the correlation between two identical uniform sequences $u = \mathbb{1}_{1,N}$ with length N at the same lag τ, this is $CC_{u,u}[\tau]$. Having two binary sequences their correlation will be bounded from above by $CC_{u,u}[\tau]$. From an interference perspective, this will correspond to having interference at each possible chirp in the overlapping interval of the CPIs. Let us therefore define the Correlation Sum (CS) as

$$CS(f, g, \boldsymbol{T}) = \sum_{\tau \in \boldsymbol{T}} CC_{f,g}[\tau] \tag{2.43}$$

which is the sum of the crosscorrelations of two sequences f and g at lags τ stored in $\boldsymbol{T}$.

2.10 Simulated Annealing

Simulated Annealing (SA) is a probabilistic heuristic algorithm designed to approximate a global minimum of a function that may have several local minima. The algorithm is typically used in approximating solutions to discrete combinatorial optimization problems, but can also be used to approximate solutions to continuous problems. The algorithm has for example proven useful in approximating solutions to the NP-hard “traveling salesman problem” (TSP) and the “knapsack problem” [31]. The main advantages of the Simulated Annealing algorithm is both that it is easy to implement and its ability to avoid getting stuck in a local minima. The drawback is on the other hand the difficulty of finding reasonable algorithm parameters, which also can have a large impact on the performance of the method [32].

The algorithm works by initializing a random solution a and a temperature T_0. From a, a neighboring solution a' is found. A neighboring solution is a solution close to a and how this a' is chosen is dependent on the problem structure. The process of finding a neighboring solution is called perturbation [33]. The energy, which is just another name for the cost function, is calculated for both solutions and if the newly found solution a' has a lower energy than the current solution a, then a' is accepted as the new current solution. If a' turns out to be a worse solution than a, in essence a' has higher energy than a, then a' might still be accepted as the current solution with some probability which depends on the temperature of the method.

The temperature of the method is governed by the cooling function, which can be defined in varying ways depending on the desired properties of the method. For example, a cooling function yielding a slowly decaying temperature will have a high probability of converging, but will at the same time demand many iterations in doing so. On the other hand, a cooling function yielding a quickly decaying temperature might not get enough iterations to find an optimal solution [34]. Two common cooling schedules are the linear cooling schedule $T_{n+1} = T_0 - \beta t$ and the geometric cooling schedule $T_{n+1} = T_n \cdot \alpha$, where T_0 is the initial temperature, t is the iteration time and β and $0 < \alpha < 1$ some constants. In the geometric schedule, α is generally set to values close to 1, for example in the range $\alpha \approx [0.7,\ 0.99]$. The cooling function, regardless of what kind is used, will in some manner decrease the temperature of the method. This implies that the probability of accepting a worse solution decreases with each iteration, and the method is more prone to accepting worse solutions in the beginning than in the end of the optimization. This step also allows the method to avoid getting stuck in local minima. Due to the temperature's importance to the convergence, the initial temperature T_0 becomes crucial to the convergence of the method. A high initial temperature will allow for many iterations and a high probability of convergence, while a lower T_0 has a lower probability of convergence, but will be faster due to the fewer iterations [35, Ch.4].

Given a set $A = \{a_1, a_2, ..., a_3\}$ on which a cost function $Energy : A \to \mathbb{R}$ is defined, together with some perturbation function $Perturb : A \to A$ and a cooling schedule $cooling$, the algorithm can be described as

As stated before the advantage of the Simulated Annealing algorithm is its ability to avoid getting stuck in local minima, and also that it is easy to implement. The drawbacks of the algorithm on the other hand is the difficulty that can arise with choosing parameters such as the cooling rate α and the initial and end temperatures T_0 and T_{min} [35, Ch.4].

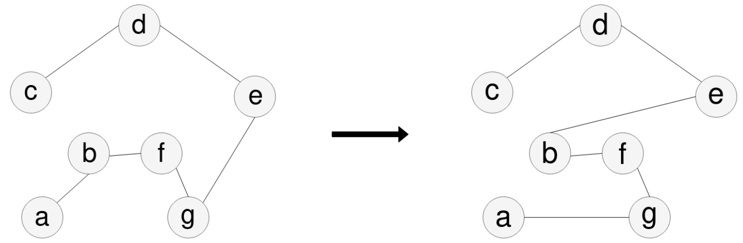
Algorithm 1: Simulated Annealing Algorithm

```
Input: $T_0$, $T_{min}$

$a \leftarrow a_0 \in_R A$          ▷ Initiate current solution $a$ as random element in A
$T \leftarrow T_0$
$E \leftarrow Energy(a)$                ▷ Calculate energy of current solution

while $T > T_{min}$ do
    $a' \leftarrow Perturb(a)$                ▷ Find neighbor solution $a'$ to a
    if $E \geq Energy(a')$ then
        $a \leftarrow a'$
        $E \leftarrow Energy(a)$
    else
        if $v \leq \exp(\frac{E - Energy(a')}{T}),\ v \in \mathcal{U}(0,1)$ then
            $a \leftarrow a'$
            $E \leftarrow Energy(a)$
        end
    end
    $T \leftarrow cooling(T)$
end
Return $a$
```

2.11 Traveling Salesman Problem

The traveling salesman problem (TSP) is an NP-hard graph problem, where a salesman travels between a number of cities. The salesman has to visit all of the cities but also wants to travel the shortest distance possible. This is a graph problem where the visiting order between some nodes is to be determined such that the distance is minimized.

The SA algorithm is a common method to use to solve this problem, where the energy function is the total traveled distance [36]. The visiting order is initialized as a random vector of the node names, where the index of the nodes in the vector corresponds to when in the order the node is visited. The perturbation in the TSP problem is typically implemented as interchanging two nodes in the visiting order, and hence changing positions of two elements in the vector. A visualization of the perturbation function in the SA algorithm applied to the TSP is shown in Figure 2.11.



$$\pi = [a, b, f, g, e, d, c] \qquad \pi' = [a, g, f, b, e, d, c]$$

Figure 2.11: A figure displaying the perturbation of a solution π in the TSP case when solving the problem using the Simulated Annealing algorithm.

2.12 Previous Research

To the best of the author's knowledge, there are no previous works with the same aim as this thesis. However, research in areas related to this work can be found. For example, in information and communication theory a lot of research has been done on finding sequences with low auto-correlation sidelobes.

2.12.1 Bi-Phase Codes

Barker codes and Minimum Peak Sidelobe (MPS) sequences are examples of sequences with low auto-correlation. These codes are bi-phase codes, meaning that each element in the sequences equals 1 or -1. These codes are designed to have minimal sidelobe levels in the auto-correlation function for a given code length N. The longest known bi-phase Barker sequence is of length $N = 13$, and due to the need for longer sequences other codes such as the MPS and MLS sequences were found. These are similar to the Barker sequences but have less strict constraints on the auto-correlation coefficients. The MPS sequences are designed to have auto-correlation coefficients which are low relative to the length of the sequence [14, pp. 817-824]. There are other examples of similar bi-phase sequences that are even longer than the before mentioned sequences, such as Maximum Length Sequences (MLS) and Gold sequences [37].

2.12.2 Sparse Antenna Arrays

In the related field of antenna array design, there is much work to be found on the subject of Sparse Antenna Arrays, in which Non-Uniform Linear Array (NULA) is most related. The aperture of the radar antenna array affects the DOA estimation since the distance between the first and last elements of the array will affect the width of the spectrum used in DOA estimation [38]. In order to avoid aliasing in the DOA spectrum, the minimal distance d between antennas in an antenna array is usually set to $d \leq \frac{\lambda}{2}$, but this small distance between antenna elements might cause mutual coupling [39]. This close distance between the antenna elements also implies that increasing the angular measurement resolution by increasing the aperture can be very expensive, due to the many elements needed. A NULA array design can be beneficial to use since it compared to a ULA allows for using fewer antenna elements which decreases the risk of mutual coupling between antennas, and NULAs are also able to detect more signal sources. An NULA array can be designed in a number of ways, such as a Minimum Hole Array, or Minimum Redundancy Array or Co-Prime Array [38].

One major advantage of the coprime array design is that there exists a closed form expression for the placements of the elements assuming that the number of elements is known. A coprime array is designed by choosing two relatively prime integers P and Q, such that their only common divisor is 1 and where $P < Q$. Using these integers two sets G containing indices are defined as

$$G_1 = \{p \cdot Q \mid p = 0, 1, 2, ..., 2P - 1\} \tag{2.44}$$

$$G_2 = \{q \cdot P \mid q = 0, 1, 2, ..., Q - 1\} \tag{2.45}$$

from which indices of the element positions in the coprime array are found as

$$G_{coprime} = G_1 \cup G_2 \tag{2.46}$$

which is used to construct the coprime array $\Omega_{coprime}$ [38]. The advantage of designing an antenna array in a coprime way is that the grating lobes of the two arrays appears on different positions of the spectrum, meaning that there is no constructive interference of sidelobes. Hence, the sidelobes are lower and the dynamic range is increased [40].

Chapter 3

Method

In this section, some background to the problem is given based on the theory presented in the previous chapter, along with some motivation as to why and when the problem occurs, and the cases regarded in this thesis are presented. From this background and theory, the problem is stated mathematically and then the method used to solve the problem is introduced. Along with the method used in the optimization problem, the different parameters and weights used are displayed. The section also introduces the different sequences used in the thesis, both sequences resulting from running the method and primitive sequences to be used for comparison with the results. In the end of the chapter the alternative "Coprime" method is introduced as well as an explanation of the Monte Carlo simulations used in the thesis.

3.1 Background and Motivation

As stated in the introduction, the aim of the thesis is to arrange chirps in sparse CPIs such that radars transmitting these CPIs at given relative times will coexist in the sense of not causing mutual interference. To this end, assume two identical FMCW radars are facing each other outside of each other's field of view, covering an area without reflective objects. Assume also that during some time interval both of these radars transmit their CPIs containing no idle time, and that the distance R is small enough such that the time the RF waves travel the distance R can be omitted. From the theory section and equations (2.2) and (2.3) it is known that the direct power density decreases with range as $\frac{1}{R^2}$ while the indirect power density, where the signal has been reflected, decreases with range as $\frac{1}{R^4}$. This implies that the two radars facing each other at a range well outside each other's field of view are probable to interfere with each other. In order to not degrade the CPI rate, and hence degrade the overall radar performance, Time Division Multiplexing is not considered. From the theory section "2.3 FMCW Radar Signal Model" it is also known that FMCW radars sample incoming signals while chirping, and hence the CPIs have to be designed in a sparse way, such that only one of the radars transmits a chirp at a time.

Given two radars, the amount of interference between the two CPIs will depend on how much each chirp in the first CPI overlap the chirps in the second CPI.

Without loss of generality, the worst-case scenario of having the CPIs time-shifted relative to each other only with discrete time steps with size equal to the chirp time T_c is assumed. This way the interference-causing chirps overlap each other fully, and cause maximal interference. This assumption allows for regarding the CPIs in slow time only, and a CPI can then be represented by a sequence of ones and zeros, where 1 represents a transmitted chirp, and 0 represents not transmitting a chirp. The binary sequences which govern when the radars transmit their signals are again denoted Ω, in which $\Omega_i \in \{0, 1\} \forall\, i \in \{1, 2, ..., M\}$ for M chirps. In the following, the superscript will denote different sequences, while the subscript denotes the index in the sequence. For example a 1 on index n represents a CPI with a chirp at the n:th position.

Many different problem configurations can be regarded, concerning a different number of sequences, numbers of chirps, and overlap between sequences. In this thesis, the following two cases were considered:

Homogeneous Case
The case when two radars transmit identical but time-shifted CPIs. This implies that the sequence has to be able to overlap itself while not causing interference.

Heterogeneous Case
The case when three radars are allowed to transmit different and time-shifted CPIs. However, it is assumed that the number of chirps in the sequences are equal.

Both of these cases are depicted in Figure 3.1. In each case, the CPIs are assumed to have the same CPI time $T_f = 512T_c$, with T_c being the chirp time. This implies that each CPI can contain a maximum of 512 chirps. It is also assumed that the CPIs overlap each other in time with 103 chirps, which corresponds to about 20% of the CPI length. This overlap is depicted in Figure 3.1. In both cases, the end goal was to find orthogonal sequences, in essence sequences that do not cause mutual interference, and hence have 0 correlation at the lag corresponding to an overlap of 103 chirps. To measure the interference between two sequences the Correlation Sum (CS), introduced in the theory section in (2.42), was used. For example, assume the two radars start to transmit according to the binary sequences $\Omega^{(1)}$ and $\Omega^{(2)}$ at times t_1 and t_2 respectively, where both sequences are of length 512 and $t_1 < t_2$. The interference between the sequences when t_1 and t_2 are set such that the CPIs overlap 103 lags is found as the correlation between the sequences at lag 408 if the correlation is carried out in the order $\text{Corr}(\Omega^{(1)}, \Omega^{(2)})$. Hence, as a measure of the interference between the sequences $\Omega^{(1)}$ and $\Omega^{(2)}$ with this overlap the Correlation Sum $\text{CS}(\Omega^{(1)}, \Omega^{(2)}, 408)$ is used.

