Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2018

High Resolution Frequency Estimation in an FMCW Radar Application

Johan Svensson

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

High Resolution Frequency Estimation in an FMCW Radar Application

Johan Svensson

LiTH-ISY-EX--18/5141--SE

Supervisor: **Kamiar Radnosrati**
ISY, Linköping University
Urban Blomberg
Rosemount Tank Radar AB

Examiner: **Fredrik Gustafsson**
ISY, Linköping University

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

FMCW radars are widely used in the process industry for range estimation, usually for estimating the liquid level in a tank. Since the tank system, often is an automatically controlled system, reliable estimates of the surface level are required, e.g. to avoid the tank from pouring over or become empty.

The goal of this thesis is to investigate methods which can distinguish frequencies closer to each other than the FFT resolution limit. Two properties are of interest, the accuracy and the resolution performance. Three such methods have been evaluated: one that tries to compensate for the leakage and interference of close frequencies, one subspace-based method and one deconvolution method. The deconvolution is performed with the iterative Lucy Richardson algorithm. The methods are evaluated against each other and against a typical FFT based algorithm.

The methods sensitivity to amplitude differences is examined together with the robustness against noise and disturbances which appear due to imperfections in the radar unit. The deconvolution algorithm is the one that performs the best. The subspace-based method SURE requires prior knowledge of the number of ingoing frequencies which is difficult to know for real data from an FMCW radar. The leakage compensation method main weakness is the influence of the phase difference between close frequencies.

The deconvolution algorithm is evaluated on some real data, and it is proven that it has better resolution performance than the FFT. However, the accuracy of the estimates are dependent on the number of iterations used. With a large number of iterations, the algorithm finds peaks with small amplitude nearby the large peaks and they will thus interact and hence contribute to a worse accuracy even in the undisturbed case. If too few iterations are used in the deconvolution algorithm the resolution performance is about the same as the FFT algorithm. With a suitable choice of iterations about 40–50 mm, extra of continuous measurements are achieved. However, the estimation error of the gained resolution can in the worst case be about 40–50 mm.

Acknowledgments

First of all, I would like to thank Rosemount Tank Radar AB for the opportunity to write this thesis. I am thankful for all the support, helpful comments and ideas during this time. My supervisor Urban Blomberg is worth some extra gratitude for always taking time to discuss any problem or ideas. Also, I would like to thank Lars-Ove Larsson for his ideas and comments during the thesis.

Secondly, I would like to thank the people involved in the thesis from the university, my supervisor Kamiar Radnosrati and examiner Fredrik Gustafsson for their helpful comments and time spent on the thesis.

Thirdly, I would like to thank all that I have come to know during the years at the University. An extra thanks to Fredrik Nilsson for being an excellent lab partner in all laboratory work during the years and for being my opponent for this thesis.

Last but not the least, I would like to thank my family and girlfriend Annie Karlsson for the support and encouragement during this 5-year journey

Linköping, May 2018
Johan Svensson

Contents

Notation

Mathematical notation

Notation	Meaning
$\bar{a}$	Complex conjugate of a
$\bar{\mathbf{A}}$	Complex conjugate of the elements in the matrix $\mathbf{A}$
$\mathbf{A}^T$	Transpose of the matrix $\mathbf{A}$
$\mathbf{A}^H$	Hermitian transpose, the same as $\bar{\mathbf{A}}^T$
$\hat{a}$	Estimate of the parameter a

Abbreviations

Abbreviation	Meaning
ACF	Auto Correlation Function
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
GNN	Global Nearest Neighbour
MSE	Mean Squared Error
PSF	Point Spread Function
RMSE	Root Mean Squared Error
SNR	Signal to Noise Ratio
SURE	SUbspace Rotation Estimation
VCO	Voltage Controlled Oscillator

1 Introduction

In this chapter, the background of the problem is explained together with the problem formulation and the delimitation of the thesis. In Section 1.1, the background of the problem is provided together with a few examples of applications in the literature. Then, the problem formulation and the questions/statements to be answered/investigated in this thesis are given in Section 1.2. The delimitations are then described in Section 1.3 followed by the method in Section 1.4. Finally, the outline of this work is described in Section 1.5.