Due to the potential difficulties of making two radar systems transmit their CPIs at exact times and preserving an exact overlap of 103 chirps, allowing the CPIs to drift relative to each other might be an important feature. To allow for this, the CPIs have to be designed such that a small drift does not cause interference. A CPI designed this way will be referred to as a CPI, or sequence, with a *correlation buffer*, which means that the sequences are allowed a relative time drift of T_c be-

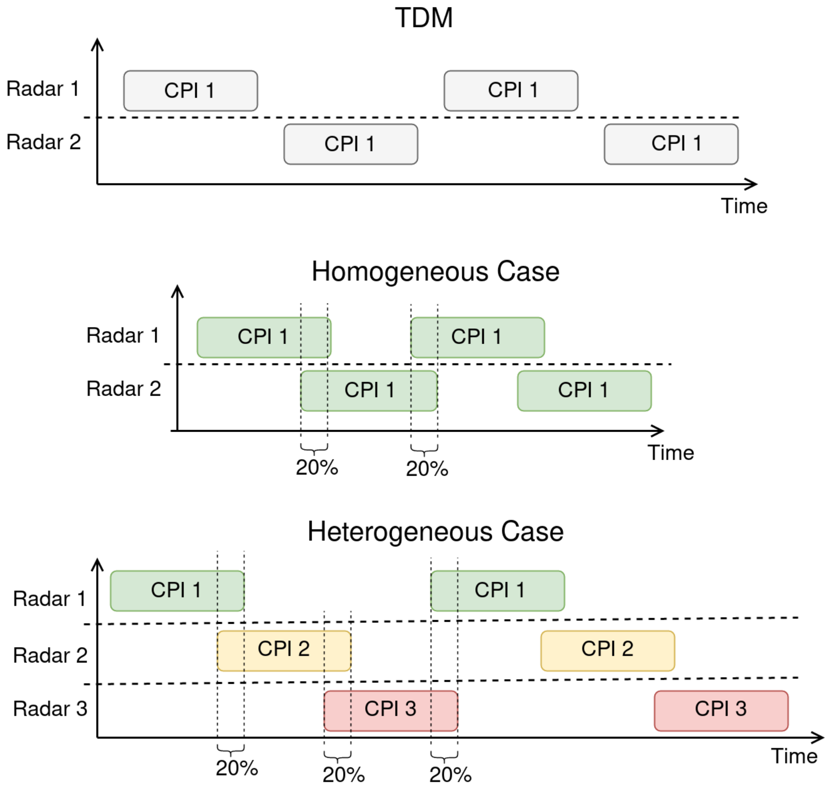


Figure 3.1: A figure depicting the different ideas of achieving co-existing radars. At the top the time division idea is depicted, followed by the homogeneous case, and the heterogeneous case.

tween the CPIs in either direction, without causing interference. In practice, this corresponds to having a cross-correlation equal to zero at lag 408, but also at lag 407 and 409.

From the theory section, it is known that shortening the sample time will result in a lower velocity resolution v_{res}, and hence it is desirable to keep the first and last chirps in each CPI. To preserve the v_{max} properties of a uniform CPI it is also needed to have at least two consecutive chirps in the CPI. Since the CPIs are to be designed in a sparse way to reduce interference, some chirps have to be removed in order to avoid overlapping chirps between the CPIs. From the theory section on Integration Gain, it is known that the Integration Gain is proportional to the number of chirps used in a CPI. As such, removing chirps will decrease the integration gain and thus risk increasing the Peak Sidelobe ratio. Getting higher sidelobes relative to the mainlobe causes a risk of getting worsened detection abilities, as explained in the section “2.5.4 Constant False Alarm Rate”. At the same time, as introduced in section “2.9.2 Full Width at Half Maximum” the width of the mainlobe will affect the resolvability of two peaks, as narrower peaks having a smaller FWHM value are easier to resolve. It is therefore desirable for the sequences to have as small PSR and FWHM as possible due to the implications these metrics have on the frequency spectrum and thus on the detection abilities.

3.2 Problem Statement

The problem can be regarded as consisting of 2 parts, where part one, P_1, concerns finding the optimal ordering of k number of ones and the rest zeros in a 512-long sequence. The other part, P_2, concerns finding the optimal number of ones to use in the sequence. Defining a vector of weights $\overline{\mathbf{w}}$ and a vector of figures of merit of some sequences $\Omega^{(1)}$ and $\Omega^{(2)}$ at some lag or lags τ as $\overline{\mathbf{f}}(\Omega^{(1)}, \Omega^{(2)}, \tau)$ defined by

$$\overline{\mathbf{f}}(\Omega^{(1)}, \Omega^{(2)}, \tau) = \begin{pmatrix} \mathrm{CS}(\Omega^{(1)}, \Omega^{(2)}, \tau) & \mathrm{PSR}(\Omega^{(1)}) & \mathrm{FWHM}(\Omega^{(1)}) \end{pmatrix} \tag{3.1}$$

$$\text{and} \quad \overline{\mathbf{w}} = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \tag{3.2}$$

the optimization problem can be described as

$$P_2 = \begin{cases} \underset{k}{\text{minimize}} \quad P_1(k) \\ \text{subject to} \\ \qquad\qquad k \in \{1, 2, ..., 512\} \end{cases} \tag{3.3}$$

$$P_1(k) = \begin{cases} \underset{\Omega}{\text{minimize}} \quad \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^{\mathsf{T}}(\Omega^{(1)}, \Omega^{(2)}, 408) \\ \text{subject to} \\ \qquad\qquad ||\Omega||_1 = k \\ \qquad\qquad \Omega_m \in \{1, 0\}, m \in \{1, 2, ..., M\} \\ \qquad\qquad \Omega_1 = 1,\ \Omega_M = 1 \\ \qquad\qquad \exists i \in \{1, 2, ..., M\} : \Omega_i = \Omega_{i+1} = 1 \end{cases} \tag{3.4}$$

where

- Ω is defined as in the theory section, which is an integer sequence of ones and zeros. In P_1 only the order of the elements in Ω is changed, while the number of non-zero entries in Ω is changed in P_2.
- $||\Omega||_1$ is the L_1 norm, measuring the number of ones in Ω. Thus there is a constraint that the number of ones used in the sequence should be k, which is given as input to the problem.
- $\Omega_1 = 1$ and $\Omega_M = 1$ puts constraints on the endpoints of the sequence, in order to preserve the length of the sequence and thereby the velocity resolution, as seen in (2.23).
- $\exists i \in \{1, 2, ..., M\} : \Omega_i = \Omega_{i+1} = 1$ puts the constraint that the sequence has to have at least one pair of consecutive chirps, which makes $\Delta t_{NUS} = \Delta t_c$ in (2.37) which gives $v_{max,NUS} = v_{max}$.

- k is limited to the interval [1,512], since 512 is the length of Ω.

In the homogeneous case, using the same sequence repeatedly is considered, and hence $\Omega^{(1)} = \Omega^{(2)}$ in $P_1(k)$. In the heterogeneous solution on the other hand this is not a strict constraint, and $\Omega^{(1)}$ and $\Omega^{(2)}$ are allowed to be different. Note that the PSR is negative by definition, and given the introduced definition of the cost function

$$\overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^{\mathsf{T}}(\Omega^{(1)}, \Omega^{(2)}, 408) \tag{3.5}$$

a sequence will benefit from having a PSR value as negative as possible, and values of the CS and FHWM close to 0. In other words, sequences having as few interfered chirps as possible while having a spectrum with a high dynamic range and narrow mainlobe will be premiered by this cost function.

By solving problem P_2 it was expected that a sequence with the optimal number of zeros organized in an optimal way would be found. The expected sequence would be optimal in the sense of having desirable spectral qualities while not causing interference at a 103-chirp overlap would be returned. In order to be able to compare this optimal sequence some primitive sequences were produced, which are designed only to have 0 correlation at one or more lags. Note that the problem of using a correlation buffer is formulated in the same way but with the lag number 408 replaced by the list of lags $[407, 408, 409]$. The sequences used are described in short below, and introduced more thoroughly in the beginning of chapter “4 Results”.

Uniform sequence
A uniform sequence of length 512 exclusively consisting of ones, hence representing a uniform chirp burst.

Primitive sequence without buffer
A sequence defined to have 0 correlation at exactly lag 408 and -408 using as few zeros as possible.

Primitive sequence with buffer
A sequence defined to have 0 correlation at exactly lags 407 to 409 and -409 to -407 using as few zeros as possible.

Primitive sequence 2
A sequence defined to have 0 correlation at exactly lag 408 and -408, using 120 zeros.

Homogeneous case sequence without buffer
The optimal sequence found by the Simulated Annealing algorithm solving the homogeneous problem, using the same sequence repeatedly, by having 0 correlation at lags ±408.

Homogeneous case with buffer
The optimal sequence found by the SA algorithm which solves the homogeneous problem, using the same sequence repeatedly, by having 0 correlation at lags 407 to 409 and -409 to -407.

Heterogeneous case sequence without buffer
The optimal sequence found by the SA algorithm which solves the heterogeneous problem, using different sequences, by having 0 correlation at lag ± 408.

Heterogeneous case sequence with buffer
The optimal sequence found by the SA algorithm which solves the heterogeneous problem, using different sequences, by having 0 correlation at lags 407 to 409 and -409 to -407.

3.3 Simulated Annealing

To solve the optimization problems posed in the previous section, the Simulated Annealing algorithm, presented in Algorithm 1, was used. Since the problem is divided into two parts also two algorithms were used, where one algorithm solves the problem of ordering the sequence in an optimal way in P_1, and a second algorithm finds the optimal number of ones to use in the sequence in P_2. The algorithms were implemented a bit differently in the heterogeneous and homogeneous case and hence both algorithms are explained.

3.3.1 Homogeneous solution algorithms

Again the notation $\mathbb{1}_{N\times M}$ is used for the matrix with N rows and M columns with all entries being 1. The algorithm used to solve P_1 in the homogeneous case formulated above is defined in Algorithm 2. The algorithm begins by initiating a solution with k zeros placed randomly in a 512-long vector. The energy E, in essence the cost function, of this solution is evaluated and compared with the energy E' of a neighboring solution found through the perturbation function. If the neighboring solution has a lower energy it is accepted as the current solution. However, if the neighboring solution has a higher energy than the current solution it is accepted anyways with a temperature-dependent probability $\exp(\frac{E-E'}{T})$. The temperature is decreased according to the cooling schedule, and the process is repeated until the temperature reaches the minimal temperature T_{min}.

In this algorithm, the cooling schedule is defined as $T_{n+1} = \alpha \cdot T$ where $\alpha = 0.999$ to make the method converge with high probability, at the cost of being quite slow. The energy function is defined as $\overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega, \Omega, 408)$ with $\overline{\mathbf{w}}$ and $\overline{\mathbf{f}}$ as defined in (3.1), and effectively this is a weighted sum of the figures of merit of the sequence Ω and the weights specified in $\overline{\mathbf{w}}$. The weights used in the energy function are given by Table 3.1. To denote that element a is chosen at random from the set b in the algorithm the notation $a \in_R b$ is used. The perturbation function that finds a

Algorithm 2: Homogeneous Solution SA Algorithm P_1

```
Input: $k$, $T_0^{P_1}$, $T_{min}^{P_1}$

$\Omega \leftarrow \mathbb{1}_{1\times 512}$                    ▷ Initialize as vector with ones
$\Omega_{a_i} = 0, a_i \in_R \{1, 2, ..., 512\} \ \forall \ i = 1, 2, ..., 512 - k$

$T \leftarrow T_0^{P_1}$
$E \leftarrow \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega, \Omega, 408)$                    ▷ Calculate current energy

while $T > T_{min}^{P_1}$ do
    $\Omega' \leftarrow Perturb(\Omega)$                    ▷ Find new solution $\Omega'$
    if $E \geq \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega', \Omega', 408)$ then
        $\Omega \leftarrow \Omega'$
        $E \leftarrow \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega, \Omega, 408)$
    else
        if $v \leq \exp(\frac{E - \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega', \Omega', 408)}{T}), \ v \in \mathcal{U}(0, 1)$ then
            $\Omega \leftarrow \Omega'$
            $E \leftarrow \overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(\Omega, \Omega, 408)$
        end
    end
    $T \leftarrow 0.999 \cdot T$
end
Return $\Omega$
```

neighbor solution in the SA Algorithm used in the homogeneous case Algorithm 2 is described in Algorithm 3.

In problem P_2 the search space is limited to the integer interval $k \in \{3, 4, ..., 512\}$, where k denotes the number of ones to use in the P_1 optimization. To find the optimal k an exhaustive search algorithm was used, which iteratively evaluated $P_1(k) \ \forall \ k \in \{3, 4, ..., 512\}$ and returned the k which minimized $P_1(k)$. Essentially the solution was found by evaluating Algorithm 2 for some number of ones k, and doing this both finds what number of ones to use in the sequence and how the sequence is to be organized to be considered optimal with regards to the energy function. The interval to which k is bounded starts at 3 since this is the minimal number of ones needed to fulfill the constraints posed in 3.4. However, values of k close to 0 and 512 did not give any interesting solutions since a small value k gives fewer zeros than needed to get 0 cross-correlation. At the same time, too high k values gives too bad spectral qualities. Hence, in practice, k values were chosen in the interval $[250, 450]$ to make the solution converge faster.

Algorithm 3: Perturbation function used in Algorithm 2

```
Perturb($\Omega$):
    $\Omega' \leftarrow \Omega$                                  ▷ Create copy $\Omega'$ of sequence $\Omega$
    $G \leftarrow$ indices where $\Omega = 1$ excluding first and last index
    $H \leftarrow$ indices where $\Omega = 0$
    $g \xleftarrow{R} G$                                  ▷ Draw $g$ uniformly from $G$
    $h \xleftarrow{R} H$                                  ▷ Draw $h$ uniformly from $H$
    $\Omega'_g \leftarrow 0$                              ▷ Replace element at index $g$ with 0
    $\Omega'_h \leftarrow 1$                              ▷ Replace element at index $h$ with 1
return $\Omega'$
```

Table 3.1: Table displaying the weights used in $\overline{\mathbf{w}}$ when optimizing P_1 in the Homogeneous solution.

Homogeneous Solution Weights			
Sequence	CS	PSR	FWHM
Without buffer	2.3	4.5	0.25
With buffer	4.8	4.6	0.25

3.3.2 Heterogeneous solution algorithms

In the heterogeneous solution, three sequences allowed to be different were used. Again the algorithm used was a Simulated Annealing algorithm, but since the sequences now were allowed to be different some changes had to be made compared to the homogeneous case. The cooling schedule was defined just as in the homogeneous case algorithm, as $T_{n+1} = \alpha \cdot T$ where $\alpha = 0.999$. The energy function on the other hand was implemented differently since there in this case is more than one sequence. The energy is instead calculated as the sum of the energies of each sequence, as described in Algorithm 5. Thus in the heterogeneous case algorithm, described in Algorithm 4, compared to the homogeneous case dito, each of the three sequences was updated one at a time, but the energy was evaluated using the energy of all the solutions combined. As seen in the description of the energy function, in Algorithm 5, it works by iterating over all sequences and summing up their costs. Another difference to the homogeneous case method is how to handle the correlation. Now the CPIs overlap each other in consecutive pairs, for example CPI 1 overlaps CPI 2, and CPI 2 overlaps CPI 3, according to how the heterogeneous case has been defined. Hence, the correlation measure is calculated between each consecutive pair of sequences in the order described in the heterogeneous case. Due to this, in the energy function, the i and j indices are always consecutive pairs, with a periodicity of 3, for example $(1,2) \rightarrow (2,3) \rightarrow (3,1)$.

How the correlation is used in the energy function is also shown in Algorithm 5.

Algorithm 4: Heterogeneous Solution SA Algorithm P_1

```
Input: $k$, $T_0$, $T_{min}$,
 $S \leftarrow$ empty list                         ▷ List to fill with sequences
for $i$ in $nbr\_sequences$ do
    $\Omega \leftarrow \mathbb{1}_{1\times 512}$                          ▷ Initialize one sequence
    $\Omega_{a_i} = 0,\ a_i \in_R \{1, 2, ..., 512\}\ \forall\ i = 1, 2,\ \dots, 512 - k$
    $S$.insert($\Omega$)
end

$T \leftarrow T_0$
$E \leftarrow Energy(S)$

while $T > T_{min}$ do
    for $\Omega$ in $S$ do
        $temp \leftarrow copy(S)$
        $\Omega' \leftarrow Perturb(\Omega)$
        replace $\Omega$ with $\Omega'$ in $temp$
        if $E \geq Energy(temp)$ then
            $S \leftarrow temp$
            $E \leftarrow Energy(temp)$
        else
            if $v \leq \exp(\frac{E - Energy(temp)}{T}),\ v \in \mathcal{U}(0, 1)$ then
                $S \leftarrow temp$
                $E \leftarrow Energy(temp)$
            end
        end
    end
    $T \leftarrow 0.999 \cdot T$
end
Return $S$
```

It is assumed that each sequence in the heterogeneous case uses the same amount of ones in the sequence, and hence to solve the P_2 problem an exhaustive search algorithm is used again. Thus, the only difference between solving the P_2 problem in the heterogeneous and homogeneous case was found in the algorithm solving the P_1 problem. The weights used in the heterogeneous case are found in Table 3.2.

The perturbation in the algorithms solving P_1 was inspired by the TSP problem perturbation, where the structure of the problem makes the definition of "neighbor solution" difficult to interpret. Hence, in the same way as in the TSP problem,

Algorithm 5: Heterogeneous case P_1 Energy function

```
Energy:
Input: S                                              ▷ Input list with sequences
energy=0
for i in {1,3} do
    j=mod(i, 3)+1                                     ▷ Find consecutive indices
    energy $+=$ $\overline{\mathbf{w}} \cdot \overline{\mathbf{f}}^T(S_i, S_j, 408)$    ▷ Add energy of seq. $S_i$ and $S_j$
end
return energy
```

Table 3.2: Table displaying the weights used in $\overline{\mathbf{w}}$ when optimizing P_1 in the Heterogeneous solution.

Heterogeneous Solution Weights			
Sequence	CS	PSR	FWHM
Without buffer	3.2	4.2	1/4
With buffer	6.2	4.2	1/4

a neighboring solution is chosen by switching the position of two elements with different values. Since the algorithm initializes the sequences as random sequences with k ones, and these only are re-ordered, the algorithm preserves the constraint on using a constant amount of ones when solving P_1. Both when setting some random elements to 0 when initializing the sequences, and when choosing between elements to switch in the sequences in the perturbation, the first and last elements are excluded from the set of possible elements to choose. Hence these entries remain equal to 1 throughout the optimization process, and thus the algorithms obey the third constraint presented in (3.4).

As explained in the theory section, choosing optimization parameters to use in the Simulated Annealing algorithm is notoriously hard. The cooling function along with the temperature parameters, for example T_{min} and T_0, was chosen to guarantee a slow convergence and many iterations when solving the P_1 problem. This was due to the size of the problem, where a sequence of 512 elements able to take 2 values each can be organized in 2^{512} ways. To get this slow convergence the T_{min} and T_0 were set such that there was a large difference between them, and the cooling parameter α was set close to 1 to make the temperature decrease between each iteration small. The weight of the cost function Correlation Sum was set sufficiently high to generate results with no correlation. This was done since no solution with any overlapping chirps in the CPIs was wanted, as any coherent interference invalidates the information in the CPI, rendering the frequency spectral

Table 3.3: Table displaying the optimization parameters used when optimizing P_1 in both the Homogeneous and Heterogeneous solution, with and without correlation buffer.

Heterogeneous Solution Optimization Parameters	
Parameter	Value
$T_{min}^{P_1}$	10^{-10}
$T_0^{P_1}$	10^3

qualities irrelevant. The exact weights used were found empirically, by iteratively running the algorithms and evaluating the convergence and the results. When a solution was found without any correlation at lag 408, and having a low PSR and FWHM relative to other simulations, it was deemed as a good configuration of the weights.

3.4 Coprime method

To be able to compare the simulated annealing method with some already existing similar method, the coprime method of designing sparse antenna arrays was used, which was introduced and explained in the theory section and in [38]. The issue of designing sparse antenna arrays, as done with the coprime method, is similar to the problem posed in this thesis. In the sparse antenna array case, a number of antennas are to be placed at a uniformly space grid, and the aperture of the array affects the resolution of the DOA estimate. In this thesis, the problem instead consists of placing chirps on a uniform time grid, where the time difference between the first and last chirp affects the resolution of the velocity estimate.

The comparison was conducted by first generating a sparse time CPI using the coprime method and comparing this coprime sequence with the optimal sequence found using Simulated Annealing. To make such a comparison reasonable, coprime sequences with about the same length as the optimal sequences were found. The length of a coprime (P, Q) sequence is given as $P \cdot Q$, and the condition that P and Q have to be coprime gives the condition $\text{GCD}(P, Q) = 1$. Suitable coprime numbers were found by iterating P and Q over all numbers in the interval [1,512] until the conditions $P \cdot Q \approx 512$ and $\text{GCD}(P, Q) = 1$ were met. The closest sequence length for which reasonable results were found was length 513, where 4 different possible sequences were found.

It was decided that the fairest comparison would be to compare the optimal sequence with the coprime sequence constructed with $(P, Q) = (2, 171)$, since this coprime sequence did not have the same aliasing issues as many other coprime sequences with about the same length. After generating the sequence using

$(P, Q) = (2, 171)$ and the method proposed in [38] the spectral qualities and the correlation of the coprime sequence were found.

3.5 Monte Carlo Simulations

Due to the method used having stochastic parts, where there is a possibility of accepting a worse solution, Monte Carlo methods were used to find some statistical measures of the performance of the method, such as the variance of the number of zeros used in the sequences. Monte Carlo methods were also used to evaluate the effect on the v_{res} property of a sparse CPI sequence.

To approximate the expected value and variance of the number of zeros used in a sequence and the energy state of the final solution, the method was run 500 times, from which data on the number of zeros used and the energy of the sequence were gathered. The descriptive statistics were thereafter calculated using standard methods of calculating sample mean and variance.

To find the velocity difference limit below which two targets' corresponding spectral peaks can not be resolved as 2 peaks with unit probability, Monte Carlo methods were used. This is a way of measuring the v_{res} resulting from transmitting according to one of the found sequences. The Monte Carlo method is used since the initial phase of the transmitted signals has to be varied randomly to achieve realistic results. The sought-after value is the expected probability that two peaks in the slow time spectrum can be separated at a given velocity difference. To this end, two targets were simulated at the same range but with a velocity difference. This was done 4000 times for each velocity difference Δv with an added random phase to each iteration, and the expected probability of being able to separate two peaks was calculated using the sample mean. This was done for each velocity difference in a certain range, to find at which velocity difference the system almost surely would be able to separate two peaks.

Some noise was added to the signals of the simulated targets, such that the mean SNR was 18.015 dB, with a standard deviation of 2.67 dB. A point in the spectrum was classified as a peak if its amplitude was 10 dB above the mean noise level and if its two neighboring points were smaller than the point itself. There was also a constraint put on the values of the curve connecting two peak candidates. Two points located within a predefined frequency range, 10 dB above the mean noise level and with neighbors with smaller amplitudes than the points themselves were deemed separate peaks only if the curve connecting the points at some point was -3 dB smaller than the largest peak. If this condition is not met, the highest of the two is classified as a peak and the smallest of the two is not. This classification was inspired by the Rayleigh-Resolution [15, p.785], and a figure displaying when two peaks can be separated or not is displayed in Figure 3.2. The phase was randomized in each iteration to avoid having the sources coherent, which otherwise would cause a deterministic interference between the spectra. Due to this added randomness, the Monte Carlo method was considered suitable.

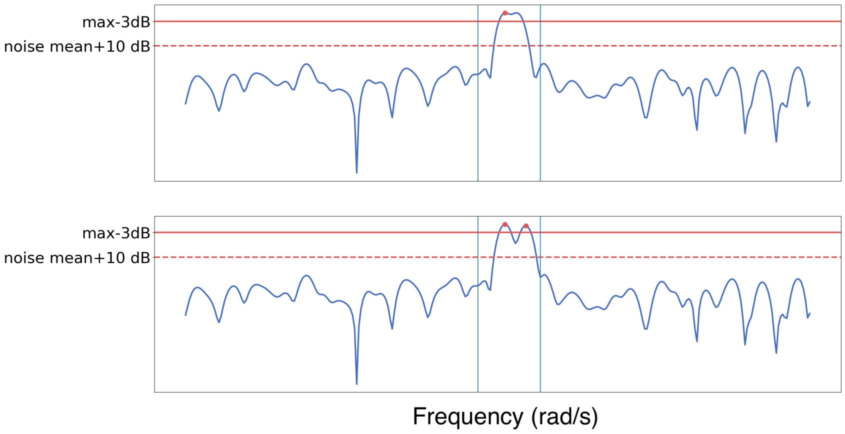


Figure 3.2: Figure showing how a peak in a logarithmic (dB) power spectrum would be classified. In the upper plot, the curve connecting the two possible peaks is never lower than the solid red line, and hence the points are classified as one peak. The blue vertical lines show the frequency range in which two points can be classified as the same peak, and the red dots represents the points classified as peaks.

3.5.1 Implementation

The algorithm and the Monte Carlo simulations were implemented in Python, using the Numpy and SciPy packages. When running the algorithm we continuously get a sequence of ones and zeros, which spectral qualities are to be determined. To this end, the power spectrum of the sequence is calculated by first multiplying the sequence with a 512-long Hann window found using the NumPy function `numpy.Hann(512)`. After the window was applied to the sequence its frequency content was found by first zero padding the end of the signal with 1536 zeros and then applying a 2048-point DFT to it by using the `numpy.fft.fft(signal,2048)` function. The sequence was zero-padded with 512 zeros added to the end of the sequence to decrease straddle loss, while keeping the signal length as a power of 2, allowing for using the FFT implementation of the DFT. Having the frequency content of the signal the power spectrum along with the FWHM and PSR could be calculated.

Chapter 4

Results

This section is divided into four parts, where the first part presents the sequences found when running the Simulated Annealing algorithm, along with the sequences' power spectra, correlation plots, and figures of merit, in both concerned cases with and without buffering lags in the correlation functions. In the second part, the results of using the coprime method of generating sequences are presented. In the third part, Monte Carlo simulations of the velocity resolutions are shown, and lastly some results of the convergence of the method are also presented.

4.1 Simulated Annealing Results

In this subsection the optimal sequences found by the Simulated Annealing algorithm are presented along with their resulting power spectra and correlation plots, as well as tables presenting the sequences' figures of merit. The subsection begins with displaying the values of the figures of merit of each sequence, both the uniform sequence, the primitive sequences, and the sequences found running the Simulated Annealing Algorithm and using the coprime method. After these tables, barcode plots of the actual sequences are displayed along with their normalized crosscorrelation and power spectrum plots. The spectra in this section are presented as normalized power spectra, where each has been normalized by the maximum amplitude of the spectrum of a uniform sequence. The mainlobe and sidelobe dB values are also given for the normalized spectra.

The first plot presented in Figure 4.1 displays the primitive sequences where they are visualized as barcode plots. Despite their similarities, the first and second plot, corresponding to sequences without and with correlation buffer respectively, are actually slightly different as the white areas in the second plot are 2 lags wider than in the first plot. This is not visible here, but looking at the correlation plots further on will make the difference clear. In the third barcode plot in the figure, a number of zeros are inserted in the middle of the sequence in order to make the number of zeros in the "Primitive 2" sequence equal the number of zeros in the optimal sequence without a correlation buffer in the homogeneous case.

Table 4.1: Table displaying the figures of merit of the uniform and coprime sequence and the primitive and optimal homogeneous case sequences. Due to table size limitations "Homogeneous" is abbreviated to "Hom", and "With Buffer" is abbreviated to "w.b".

Category	Sequence Name	PSR (dB)	FWHM (rad/s)	MSA[1] (dB)	MA[2] (dB)	Nbr Zeros
Uniform	Uniform	−32.2	0.0366	−32.2	0	0
Primitive seqs.	Primitive	−20.9	0.0292	−21.7	−0.8	104
	Primitive w.b[3]	−20.7	0.0278	−21.5	−0.8	107
	Primitive 2	−15.4	0.0282	−16.8	−1.4	120
Optimal seqs.	Hom.	−32.6	0.0225	−33.4	−0.7	120
	Hom. w.b [3]	−31.8	0.0235	−32.5	−0.8	127
Coprime(P,Q)	Coprime(2,171)	−18.9	0.0272	−26.6	−7.8	340

[1] Max Sidelobe Amplitude
[2] Mainlobe Amplitude
[3] With Buffer

Table 4.2: Table displaying the figures of merit of the Heterogeneous case sequences. The upper half of the table displays the sequences designed with a correlation buffer, and the lower part displays the same figures of merit of the sequences without this buffer.