1.1 Background

Frequency Modulated Continuous Wave (FMCW) radar is widely used in various applications such as the automotive industry and the process industry for range estimation. Two well-known examples are estimating the distance to the surrounding of a car or measuring the liquid level in a tank. When estimating the range, usually the Fast Fourier Transform (FFT) is applied due to its cheap computation and robustness to disturbances. However, as a major drawback, it can not distinguish frequencies when they are too close to each other. In the application of measuring the liquid level in a tank, this problem encounters for example when the surface is close to either a beam or the bottom. The bottom or the beam echo will then interact with the echo from the surface and consequently introduce an error in the estimated surface position. The problem also encounters when there is a thin layer of a different liquid on top of the original liquid whose level is going to be estimated. The most common case that is typical for Rosemount's costumers is oil on top of water. In case the thickness of the thin layer is below the FFT resolution limit this will typically introduce an error in the surface estimate.

Different methods have been developed to overcome the resolution limit of the FFT in the literature. One typical example is a subspace method called space alternating generalized maximization evaluated in [14]. These subspace methods work well when the model order of the signal is known i.e., the number of sinusoids in the signal. A different viewpoint of the problem is to see it as a multiple target tracking problem as in [4]. This approach works when the states (distance to the surroundings) change over time. For the automotive industry, the states change, either by the movement of the car and thus changing the distances to the surroundings or the movement of surrounding objects relative the car with the radar. However in an oil tank, the most common case is that the liquid level is constant, thus the tank is behaving as a static system. The tank can behave as a static system for long time periods, which is believed to be the main problem especially when the surface is close to a disturbance. The authors in [7] and [13] introduce algorithms based on the chirp Z-transform to increase the accuracy of the estimates. This is done by calculating the Chirp Z-transform of the FMCW signal to increase the bin size in the observed periodogram, thus increase the accuracy of the estimated surface level. However, Rosemounts problem in their algorithm is not the accuracy of the estimate of the surface level rather the resolution performance, which is not affected by using the Chirp Z–transform, thus not an algorithm to be considered.

1.2 Problem Formulation

The main purpose of this thesis is to perform a survey on different existing methods to deal with FFT resolution limit. Not only the resolution performance is of interest but also the accuracy of the estimated echoes. Additionally, it is of high importance to come up with methods that are robust against disturbances. Also, when the surface is nearby a disturbance with larger amplitude the resolution performance and accuracy is of interest. To make the problem statement more clear the following questions/statements are going to be investigated.

- Investigate the accuracy and resolution performance of methods that overcome the FFT resolution limit.
- Are the chosen methods sensitive to noise and other disturbances that exist in the radar unit?
- Are the resolution performance and the accuracy affected when the surface echo has smaller amplitude than the disturbing echo?

1.3 Delimitation

The main delimitation in this thesis is that only FMCW for range estimation in a tank will be considered. Additionally, we will not distinguish between the surface, beam, bottom, double bounces, etc. Instead, the focus is on separating two close echoes from each other. Also, the computational aspects will not be considered.

1.4 Method

The first step of the thesis is to obtain an understanding of the FMCW signal and its limitations when the FFT is applied. Then, a literature review on the state of the art of the problem is performed and three candidates are selected and compared with each other. One of the subspace-based methods, one method that tries to compensate for the leakage and interference of two close frequencies, one that originates from the idea of deconvolution. Also, a typical FFT based algorithm is described to see the limit that is desired to outperform. The algorithms are then tested in a simulation study that tries to mimic the challenges that affect the resolution and accuracy. The best algorithm is then evaluated further to more application specific setups in simulation and finally against real data.

1.5 Outline

In Chapter 2, the basics of the FMCW signal are introduced together with the disturbances that are present due to imperfections in the radar unit. This is followed by the theoretical background of the chosen methods in Chapter 3. The basics of a typical FFT based algorithm is also provided in the chapter. Chapter 4 provides a performance evaluation of the methods on both a simulation study and on some real data followed by a discussion of the pros and cons of the methods. Finally, Chapter 5 summarizes the thesis and concludes the work. The chapter ends with some suggestions for future work. Appendix A presents some extra results from the real data, followed by Appendix B where the Matlab code for the leakage compensation and the subspace-based method are presented. Last is Appendix C, where a flow chart for the decision structure for the leakage compensation is given.