	Sequence Name	PSR (dB)	FWHM (rad/s)	MSA[1] (dB)	MA[2] (dB)	Nbr Zeros
With Buffer	Seq. 1	−27.4	0.0232	−28.2	−0.8	135
	Seq. 2	−31.7	0.0271	−32.5	−0.9	135
	Seq. 3	−32.4	0.0243	−33.2	−0.8	135
Without Buffer	Seq. 1	−33.1	0.0231	−33.9	−0.7	122
	Seq. 2	−32.2	0.0246	−33.0	−0.8	122
	Seq. 3	−32.9	0.0220	−33.7	−0.8	122

[1] Max Sidelobe Amplitude
[2] Mainlobe Amplitude

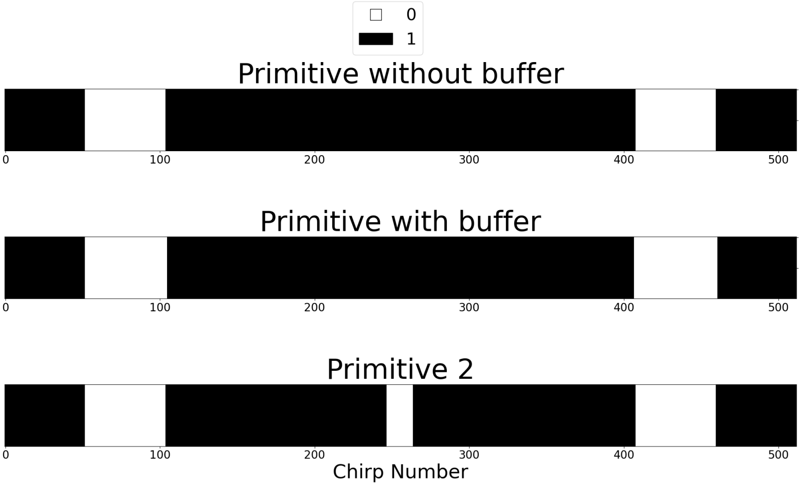


Figure 4.1: Figure showing barcode plots of the three primitive sequences used, where black lines correspond to 1 and white lines correspond to 0.

4.1.1 Homogeneous Case Results

The homogeneous case sequences were designed both with and without a correlation buffer and hence this section, presenting the results of the homogeneous sequences, is divided into two parts where the sequences with and without buffer are handled separately. The figures of merit of these sequences can be found in Table 4.1. In the following the power spectra and correlation plots of the sequences are presented along with barcode plots displaying the actual sequences.

Without Correlation Buffer

Here the results of the homogeneous case sequences without a correlation buffer are presented. The sequence found optimal uses 120 zeros which are organized as shown in Figure 4.2 as a barcode plot where each black line represents a 1 in the sequence, and a white line represents a 0. It is displayed in Figure 4.5 that the auto-correlation is 0 at lag 408, which means that the Simulated Annealing method successfully found a sequence that would not cause interference when overlapping itself with 103 lags. It can also be seen in Table 4.1 that the absolute PSR value of the found sequence is slightly larger than the absolute PSR of the uniform sequence, and hence this optimal sequence has a slightly better spectral dynamic range than both the uniform and the primitive sequence. This can also be observed in Figure 4.3 where the maximum sidelobe amplitude of the optimal sequence is significantly lower than the corresponding maximum sidelobe amplitude of the primitive sequence. The FWHM of this optimal sequence is slightly smaller than

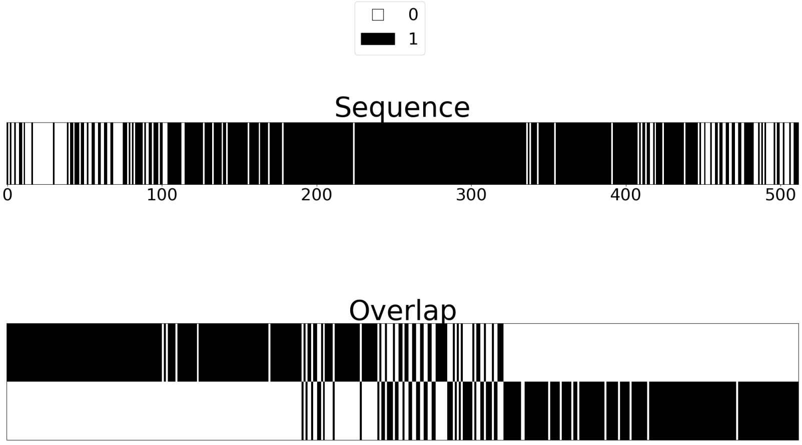


Figure 4.2: Figure showing a barcode plot of the optimal sequence for the homogeneous case without a correlation buffer, where black lines correspond to 1 and white lines correspond to 0. The upper plot shows the actual sequence and the lower plot shows the 103-point overlap where it can be seen that there is never more than one 1 placed at the same time.

that of the primitive sequence without a correlation buffer, but larger than that of the uniform sequence. Note that for this sequence two spectral plots are presented, one comparing the optimal and uniform sequence with the ordinary primitive sequence, and one instead comparing these with the "Primitive 2" sequence, seen in Figure 4.4.

With Correlation Buffer

Here the homogeneous case sequences with a correlation buffer are shown. This sequence is designed to have 0 correlation at 3 lags and the algorithm found it optimal to use 127 zeros to achieve this, as seen in Table 4.1. As seen both in this table and Figure 4.8, the algorithm succeeds in finding a sequence with 0 correlation at lags 407-409 and hence there are no interfered chirps at overlaps of 102, 103, or 104 chirps. The absolute PSR value of this sequence is a bit lower than in the case of having no correlation buffer, but is still better than the absolute PSR of the primitive sequence. The FWHM of this sequence is close to that of the optimal homogeneous sequence without a correlation buffer.

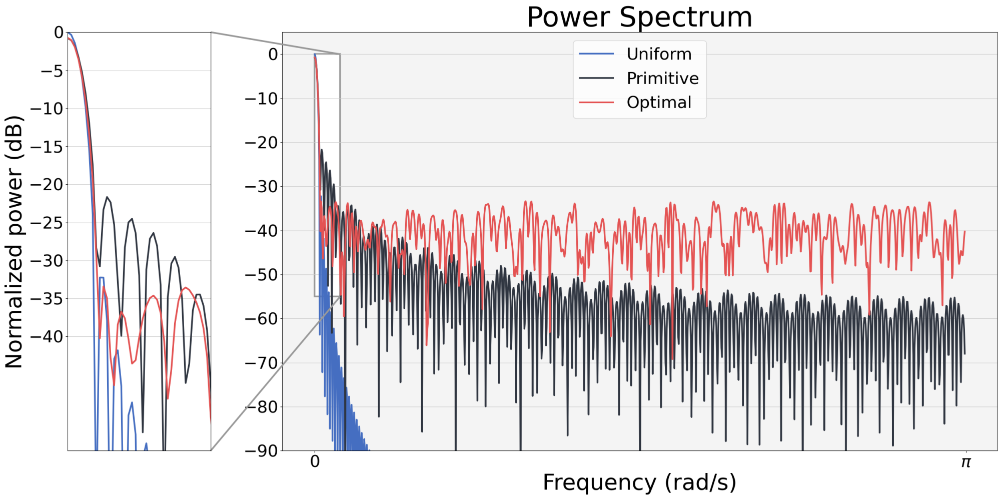


Figure 4.3: Figure showing the power spectrum of the Uniform, Primitive, and optimal sequence in the homogeneous case without correlation buffer.

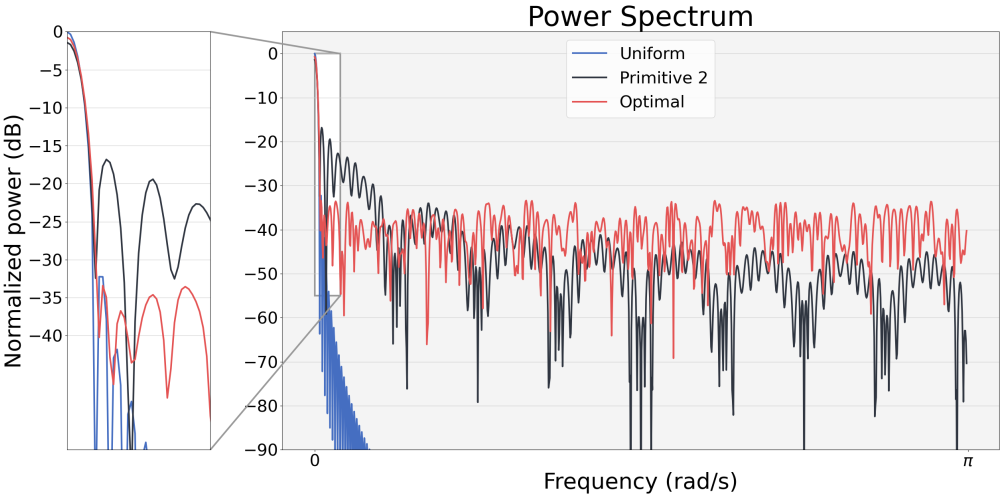


Figure 4.4: Figure showing the power spectrum of the Uniform, Optimal and the "Primitive 2" sequence in the homogeneous case without correlation buffer.

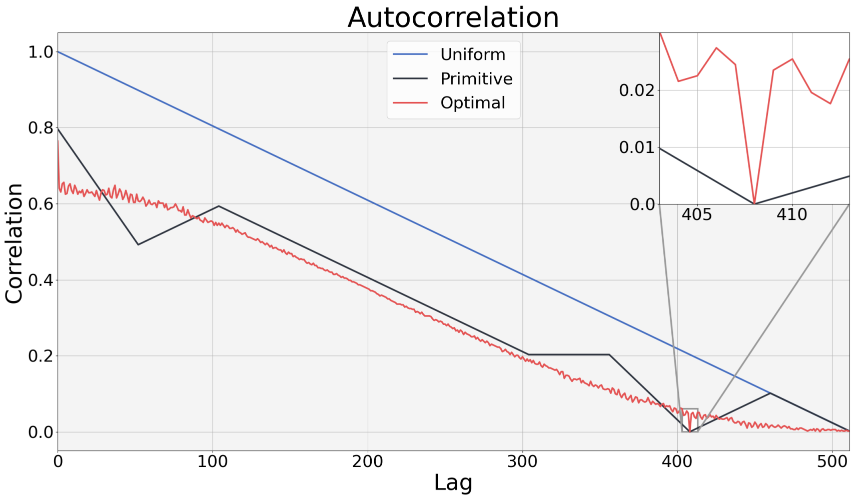


Figure 4.5: Figure showing the auto-correlation plot of the Uniform, Primitive and the Optimal sequence in the homogeneous case without correlation buffer. Only half of the auto-correlation results are presented, due to the symmetry of the autocorrelation function. The zoomed-in part of the plot clearly shows that the correlation at lag 408 is 0.

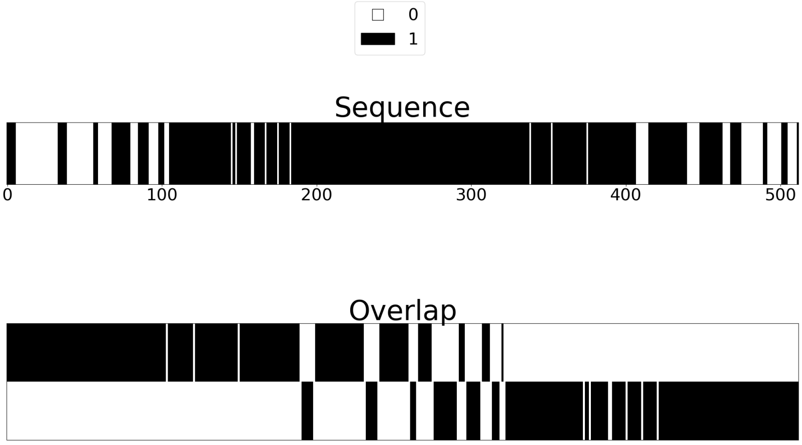


Figure 4.6: Figure showing a barcode plot of the optimal sequence for the homogeneous case with a correlation buffer, where black lines correspond to 1 and white lines correspond to 0. The upper plot shows the actual sequence and the lower plot shows the 103-point overlap where it can be seen that there is never more than one 1 placed at the same time.

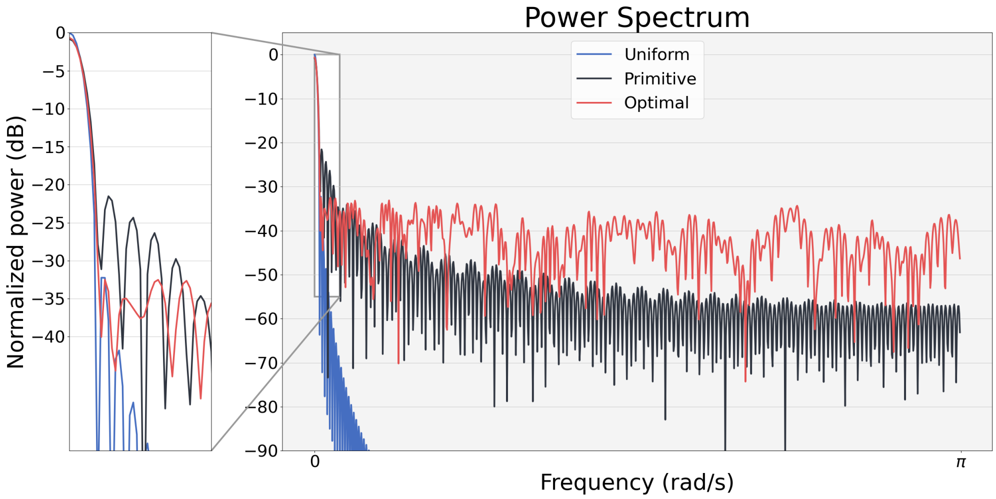


Figure 4.7: Figure showing the power spectrum of the Uniform, Primitive and the Optimal sequence in the homogeneous case with correlation buffer.

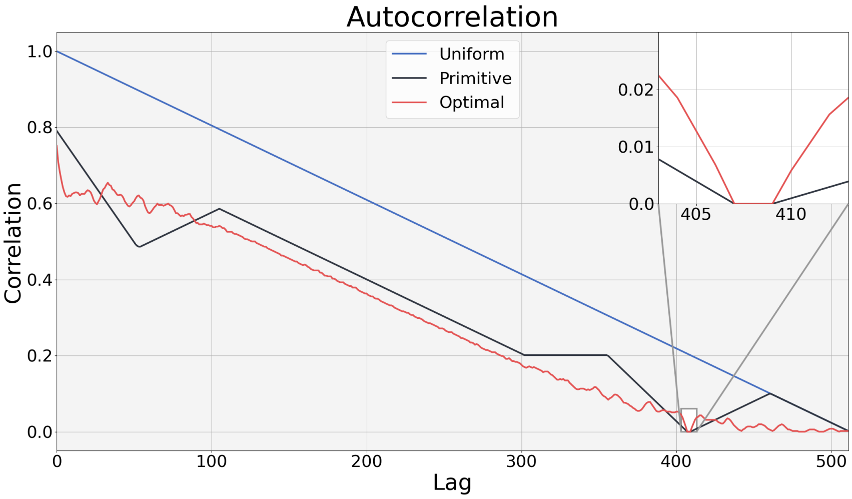


Figure 4.8: Figure showing the auto-correlation of the Uniform, Primitive and the Optimal sequence in the homogeneous case with correlation buffer. Only half of the auto-correlation results are presented, due to the symmetry of the autocorrelation function. The zoomed-in part of the plot shows that the correlation at lags 407-409 is 0.

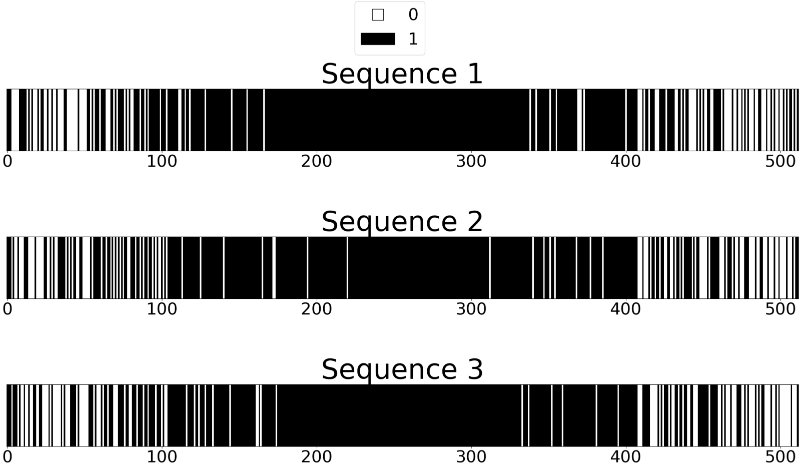


Figure 4.9: Figure showing barcode plots of the three optimal sequences in the heterogeneous case without a correlation buffer, where black lines correspond to 1 and white lines correspond to 0.

4.1.2 Heterogeneous Case Results

Here the results of the heterogeneous case sequences are presented, where three different sequences have been optimized to solve the problem. For each triplet of sequences the figures of merit are presented, along with the correlation plots and power spectra. The correlation plots shown are the correlations of each consecutive pair of the sequences, in accordance with the heterogeneous case description and Figure 3.1.

Without Correlation Buffer

The heterogeneous case sequences without correlation buffers were found optimal when using 122 zeros, as seen in Table 4.2. In Figure 4.10 it is seen that the correlation equals 0 at the relevant lag, and hence there are no interfered chirps at a CPI overlap of 103 lags between each of the sequences. Note that this is true for overlaps in the order shown in the method section for the heterogeneous case. From Table 4.2 it can be seen that the PSR values of the sequences found in the heterogeneous case generally is close to the PSR of the sequences found in the homogeneous case, and still notably lower than the PSR of the primitive sequences and on par with the PSR of the uniform sequence. The FWHM is close to that of the homogeneous solution, and each of the sequences in the heterogeneous case without a correlation buffer is generally similar to each other.

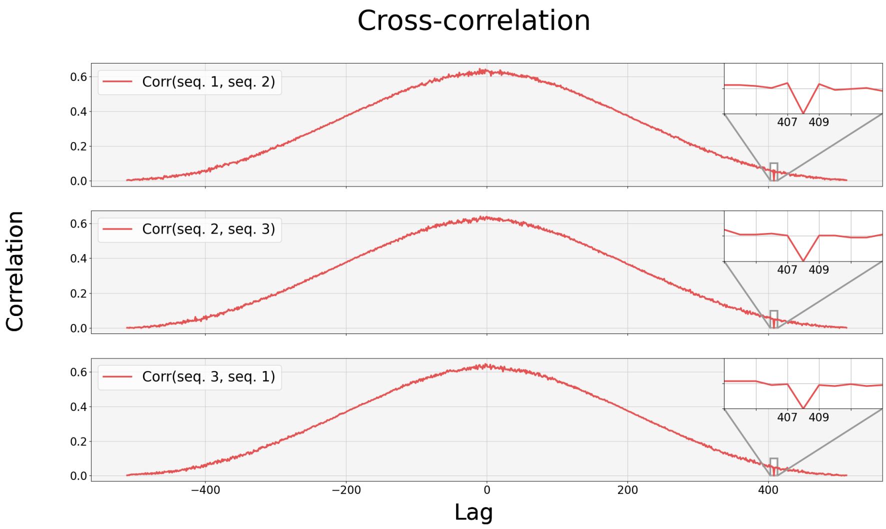


Figure 4.10: Figure showing the cross-correlation between the three optimal sequences in the heterogenous case using no correlation buffer. As seen in the zoomed-in part of the plot, the correlation at lag 408 is 0 in each case.

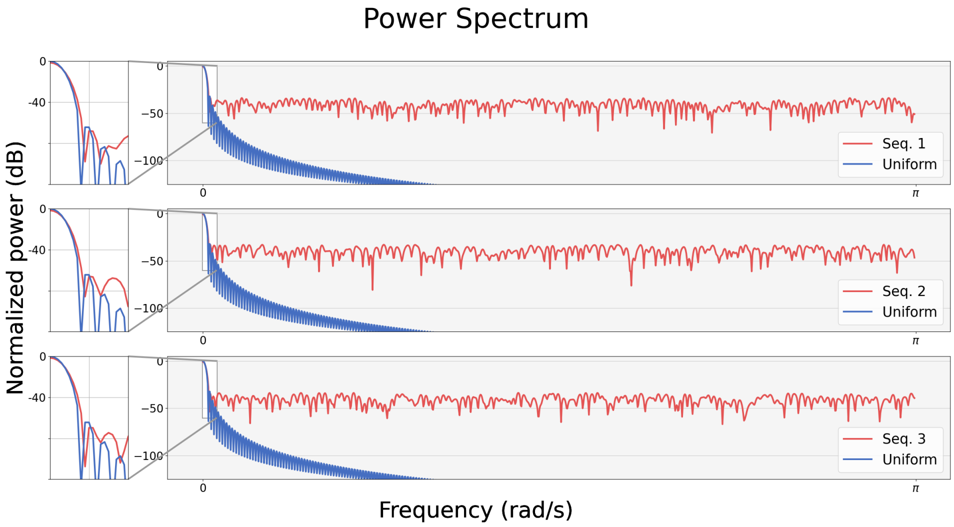


Figure 4.11: Figure showing the Power Spectrum of the three sequences along with the uniform sequence found in the heterogeneous case without correlation buffer.

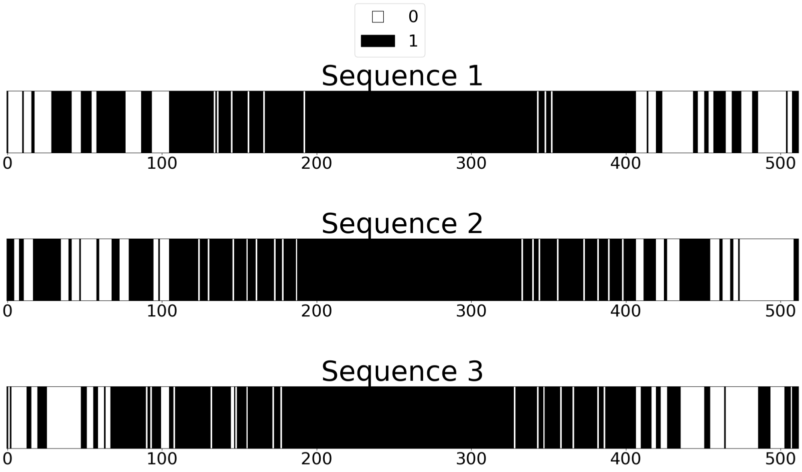


Figure 4.12: Figure showing barcode plots of the three optimal sequences for the heterogeneous case with a correlation buffer, where black lines correspond to 1 and white lines correspond to 0.

With Correlation Buffer

In the heterogeneous case, the sequences designed with a correlation buffer were found optimal when using 135 zeros, and it is seen in Table 4.2 that the PSR values of the sequences in the heterogeneous case with a correlation buffer are similar to the PSR values of the sequence in the corresponding homogeneous case, except for sequence 1 in the heterogeneous case with a buffer which has a slightly larger PSR value. This is also true for the FWHM values which are about the same size in both cases. Again, the method successfully has found sequences with no correlation at lags 407-409, which can be seen in Figure 4.12.

4.1.3 Descriptive Statistics

To get some descriptive statistics on the performance of the method, the homogeneous algorithm without buffer was run 500 times, from which the 95% confidence interval displayed in Table 4.3 on the number of zeros used was found. The lower and upper confidence bound CI_l and CI_u respectively were found using the expected value 120.474 and the standard deviation 3.099. A confidence interval for the energy of the final solution was also found as displayed in Table 4.4 where the expected value was found as -152.666 and the standard deviation 1.035.

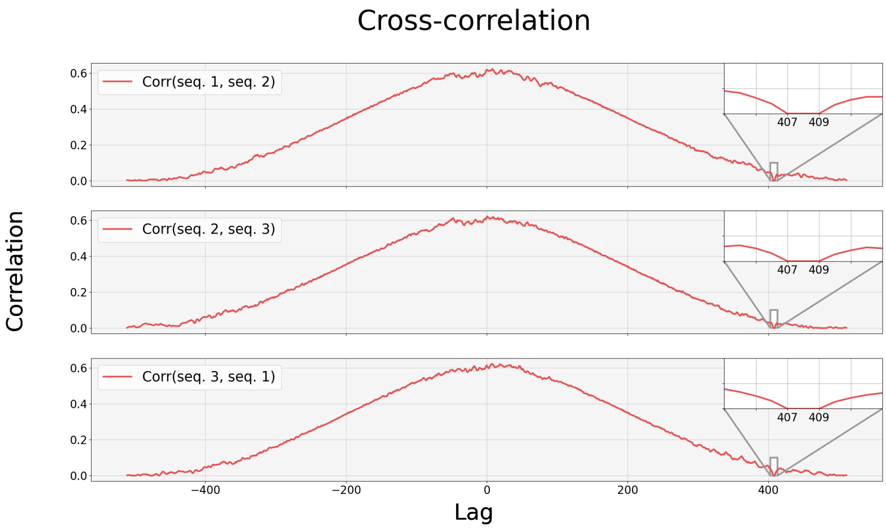


Figure 4.13: Figure showing the cross-correlation between the three optimal sequences in the heterogeneous case with a correlation buffer. As can be seen in the figure the cross-correlation is 0 at the specified lag 408 and the surrounding lags 407 and 409.

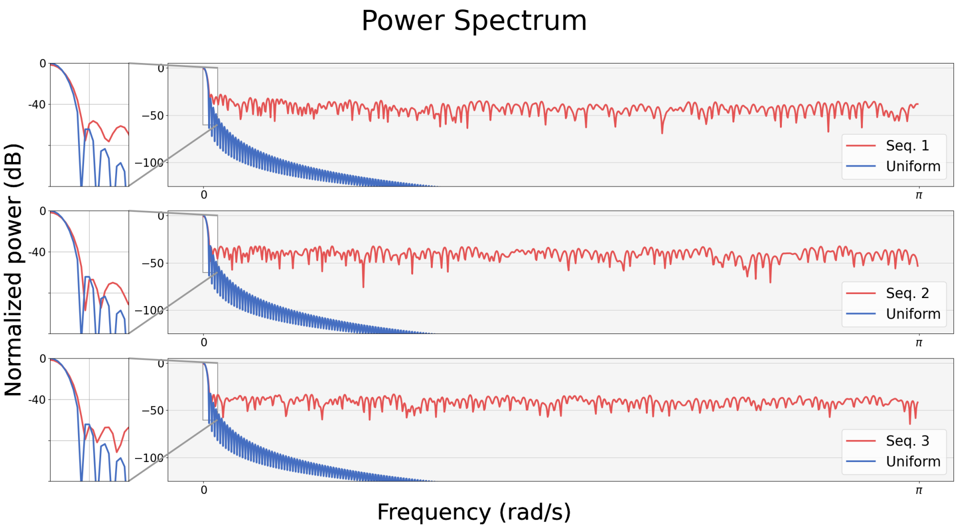


Figure 4.14: Figure displaying the Power Spectrum of the three optimal sequences in the heterogeneous case with a correlation buffer.

Table 4.3: Table displaying a 95% confidence interval of the number of zeros k used in the final solution found when running the algorithm. The confidence interval is based on 500 iterations of the algorithm.

CI_l	$E(k)$	CI_u
120.201	120.474	120.747

Table 4.4: Table displaying a 95% confidence interval of the energy of the final solution found when running the algorithm. The confidence interval is based on 500 iterations of the algorithm.

CI_l	$E(\text{energy})$	CI_u
-152.757	-152.666	-152.576

4.2 Coprime Method

In this section results from using the coprime method are presented. Coprime sequences of length 513 were found since these were close in length to the ones found by using Simulated Annealing. The similarity in length was needed to enable comparison between the methods.