2 FMCW radar

In this chapter, the fundamental parts of the FMCW signal are introduced. The ideal signal model is derived in Section 2.1. The radar unit is not perfect hence introduces disturbances that are discussed and modeled in Section 2.2.

2.1 Ideal signal model

One of the fundamental ways to measure the liquid level in a tank is to use an FMCW radar. In these systems, a radar unit transmits a frequency modulated signal generated from a Voltage Controlled Oscillator (VCO) with modulation frequency $f(t)$. In this application, the frequency modulation is linear and sweeps over a frequency band B, around the center frequency f_c.

In what follows the following definitions are used:

$$\begin{aligned} A_T &= \text{amplitude}, \\ \phi_T &= \text{phase [rad]}, \\ f_c &= \text{center frequency [Hz]}, \\ B &= \text{modulation bandwidth [Hz]}, \\ T &= \text{time for a sweep [s]}. \end{aligned}$$

With the definitions above the frequency of the transmitted signal can be written as

$$f(t) = f_c + \frac{B}{T}t, \qquad -\frac{T}{2} \leq t \leq \frac{T}{2}. \tag{2.1}$$

Then the angular argument of the transmitted signal is defined as the integral of $f(t)$ according to

$$\arg(t) = \int_0^t (f_c + \frac{B}{T}\tau) d\tau = f_c t + \frac{B}{2T} t^2. \tag{2.2}$$

With the earlier definitions and the angular argument introduced in equation (2.2) the transmitted signal can be written according to

$$v_T(t) = A_T \cos(2\pi(f_c + \frac{B}{2T}t)t + \phi_T). \tag{2.3}$$

Figure 2.1 presents a time plot of the transmitted signal and it can be seen that the frequency is increasing with time.

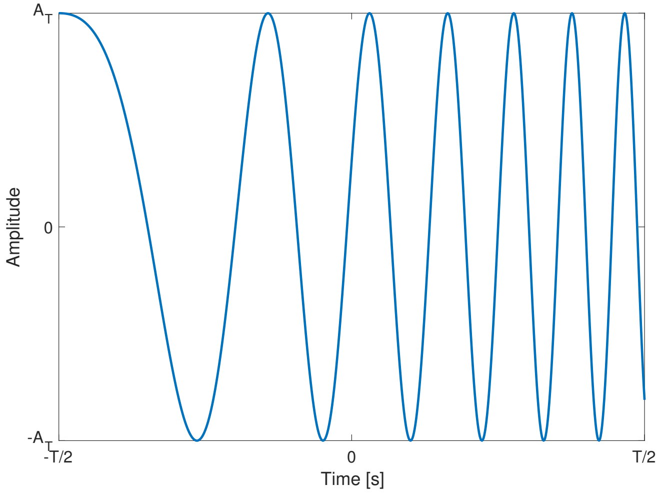


Figure 2.1: *Time plot of the transmitted signal.*

The transmitted radar signal will, after some time, hit a surface and some parts of the signal will be reflected and bounced back to the radar unit. How much of the signal that will be reflected depends on the reflection constant ρ, $\rho < 1$, of the material. If an object is at distance h [m] the received signal will be delayed by $\tau_d = \frac{2h}{c}$ [s] compared to the transmitted signal, where c $[\frac{\text{m}}{\text{s}}]$ is the speed of light. The received signal, $v_R(t)$, will then be a time-delayed sinusoid compared to the transmitted version with a frequency and phase shift and a change in amplitude according to ρA_T. Both the transmitted and received signal depend on the time t, but the time dependence can be removed if they are mixed together. Doing this, one receives what is called the beat signal or the tank signal. In Figure 2.2, a sketch is presented of how the beat signal is generated by mixing the transmitted and the received signal.