The pairs of coprime integers P and Q found, which through the coprime method achieve sequences of length 513, are presented in Table 4.5. In Figure 4.15 the power spectrums resulting from applying the FFT to each coprime sequence are presented. Before applying the FFT to the sequences, each sequence has been multiplied with a Hann window. In these spectra, the power spectrum resulting from the FFT of the homogeneous case optimal sequence without correlation buffer is included as well, seen as the red curve. In the power spectrum it can be seen that for each coprime integer pair, except for $(P, Q) = (2, 171)$, aliasing of the spectral peak occurs at lower angular frequencies than compared to the frequency peak in the power spectrum of the optimal sequence. Further on the power spectrum of sequence $(P, Q) = (2, 171)$ is displayed in Figure 4.15 is plotted again but on its own in Figure 4.16, in which it can be noted that the peak at the far right end of the spectrum, close to π at the frequency axis, is not an aliased peak but is instead a sidelobe. Note that the sidelobes of the coprime sequence are notably higher than those of the optimal solution and that the mainlobe has quite low power, due to the very few ones compared to the number of ones in the optimal sequence. Lastly, in Figure 4.17 the auto-correlation of the coprime sequence designed with $(P, Q) = (2, 171)$ is displayed, where it can be seen that the correlation fluctuates between 0 and 1 after about lag 350, and notably the correlation is 0 at lag 408.

Table 4.5: A table displaying the pairs of coprime integers P and Q used to construct coprime arrays.

Pair	P	Q
1	2	171
2	5	57
3	10	27
4	14	19

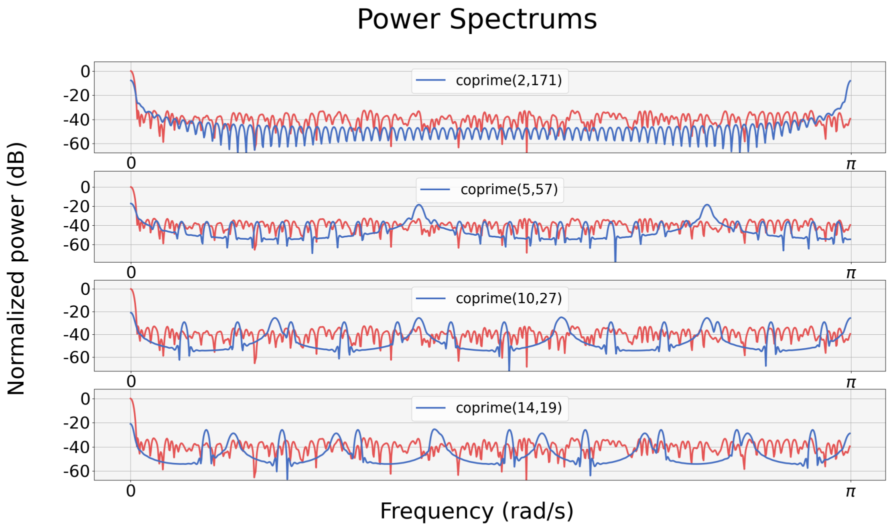


Figure 4.15: Figure displaying the Power Spectrum of the four sequences found using the coprime method(blue), along with the homogeneous case optimal sequence without buffer(red).

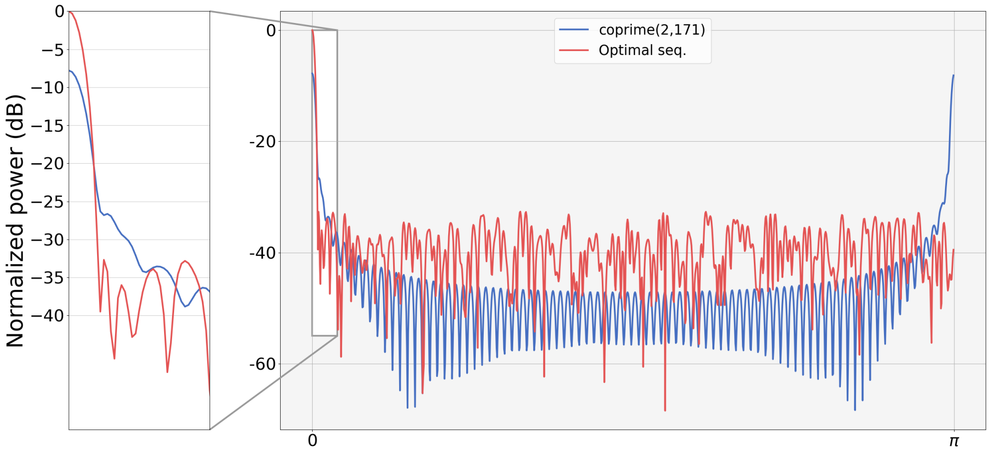


Figure 4.16: Figure displaying the Power Spectrum of the $P = 2$ $Q = 171$ sequence found using the coprime method.

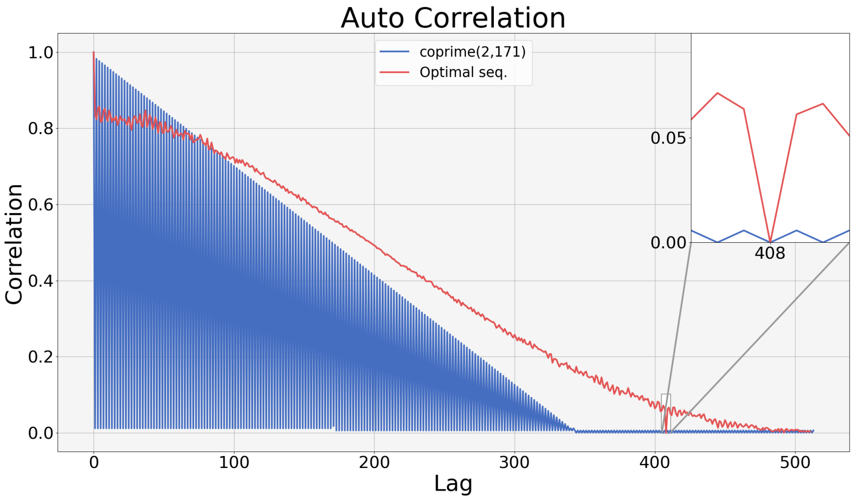


Figure 4.17: Figure displaying the Auto-Correlation of the $P = 2$ $Q = 171$ sequence found using the coprime method, along with the auto-correlation of the homogeneous case optimal sequence without buffer.

Table 4.6: Table displaying the minimum velocity difference at which the probability of detecting 2 targets equals 1 for the primitive, short and homogeneous optimal sequence.

	Without Buffer (Δv)	With Buffer (Δv)
Uniform	0.0052	0.0052
Optimal	0.0056	0.0058
Primitive	0.0062	0.0063
Short	0.0066	0.0067

4.3 Monte Carlo Simulations

In this subsection, Monte Carlo simulations of the velocity resolution are presented to show how the probability of separating two targets with slightly different velocities changes depending on the sequence used when transmitting chirps. The sequences found by the simulating annealing algorithm are tested along with the primitive sequence as well as a uniform sequence which is shorter than the optimal sequences. Remember that the primitive and optimal sequences are of length 512, and that it is assumed that the overlap between the sequences is 103 lags. The short sequence is an alternative to having this overlap, in essence removing the 103 lags that overlaps, and hence getting a shorter sequence but without any overlap and interference. Thus, the short sequence is of length 409.

In Table 4.6 the minimal velocity difference Δv needed to resolve two objects almost surely is shown. These values correspond to the values at the x-axis in Figure 4.18 where the curves first reach $P(\text{resolved}) = 1$. In both the table and the figure it is seen that there is no notable difference in the achieved velocity resolution when using a correlation buffer or not. However, the achieved velocity resolution differs between the sequences, where the optimal sequences can separate two targets with a smaller velocity difference than the primitive and short sequences.

In the heterogeneous case, the corresponding results have been calculated for each of the three sequences with and without a correlation buffer. As can be seen in Figure 4.19 there is no substantial difference between the performances of the different sequences regardless of whether using the correlation buffer or not. Each of the sequences performs equally well in resolving 2 targets as the optimal sequence in the homogeneous case.

These results are based on simulated data with added noise and with the assumption that the two targets are located at the same radial range. In a real-life scenario, the actual velocity differences needed will probably differ from these results, depending on for example the receiver noise figure. Hence, the key takeaway from this Monte Carlo simulation is not the exact values of the velocity differences

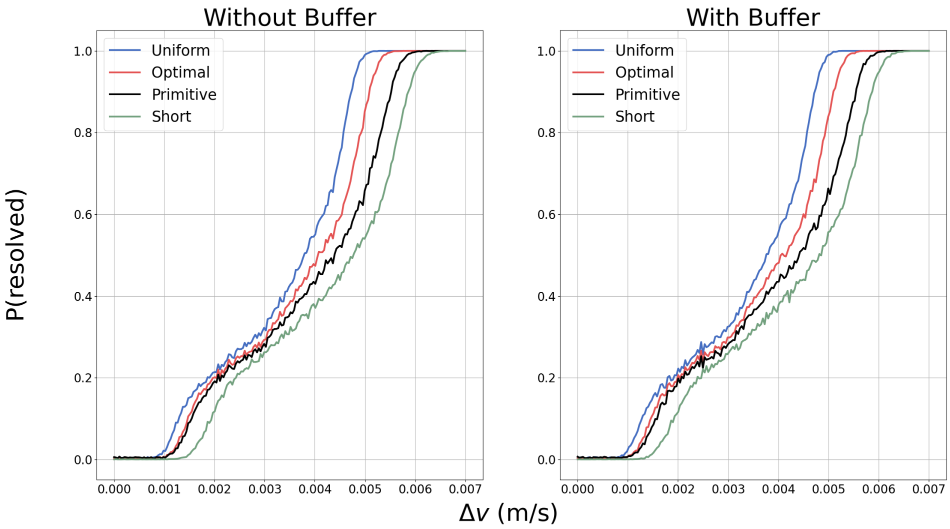


Figure 4.18: Figure showing the probability of detecting 2 targets as a function of the velocity difference between these targets when transmitting chirps according to the different sequences in the homogeneous case.

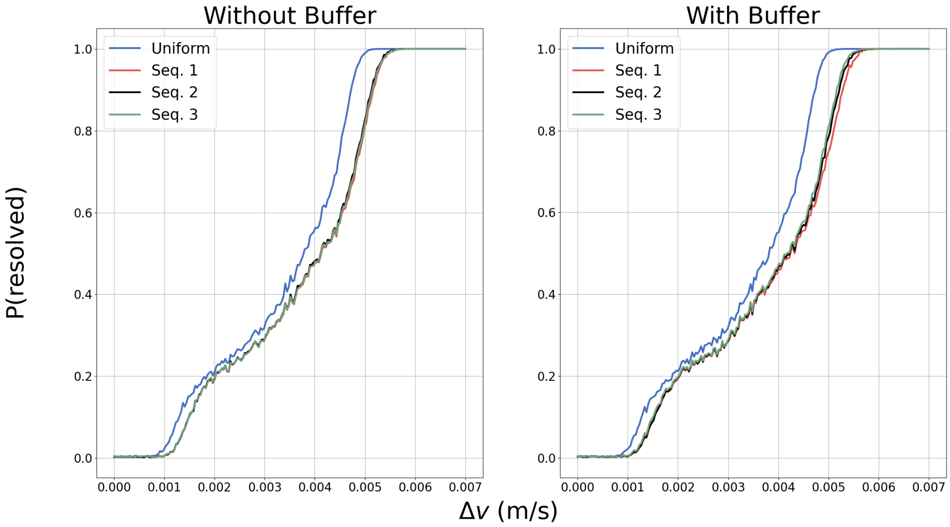


Figure 4.19: Figure showing the detection probability as a function of the velocity difference Δv of the three optimal sequences found in the heterogeneous case

needed to separate the peaks, but rather shows that there will be a difference in velocity resolution when transmitting chirps according to the sequences presented.

4.4 Method Convergence

In this section, some results of the P_1 method used to find the optimal order of a given number of ones and zeros are presented. In Figure 4.20 the energy of the P_1 Simulated Annealing method, introduced in the method section, can be seen. Obviously the energy decays and converges to some limit where the method cannot find a solution with a lower energy state. From this plot, it is also seen that the method is more prone to accepting worse solutions during the early iterations than during the later iterations, which is expected and is a result of the method's cooling schedule. Finally, in Figure 4.21 one example of the objective function in P_2 is shown, where the energy resulting from the P_1 Simulated Annealing is plotted for each number of zeros. Due to the weights set, where the correlation has a high weight, the energy is high for a low number of zeros where it is impossible for the P_1 Simulated Annealing method to find a solution with 0 correlation at the specified lag. For some number of zeros, about 120, the P_1 Simulated Annealing is able to find solutions with 0 correlation at the specified lag. For solutions using more than 120 zeros the energy again starts to increase due to the negative effects the many zeros have on the spectral quality, which increases the FWHM and PSR values.

When looking at the results presented in Figure 4.20 and 4.21, remember that the method used is partially stochastic, and hence the energy plots and the objective function shown are only instances of all possible energy plots and objective functions. Probably each energy plot and objective function will be very similar between iterations and similar to the results plotted here, but some differences will occur due to the stochasticity.

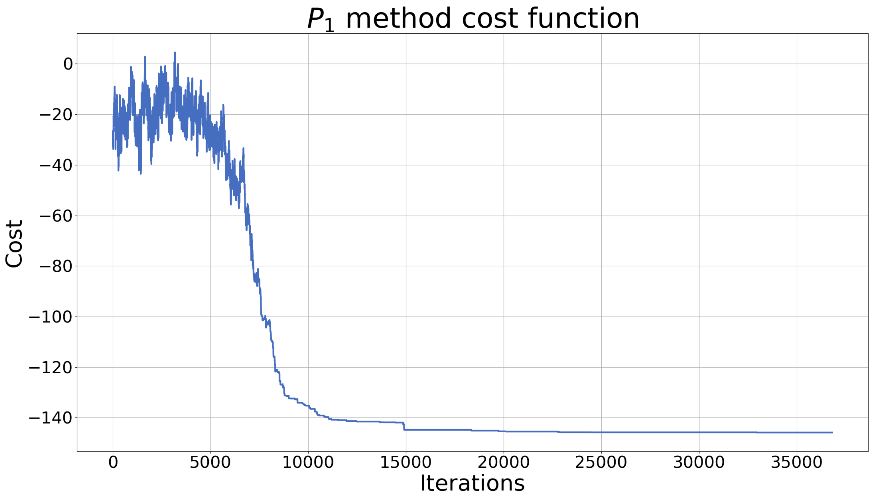


Figure 4.20: Figure displaying the energy at each iteration of one run of the P_1 Simulated Annealing algorithm. As can be seen in the plot the energy fluctuates around some decaying mean. The intensity of the fluctuations also seems to decrease with each iteration.