$$\tilde{v}_{if}(t) = v_T(t)v_R(t) = v_T(t)\rho v_T(t - \tau_d) \tag{2.4}$$

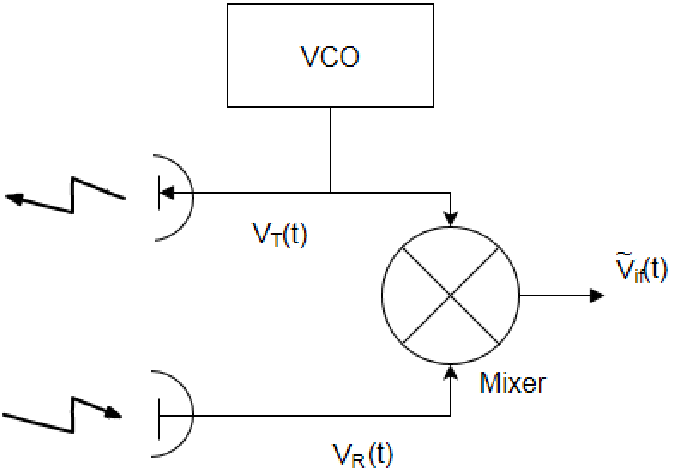


Figure 2.2: *Generation of the beat signal by mixing the transmitted and received signal.*

Inserting equation (2.3) and the same signal but delayed with τ_d and a change in amplitude in equation (2.4) one gets

$$\begin{aligned}\tilde{v}_{if}(t) = A_T^2\rho\cos\Big(2\pi(f_c + \frac{B}{2T}t)t + \phi_T\Big)&\\ \cos\Big(2\pi(f_c + \frac{B}{2T}(t-\tau_d))(t-\tau_d) + \phi_T\Big)&\\ = \frac{\rho A_T^2}{2}\Big(\cos\Big(2\pi(2f_c - \frac{B}{2T}(2\tau_d t + 3t))t) + 2\phi_T - f_c\tau_d\Big)&\\ + \cos\Big(2\pi(\frac{B}{T}\tau_d t + f_c\tau_d - \frac{B}{2T}\tau_d^2)\Big)\Big),& \quad (2.5)\end{aligned}$$

where $\cos(\alpha)\cos(\beta) = \frac{1}{2}(\cos(\alpha+\beta) + \cos(\alpha-\beta))$ is used in the last step. The first term consists of twice the center frequency f_c which consequently will be extremely high frequent since f_c is of size 10–25 GHz and will thus not be able to propagate in the system. From now on, when talking about the beat signal or tank signal, only the last term is considered which can be written as

$$v_{if}(t) = \frac{\rho A_T^2}{2}\cos\Big(2\pi(\frac{B}{T}\tau_d t + f_c\tau_d - \frac{B}{2T}\tau_d^2)\Big) = A\cos\Big(2\pi f_{if}(h)t + \phi_{if}(h)\Big), \quad (2.6)$$

where

$$f_{if}(h) = \frac{B}{T}\tau_d = \frac{2B}{cT}h, \quad (2.7)$$

$$\phi_{if}(h) = (f_c - \frac{B}{2T}\tau_d)\tau_d = \frac{4\pi}{c}(f_c - \frac{B}{Tc}h)h, \tag{2.8}$$

and

$$A = \frac{\rho A_T^2}{2}. \tag{2.9}$$

Now a signal with frequencies only dependent on τ_d is obtained and as mentioned before $\tau_d = \frac{2h}{c}$, thus if it is possible to estimate the frequency f_{if} of the beat signal the distance h can easily be computed. The problem of estimating the distance h has come to a frequency estimation problem. In practice, the received signal does not only consist of the echo from the surface, but also from disturbances such as beams, reflections from the bottom, double bounces etc. The signal received is then defined according to

$$v_{if}(t) = \sum_{k=1}^{p} A_k \cos(2\pi f_k t + \phi_k), \tag{2.10}$$

where p is the number of echoes received. To receive an understanding of how the frequency and phase change in the beat signal with changed h typical values for a radar unit will be used. The values can be examined in Table 2.1.