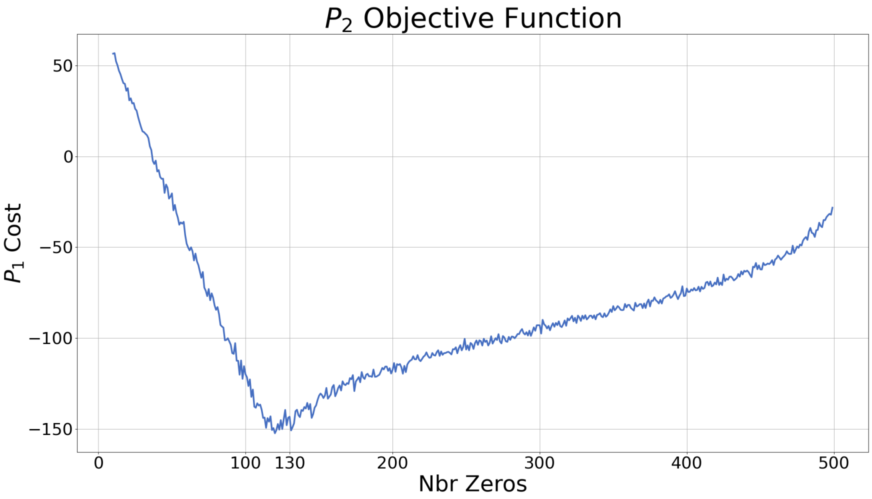


Figure 4.21: Figure showing one example of the actual objective function which the exhaustive search in the P_2 problem tries to minimize.

Chapter 5

Discussion

5.1 Optimal sequences

From the results section, it is obvious that the simulated annealing algorithm successfully finds sequences that, at a 103 lag overlap, have no interfering chirps since the correlation in each case is 0, both in the homogeneous and heterogeneous cases. Sequences with and without correlation buffers are found, and hence the method has proven itself useful in providing sequences orthogonal at both single and multiple lags. This is of course a nice result, but the problem of finding orthogonal sequences also included optimizing the sequence's spectral properties. When regarding the spectral aspects achieved when optimizing solutions to this problem, the method still performs quite well, and this can be seen in Tables 4.1 and 4.2 and in the spectral plots of the result section. An effect of taking the spectral quality into account is seen in the PSR values presented, which shows that the sequences found generally have a Peak Sidelobe Ratio close to that of the uniform solution. For the optimal sequence without any correlation buffer in the homogeneous case, the PSR is even lower than the PSR of the uniform sequence. As pointed out before, the PSR of each found solution is close to −30 dB as shown in Table 4.1, which is much lower than the PSR of the primitive solution which has a value of about −20 dB, and hence a radar transmitting according to the optimal sequences will be about as probable as a radar transmitting according to a uniform pattern in detecting targets at different ranges with different powers. On the other hand, a radar transmitting according to the primitive sequence is less likely to detect a small target in the presence of a larger target, due to the risk of the small target's mainlobe ending up below the larger target's highest sidelobe and remaining undetected. Both the primitive and the optimal solutions achieve orthogonality at the overlap 103 lags corresponding to about 20%, but get quite different spectral properties. Thus it is evident that sequences that are orthogonal in the sense of not causing interference might be better or worse with respect to the spectral quality. This might seem like a basic result, but at the same time it validates the research question posed at the beginning of the thesis: there is a point in trying to optimize the sequence with regard to both orthogonality and spectral quality.

Continuing on the discussion on the spectral quality, the difference between the sequences is the number of 0's inserted and the order in which these are placed. In Figure 4.4 we can compare the "Primitive 2" sequence with the optimal sequence. Both of these have the same number of 0's, and they both have a correlation of 0 at lag ±408, but the positions of the zeros differ. In the figure it is seen that the two sequences, despite their similarities, yield power spectrums with different PSR values. Apparently, the order of the zeros in the sequences is not only important to the correlation of the sequences but also has quite a large impact on the Peak Sidelobe Ratio and thereby the detection probability. Thus it seems like both the number of zeros inserted and the order of these affect the PSR which in turn can affect the detection probability of targets as explained in section "2.5.4 Constant False Alarm Rate".

Following the previous discussion, it might be easy to get the impression that it is beneficial to use fewer chirps than necessary. This might be true from a PSR perspective, but it comes with the cost of losing integration gain. Losing integration gain will affect the PSR since the mainlobe amplitude will decrease, but the decrease in PSR can be compensated for by arranging the 0's in a clever way, as discussed above. However, the loss of mainlobe amplitude following the loss of integration gain cannot be accounted for. This lower amplitude will lead to a lower SNR and hence a smaller detection probability in the presence of thermal noise. This implies that for example the detection range will be impaired since the signal peak amplitude is lower and the amplitude difference between the signal peak and the detection threshold is lower than in the uniform case.

From the problem formulation in the method section, we know that we want to keep the first and last chirp in order to preserve the velocity resolution, which depends on the total CPI time. This has been included in the problem optimization as a fixed constraint, such that each optimal sequence has as long CPI time as the uniform sequence. According to the theory, the length of a signal affects its mainlobe width, which in turn will have an impact on if two targets can be separated or not. However, from the FWHM values in the result section, it is seen that the values of the peak width at half maximum vary between the uniform, optimal and primitive sequences, even if these have the same length. This might seem a bit odd, but it is also in line with the results from the Monte Carlo simulations. In the Monte Carlo simulations, we can see that using the optimal sequence yields a higher probability of the system being able to separate two targets than using the primitive sequence, despite these having the same CPI time. This indicates that the mainlobe width, and thereby the velocity resolution, is affected by more than the CPI time. This is also supported by the fact that both the optimal and primitive sequences have a worse velocity resolution than the uniform sequence when again each of these sequences has the same CPI time. This result is quite unexpected, and unfortunately can not be explained given the theory or results in this thesis. However, it is believed that this increased mainlobe width is due to the applied Hann window, which is designed for uniformly sampled sequences. This is however not certain and needs to be investigated further.

The FWHM figure of merit was introduced to the problem after it was realized that the constraint on having chirps at the first and last position of the CPI was not sufficient to preserve the velocity resolution. However, it was noticed that the metric had little to no effect on the mainlobe width at -3 dB, and hence had little effect on the velocity resolution as well. Hence, the weights used when generating the sequences presented in this work were set quite low compared to the other metric's weights.

From the Monte Carlo simulations, we get the expected result that using a short uniform sequence yields a worse velocity resolution than using the sparse optimal sequences. This means that with the effort of implementing the solution some velocity resolution is gained compared to the alternative solution of avoiding interference by just using a shorter uniform sequence.

The idea of using correlation buffers in the solution came from the awareness of the difficulties it means to try and implement a solution with sparse CPIs, since this would demand perfectly time-synched chirp times and no time drift between radar units. The correlation buffer can aid in implementing such a solution by offering some flexibility in chirp time configuration and clock drift since an offset of one chirp in any direction would not cause interference. It has also successfully been tested in the work of this thesis to use more than one buffering correlation lag, however these results are not included in the report. This buffer comes with the cost of even more loss in integration gain, since more 0s in the sequence are needed to successfully find sequences orthogonal at more than 1 lag. At the cost of having to find sequences with a correlation buffer, the gain compared with using a shorter CPI seems to be a slightly better velocity resolution.

Since we use three sequences instead of one in the heterogeneous solution we get more data points to organize. This both adds more degrees of freedom to the problem but also increases its complexity. A direct comparison between the results of the two cases might therefore be erroneous since the two solutions are given equally many iterations, since the cooling schedule is the same in the two problems, but are of different sizes. Preferably the heterogeneous solution would be given more iterations to enable a more fair comparison, due to it being larger than the homogeneous solution. It would probably be possible to find similar results and draw conclusions thereafter using only 2 different sequences in the heterogeneous solution, instead of 3. The difference though is that using 2 sequences would make each sequence dependent on the other in both the beginning and the end. Using three sequences instead makes for example sequence 2 dependent on sequence 1 in the beginning, but dependent on sequence 3 in the end. This makes for some difference, and the case of using 3 sequences is more general than using 2, since the three-sequence case can be expanded into using many more sequences.

When discussing these results it is important to remember that the method used, Simulated Annealing, in no way guarantees to find the exact optimal solution, but rather approximates the optimal solution. However, the results found by the algorithm are closer to optimal in the sense of minimizing the cost function defining the problem, compared to the primitive and coprime solutions. Since the method is stochastic, two confidence intervals are presented as there is some variation in the solutions found. The confidence intervals of the number of zeros used in the solution show that the P_2 method, which tries to optimize the number of zeros used, usually finds 120 zeros optimal, with some variation. Also, a corresponding confidence interval of the energy of the final solution is presented to show that there also is some variance in how the zeros are ordered. This is a very expected result due to the nature of the method. However despite this stochasticity, the variation of the method does not seem too large as the confidence bounds are relatively narrow.

5.2 Coprime method comparison

In the literature, several codes or sequences which are designed to have low autocorrelation are presented, such as Gold-, Barker and MPS sequences, which all are introduced in the theory section. Each one of these are bi-phase sequences, consisting of elements having values 1 or -1. In this work binary sequences consisting of ones and zeros have been used to represent when, in slow time, a radar transmits a chirp or not relative to a uniform grid, hence making the chirping pattern non-uniform. The process of finding orthogonal chirping patterns with as good as possible spectral properties, that is dynamic range and mainlobe width, has essentially consisted of optimizing these binary sequences. Since there already are sequences from previous research the initial idea was to investigate how these sequences would work out as chirping patterns in a radar. However, due to the small but important difference in using 1 and -1 elements or 1 and 0, such a comparison becomes difficult. This is due to the problem of finding a reasonable analogy to using -1 elements in a chirping pattern, instead of using zeros. The physical interpretation of using -1 entries in a sequence in the context of this thesis could be to use chirps with a negative frequency slope. Another possible interpretation of using -1 entries in the sequence could be transmitting a signal with a phase shift of π in a PMCW radar context. However, both of these possible cases are out of the scope of this thesis.

Due to the difficulties of comparing the optimal sequence with some bi-phase sequences, it was instead decided to compare the optimal sequence with a sequence found using the method of constructing coprime sparse arrays, as explained in the theory section and in [38]. From the theory section on maximum velocity, it is known that the maximum unambiguous velocity depends on the minimum distance between two chirps. When trying to find suitable coprime sequences it turned out that many of these had a much lower unambiguous maximum velocity, due to aliasing resulting from large distance between the chirps. Due to this, it was decided to compare the optimal sequences to the coprime sequence designed with the pairs $(P, Q) = (2, 171)$, which was close in length and had a similar maximum unambiguous velocity.

In the result section figures of the spectrum of this sequence alone are shown, where it is seen that the sidelobes of this coprime sequence are much higher than the ones of the optimal sequence. Notably, the autocorrelation of the sequence is 0 at lag 408. This would allow two radars to operate without interference at a time overlap corresponding to 103 chirps. However, choosing between using the coprime sequence or the optimal sequence, choosing the optimal sequence is advantageous due to it having more ones, hence more chirps and a better integration gain and dynamic range. The peak at the far right end of the power spectrum of the coprime sequence is not an aliased mainlobe, but rather a sidelobe. The fact that this sidelobe appears and has a very high amplitude would make it difficult to use the coprime sequence. To avoid erroneous measurements or ambiguities a limit on the maximum measurable velocity would be needed to not include this large sidelobe in the measurement. The choice of using the coprime method to design a sequence was due to it being able to construct large sequences, as well as being easy to implement.

5.3 Method

The problem was formulated as concerning two radars transmitting the same sequence, or three radars transmitting different sequences. In both cases, the radars were assumed to be separated by a sufficiently small range, such that the time the radio waves travel between the units can be omitted, It was also assumed that no reflective objects were included in the scene. These assumptions made it possible to state the problem as finding sequences causing no interference at one specific overlap, which effectively mean that the correlation at one lag was supposed to be 0. If some targets were introduced, the problem would have to be reformulated, as more than one specific overlap would be concerned. The waves reflected at the target would travel a longer distance than the direct path, and thereby it would be interesting to decrease the interference at all lags in the range $[408, 511]$. Probably, the method method used in this thesis could be used for this purpose as well by including more lags in the Correlation Sum metric.

The problem was divided into the cases homogeneous case and heterogeneous case in order to investigate how the method would respond to the additional freedom of having more sequences. From the results section and the discussion on the results it is known that this added freedom in the heterogeneous case does not achieve as good spectral properties as the homogeneous solution when finding a sequence with a correlation buffer. It is hard to draw any conclusions about why this is, but it can be speculated that adding more freedom to an already large problem becomes too computationally heavy and that it is possible that the heterogeneous case sequences would achieve better spectral properties than the homogeneous sequence if the method would be allowed to iterate more times.

As explained in the theory section on the Simulated Annealing algorithm, it is well known that the algorithm is easy to implement and popular to use in combinatorial optimization problems, but with the drawback of being hard to choose optimization parameters to. In this work, the Simulated Annealing algorithm has been used to solve the P_1 problem of ordering of a number of zeros in a sequence.

As stated in the method section, the problem is discrete and non-convex, which omits any possibility of using a convex optimization method. The problem is also non-linear, due to the max-function used in the PSR cost function. Due to the algorithm being frequently applied to combinatorial and non-linear problems, the Simulated Annealing algorithm seemed like a convenient choice. Looking at the convergence plot in Figure 4.20 in the result section, it is obvious that for a given number of zeros to use in a sequence, the algorithm succeeds in organizing the sequence such that the total cost decreases with each iteration. At the same time, the plot also shows when a worse solution is accepted and the cost is momentarily increased. As such, it seems like the method works as it was supposed to since it tries to minimize the cost function while also trying to avoid local minima.