Table 2.1: *Typical values for the parameters in a radar unit.*

f_c	25 GHz
B	3 GHz
F_s	125 kHz
N	1000
c	$2.99792458 \cdot 10^8$ m/s

With the values in Table 2.1, equation (2.7) will lead to

$$f_{if}(h) = \frac{2B}{cT}h \approx 2.5017 \cdot h \quad \text{kHz}, \tag{2.11}$$

which means that the frequency changes about 2.5 kHz when the surface has moved 1 meter. The expression for the phase in equation (2.8) can be simplified since the second term will be much smaller than the first term, $f_c = 25 \cdot 10^9 >> \frac{B}{Tc}h \approx 1250 \cdot h$ and h is limited to approximately 25 meters by the sampling frequency and the number of samples. With this simplification, the phase can be written as

$$\phi_{if} = \frac{4\pi f_c}{c}h \approx 1.048 \cdot 10^3 \cdot h. \tag{2.12}$$

The phase changes in the order of 1048 radians per meter or expressed differently, h changes 6 mm when the phase goes from 0–2π. From this one can realize that if it would be possible to estimate the absolute phase of the beat signal an accuracy of less than 6 mm could be accomplished.

2.2 Disturbance model

Section 2.1 introduces the ideal signal model, but in practice the received signal is not an ideal sinusoid. The disturbances that are modeled in this section are the ones that occur due to imperfections in the microelectronics. Other disturbances are present as well but they are in general more dependent on the liquid in the tank and the tank itself, double bounces etc. However, they are beyond the scope of this thesis, hence neglected.

2.2.1 Amplitude modulation

The transmitted signal will differ in amplitude due to imperfections in the VCO. The VCO will not be able to transmit a signal with the same amplitude A_T all the time. An easy way to model this would be according to

$$A_T(t) = A_T(1 + a_m \cos(\omega_m t + \phi_m)), \tag{2.13}$$

where a_m is the amplitude of the imperfection, ω_m angular frequency of the amplitude variations and ϕ_m the phase. Nominal values for the constants are presented in Table 2.2. The values are obtained from Rosemount's electronic department, by computer simulations and empirical investigations of the transmitted signal.

Table 2.2: *Nominal values for the amplitude modulation imperfection.*

a_m	0.2
ω_m	$\frac{\pi}{T}$
ϕ_m	In range $[0\ 2\pi]$

To develop an understanding of how the amplitude modulation affects the periodogram, the periodogram from an ideal sinusoid is plotted together with the same sinusoid but with amplitude modulation added with the nominal values in Table 2.2. As Figure 2.3 suggests the effect is not that large but it is visible that two extra peaks, one on each side, will show up.

2.2.2 Multipath

The true echo at distance h_0 is often measured with a few ghost echoes which are originated from multiple propagation paths in the microwave guide and antenna. This results in echoes with larger frequency and smaller amplitudes due to energy loss when the signal bounces in the microwave guide. A simple model of the multipath disturbance is defined by

$$20\log_{10}(\frac{A_{k+1}}{A_k}) = A_{att}, \tag{2.14}$$

$$h_{k+1} - h_k = \Delta h. \tag{2.15}$$

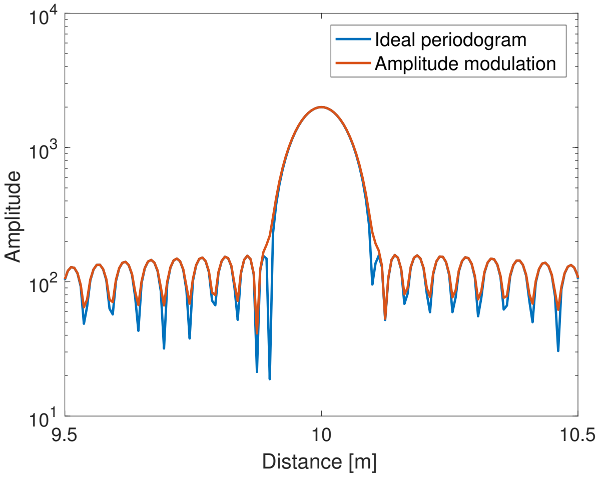


Figure 2.3: *The ideal periodogram and the periodogram with amplitude modulation added.*

The nominal values are obtained from Rosemount by investigating the result from the FFT of an undisturbed case i.e, letting the radar signal reflect against one single metal plate. The nominal values for the multipath imperfection are presented in Table 2.3.