The problem of finding the number of zeros is very different from finding the best ordering of the zeros, mainly because finding a neighboring solution is only a matter of increasing or decreasing the number of zeros used in the solution and hence this problem is not combinatorial. This allows for a wider range of possible methods to use to solve the problem, out of which the exhaustive search is one. The exhaustive search evaluates each option, and hence it is very time-consuming. If the sequence length would increase, an alternative and faster method would probably be necessary.

As stated in the theory section the issue of choosing the parameters of the algorithm remains. In this work the temperature parameters, which govern both the probability of accepting worse solutions and the number of iterations have been chosen by repeatedly testing different combinations and evaluating if the method gets enough time to converge and if the probability of accepting worse solutions decreases enough. The acceptance rate of worse solutions also depends on the cooling schedule, which also can be tricky to choose. In this work, the cooling schedule has been chosen as geometrically decreasing, as

$$T_{n+1} = T_n \alpha \tag{5.1}$$

with α being close to 1, which ensures slow convergence. The difficulty in choosing the cooling schedule is due to the convergence time, a too fast convergence might amount to the method finding a local minimum and getting no opportunity to avoid getting stuck due to the fast decreasing probability of accepting a worse solution. On the contrary, a slow-decreasing cooling schedule implies that the method uses many iterations and requires a lot of time before converging. In this work, a slowly decreasing cooling schedule has been used since it is preferred to find the best solutions possible rather than finding a solution quickly. From tries with different alpha values it also became apparent that a value close to 1 generally led to solutions with lower energy, which is preferred.

The figures of merit used in the problem formulation, which the method is made to minimize, were the Full Width at Half Maximum (FWHM), Peak Sidelobe Ratio (PSR) and Correlation Sum (CS). These were chosen to find sequences without any interfering chirps at a given overlap while preserving as much as possible of the velocity resolution and dynamic range. In similar problems, the PSR has been used, for example in designing sparse antenna arrays, and hence it seemed like a suitable metric to use. The FWHM is a common metric to use to determine the width of a spectral peak, and hence this is used with the same purpose in this work. The Correlation Sum was used to measure the number of interfering chirps between the sequences. The discrete correlation is commonly used to measure the similarity between signals which thus makes it a reasonable choice. The weights have been chosen both from experience from using the method and empirical tries with different combinations, but also from the idea that a sequence having a Correlation Sum different from 0 is useless since it will cause interference. The amount of interference will in reality depend on how the CPIs overlap. In the setting of this work though, where the CPIs are assumed to have a time overlap equal to a multiple of T_c, the chirp time, interference occurs whenever the correlation sum is not 0. The higher the correlation sum, the more chirps are interfered, but the end goal was to find sequences causing no interference, and hence no interfered chirps were tolerated. Given this, the CS weight was set to imply a high cost if the correlation was different from 0 to guarantee that there would be no correlation between the sequences. In Figure 4.21 the result of setting the CS weight higher than the others can be seen. This aids in finding sequences without any correlation, given the Correlation Sum presented in the results section.

To evaluate the method some statistical measures were estimated using Monte Carlo Methods. These statistics show that there is some variance in the method, which is expected due to the stochastic part of the Simulated Annealing Algorithm. Despite this variance, it seems like the method finds similar results in each iteration, and hence it also seems that the method converges to an approximated optimal solution.

Monte Carlo simulations were also used to show the difference in velocity resolution as a result of transmitting chirps according to the optimal, the primitive and the short sequences. As mentioned in the results section, the values presented show that there is a difference, but the values presented are not the exact velocity differences that would be needed when actually transmitting these sequences in a real test. This is due to the definition of “separable peaks”, which in real radar systems are implemented using, for example, a OS-CFAR as explained in the theory section. Running the simulation again with another definition of what is a peak, and when they can be separated, will affect the result. However, the important result is not the difference itself, but the fact that there is a difference that depends on the sequence used when transmitting chirps. In the Monte Carlo simulations it has also been assumed that the two targets are located at the exact same range, which in reality is very rare. For the purpose of showing this difference however, the definition of a peak and when two peaks can be separated, as well as the assumptions of the targets being present at the same range is reasonable.

5.4 Further Research

While this thesis might have answered some questions, new questions have arisen and in this section, some further research questions which would be interesting to investigate are presented.

The most obvious and maybe largest research question that would be interesting is to investigate the effect of transmitting sparse CPIs, such as those presented in this thesis, using other means of spectral estimation for range and velocity measurements. To limit the work in this thesis, using the DFT was set as a premise for the work. This was also done since this is a common technique used for spectral estimation. However, when removing chirps from a CPI some of the properties of the detection will be worsened. By removing chirps the integration gain is decreased and thereby it becomes harder to maintain a good SNR and high dynamic range. Using other spectral estimation techniques, such as high resolution techniques, could however aid in this regard. Examples of methods that could be tried are the non-parametric Non-Uniform FFT (NU-FFT) [41], or the parametric methods Iterative Adaptive Approach (IAA) [42, 43] and MUSIC algorithm [44]. The IAA method is also supposed to work well with sparsely sampled data.

It would also be interesting to investigate the usage of other optimization methods. The choice of using the Simulated Annealing method for the problem referred to as the P_1 problem seems like a reasonable choice. However, for the P_2 problem, other methods could possibly be more suitable. The P_1 problem is combinatorial, and the value of the cost function depends only on the order of the zeros and ones in the sequence. The P_2 problem's objective function depends on the combinatorial P_1 problem, but in hindsight looking at the objective function, it seems like other methods, such as the bisection method, possibly could be more suitable to use.

Chapter 6

Conclusion

Following the results and discussion, the first important conclusion to be made is that it is possible to find orthogonal sequences with different spectral properties, and as such it is also reasonable to try and find the orthogonal sequence with the best spectral properties. What are "good" spectral properties in this thesis is determined through the figures of merit, premiering a high dynamic range and narrow main lobe width, and sequences having these properties are also found using the Simulated Annealing method. The method succeeds in finding orthogonal sequences with spectral properties better than the simple primitive solutions, whether be it with a correlation buffer or not. The Simulated Annealing method on the other hand is not guaranteed to find the optimal solution, but rather approximates the optimal solution. Hence it is likely that there exists orthogonal sequences with better spectral qualities than the ones found, but they will also probably be similar to the sequences presented in this thesis. Even if the method used finds satisfactory results it is possible that other methods might perform better. However, given the complexity of the problem, the use of this method seems reasonable. In the thesis, it is also shown that transmitting signals in an orthogonal pattern is beneficial compared to transmitting shorter CPIs due to the gain in velocity resolution. However, implementing the solution will involve having to solve problems such as clock drift and perfect sync between units. If solving these issues is worth the gain in velocity resolution is for the user to decide.

References

[1] Tapan K. Sarkar and Magdalena Salazar Palma. A history of the evolution of RADAR. In *2014 44th European Microwave Conference*, pages 734–737, 2014.

[2] Merrill Skolnik. An introduction and overview of radar. *Radar Handbook*, 3:1–1, 2008.

[3] Rony J James. A history of radar. *IEE Review*, 35(9):343–349, 1989.

[4] Roger C Whiton, Paul L Smith, Stuart G Bigler, Kenneth E Wilk, and Albert C Harbuck. History of operational use of weather radar by US weather services. Part I: The pre-NEXRAD era. *Weather and Forecasting*, 13(2):219–243, 1998.

[5] Christian Waldschmidt, Juergen Hasch, and Wolfgang Menzel. Automotive radar—from first efforts to future systems. *IEEE Journal of Microwaves*, 1(1):135–148, 2021.

[6] M Vogt. Radar sensors (24 and 80 GHz range) for level measurement in industrial processes. In *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4. IEEE, 2018.

[7] Stefano Pisa, Erika Pittella, and Emanuele Piuzzi. A survey of radar systems for medical applications. *IEEE Aerospace and Electronic Systems Magazine*, 31(11):64–81, 2016.

[8] Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. Through fog high-resolution imaging using millimeter wave radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11464–11473, 2020.

[9] Maria S Greco, Jian Li, Teng Long, and Abdelhak Zoubir. Advances in radar systems for modern civilian and commercial applications: Part 1 [from the guest editors]. *IEEE Signal Processing Magazine*, 36(4):13–15, 2019.

[10] Sian Jin and Sumit Roy. FMCW radar network: Multiple access and interference mitigation. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):968–979, 2021.

[11] Canan Aydogdu, Musa Furkan Keskin, Gisela K Carvajal, Olof Eriksson, Hans Hellsten, Hans Herbertsson, Emil Nilsson, Mats Rydstrom, Karl Vanas, and Henk Wymeersch. Radar interference mitigation for automated driving: Exploring proactive strategies. *IEEE Signal Processing Magazine*, 37(4):72–84, 2020.

[12] Faruk Uysal. Synchronous and asynchronous radar interference mitigation. *IEEE Access*, 7:5846–5852, 2018.

[13] Cesar Iovescu & Sandeep Rao. The Fundamentals of millimeter wave radar sensors. *Texas Instruments*, pages 1–7, 2020.

[14] Mark A Richards, Jim Scheer, William A Holm, and William L Melvin. *Principles of modern radar, Vol. I: Basic Principles*, volume 1. SciTech Publishing, 2010.

[15] Mark A Richards et al. *Fundamentals of radar signal processing*, volume 1. Mcgraw-hill New York, 2005.

[16] James Scheer and William L Melvin. *Principles of modern radar, Vol.II: Advanced Techniques*, volume 2. SciTech Publishing, 2013.

[17] Richard G. Lyons. *Understanding Digital Signal Processing.* Addison Weasly Longman Inc., 1997.

[18] Suleyman Suleymanov. Design and Implementation of An FMCW Radar Signal Processing Module For Automotive Applications. *University Of Twente*, 2016.

[19] Sophocles J Orfanidis. *Introduction to signal processing, Second Edition.* Author, 2023.

[20] G. Larry Bretthorst. Nonuniform sampling: Bandwidth and aliasing. In *AIP conference proceedings*, volume 567, pages 1–28. American Institute of Physics, 2001.

[21] Petre Stoica, Randolph L Moses, et al. *Spectral analysis of signals.* Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[22] Harry M. Jol. *Ground Penetrating Radar Theory and Applications.* Elsevier, 2008.

[23] Hermann Rohling. Ordered statistic CFAR technique - an overview. In *2011 12th International Radar Symposium (IRS)*, pages 631–638, 2011.

[24] Anders Holst & Victor Ufnarovski. *Matrix Theory.* Studentliteratur AB, 2014.

[25] Mehdi Mobli & Jeffrey C. Hoch. Nonuniform sampling and non-Fourier signal processing methods in multidimensional NMR. *Progress in Nuclear Magnetic Resonance Spectroscopy*, pages 21–41.

[26] Tapan K Sarkar and Jinhwan Koh. Coherent processing across multiple staggered pulse repetition interval (PRI) dwells in radar. *Air Force Research Laboratory, Sensors Directorate, Rome, New York, Tech. Rep. AFRL-SN-RS-TR-*, pages 2004–43, 2004.

[27] Prabhu Babu and Petre Stoica. Spectral analysis of nonuniformly sampled data–a review. *Digital Signal Processing*, 20(2):359–378, 2010.

[28] M.Goppelt & H.-L Blöcher & W. Menzel. Automotive radar- investigation of mutual interference mechanisms. *Copernicus publications*, pages 1–7, 2010.

[29] M Gasior and JL Gonzalez. Improving FFT frequency measurement resolution by parabolic and Gaussian spectrum interpolation. In *AIP Conference Proceedings*, volume 732, pages 276–285. American Institute of Physics, 2004.

[30] Katamaneni SriDevi and D Elizbath Rani. Mainlobe width reduction using linear and nonlinear frequency modulation. In *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pages 918–920. IEEE, 2009.

[31] Christopher C Skiscim and Bruce L Golden. Optimization by simulated annealing: A preliminary computational study for the TSP. Technical report, Institute of Electrical and Electronics Engineers (IEEE), 1983.

[32] Franco Busetti. Simulated annealing overview. *World Wide Web URL www. geocities. com/francorbusetti/saweb. pdf*, 4, 2003.

[33] Daniel Delahaye, Supatcha Chaimatanan, and Marcel Mongeau. Simulated annealing: From basics to applications. *Handbook of metaheuristics*, pages 1–35, 2019.

[34] Walid Mahdi, Seyyid Ahmed Medjahed, and Mohammed Ouali. Performance analysis of simulated annealing cooling schedules in the context of dense image matching. *Computación y Sistemas*, 21(3):493–501, 2017.

[35] Xin-She Yang. *Nature-inspired optimization algorithms.* Academic Press, 2020.

[36] Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi, et al. Traveling salesman problem. *Encyclopedia of operations research and management science*, 1:1573–1578, 2013.

[37] F Rodríguez Henríquez, N Cruz Cortés, and JM Rocha-Pérez. Generation of gold-sequences with applications to spread spectrum systems. *IEEE communications letters*, 4(8), 2002.

[38] Ashish Patwari. Sparse linear antenna arrays: a review. *Antenna Systems*, 10, 2021.

[39] Wang Zheng, Xiaofei Zhang, Yunfei Wang, Mengjie Zhou, and Qihui Wu. Extended coprime array configuration generating large-scale antenna coarray in massive mimo system. *IEEE Transactions on Vehicular Technology*, 68(8):7841–7853, 2019.

[40] Gerardo Di Martino and Antonio Iodice. Passive beamforming with coprime arrays. *IET Radar, Sonar & Navigation*, 11(6):964–971, 2017.

[41] Alok Dutt and Vladimir Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM Journal on Scientific computing*, 14(6):1368–1393, 1993.

[42] William Roberts, Petre Stoica, Jian Li, Tarik Yardibi, and Firooz A Sadjadi. Iterative adaptive approaches to MIMO radar imaging. *IEEE Journal of Selected Topics in Signal Processing*, 4(1):5–20, 2010.

[43] Johan Karlsson, William Rowe, Luzhou Xu, George-Othon Glentis, and Jian Li. Fast missing-data IAA with application to notched spectrum SAR. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):959–971, 2014.

[44] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.