Table 2.3: *Nominal values for the multipath imperfection.*

A_{att}	-20 dB
Δh	0.14 m

How the multipath changes the periodogram compared to the ideal case can be observed in Figure 2.4. What can be seen is that extra smaller main lobes appear at a further distance than the main echo.

2.2.3 Frequency dispersion

The propagation velocity in a tube is frequency dependent. An easy way of modeling the microwave tube is as a small pipe with length L [m] and diameter d [m]. The phase velocity is then given by

$$v_{phase}(f) = \frac{c}{\sqrt{1 - \left(\frac{cP_{xy}}{\pi d f}\right)^2}}. \tag{2.16}$$

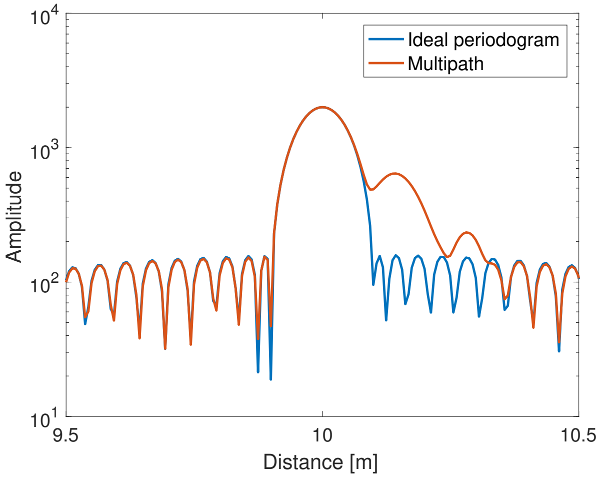


Figure 2.4: *The ideal periodogram and the periodogram with the multiple path imperfection.*

This will then lead to a time delay according to

$$\tau_{disp}(f) = \frac{2L}{c}\sqrt{1-\left(\frac{cP_{xy}}{\pi d f}\right)^2}. \tag{2.17}$$

Nominal values for the frequency dispersion are presented in Table 2.4. The model and the nominal values are provided from Rosemount's microwave department and originate from computer simulations.

Table 2.4: *Nominal values for the frequency dispersion.*

P_{xy}	Parameter depending on propagation mode	1.8412
d	Diameter of the tube	0.008 m
f	Sweep frequency	$[f_c - \frac{B}{2}, f_c + \frac{B}{2}]$ [Hz]
L	Gauge inner length	0.1 [m]

The main effect that the dispersion will have on the periodogram is to shift it to a further distance than the true echo. Since these shifts are deterministic they can be compensated for. Doing this one receives what can be seen in Figure 2.5. The main difference compared to the periodogram from an ideal sinusoid is that the dispersion makes the main lobe somewhat wider.

2.2.4 Complete signal model

With the disturbances introduced in Section 2.2.1–2.2.3 and the ideal signal model introduced in Section 2.1, every reflecting echo will consist of the following sig-

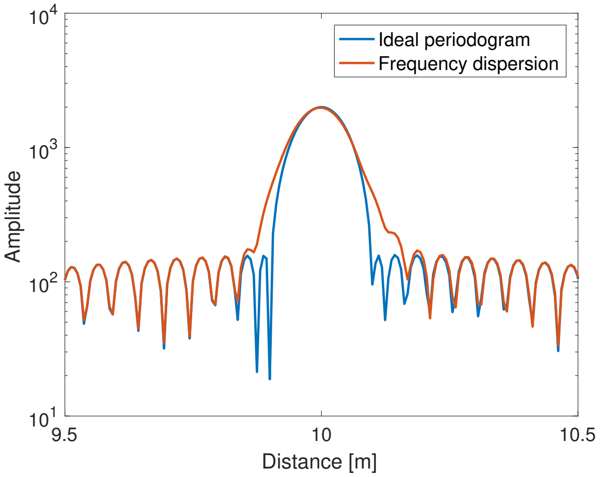


Figure 2.5: *The ideal periodogram and the periodogram with frequency dispersion added.*

nal

$$v_{if}(t) = \sum_{k=0}^{2} \Bigg(A_T(t)A_k \\ \cos\left(2\pi\left(\frac{B(\tau_{d,k}+\tau_{disp})}{T}t + \tau_{d,k} + \tau_{disp} - \frac{B(\tau_{d,k}+\tau_{disp})^2}{2T}\right)\right)\Bigg) + u(t), \quad (2.18)$$

where $u(t)$ is the noise which is assumed to be $u(t) \sim N(0, \sigma^2)$. The noise is assumed to have a Signal to Noise Ratio (SNR) of approximately 30 dB. The multipath is modeled with the main echo and two more peaks, the rest will disappear in the side lobes of the window. All the disturbances added together can be seen in Figure 2.6 compared with the ideal FFT periodogram. As in the frequency dispersion case, the periodogram is shifted back to compensate for the time shift.

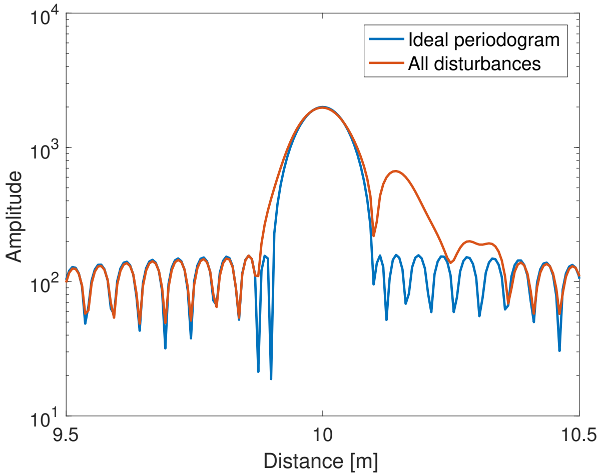


Figure 2.6: *All the disturbances added and compared with the ideal periodogram.*

3 Theory

This chapter walks through the theory of the chosen methods that are being compared. Section 3.1 describes a typical FFT based algorithm. This is the method that we want to outperform. Section 3.2 describes a leakage compensation method which tries to compensate for the interaction between two close frequencies. Section 3.3 describes the idea and theory for the deconvolution algorithm. Section 3.4 explains the theory for the subspace method SURE.

3.1 FFT algorithm

A typical FFT algorithm for this kind of application mainly consists of three steps. The first is to multiply the discrete signal with a window function to suppress the side lobes in the frequency domain. The second step is to zero-pad the signal and calculate the FFT to expose the frequency components in the signal. The last step is to extract the peaks in the periodogram and determine what distance the frequency peaks corresponds to.

3.1.1 Window

The use of a window is important to suppress the side lobes which occur from the leakage from the main lobe. The FFT algorithm studied, uses a Taylor window which provides the user the opportunity to make a trade off between the main lobe width and the side lobe suppression [9]. This opportunity has made the Taylor window popular within the area of radar signal processing [6]. The trade off between side lobe level and main lobe level is desired to adapt the radar unit to work optimally for each application.

3.1.2 FFT

For exposing the frequency content of the received time signal in equation (2.18) the FFT is applied. It is used because of its low computational cost and its robustness against low SNR. Its main disadvantage is when two frequencies are close to each other, then only one of them will be detected. The resolution limit is given by $f_{min} = \frac{F_s}{N}$, where F_s is the sampling frequency and N is the number of samples from the continues signal. Therefore the most natural approach when high frequency resolution is required, is to increase the number of samples N. This provides the higher resolution in most of the cases, but not when working with an FMCW radar signal. This can be proven by rewriting the beat frequency $f_{if}(h)$ from Chapter 2 according to

$$f_{if}(h) = \frac{2B}{cT}h \implies h = \frac{cT}{2B}f_{if}. \tag{3.1}$$

Then the FFT resolution limit can be rewritten as

$$f_{min} = \frac{F_s}{N} = \frac{1}{NT_s} = \frac{1}{T}, \tag{3.2}$$

where T_s is the sampling time and T is the total sample time. By inserting equation (3.2) in equation (3.1) one gets

$$h_{min} = \frac{cT}{2B}f_{min} = \frac{cT}{2B}\frac{1}{T} = \frac{c}{2B}. \tag{3.3}$$

Equation (3.3) proves that the standard way of providing increased resolution does not work. The only thing that can be done is to increase the bandwidth B of the transmitted signal which is not an option because of legal requirements and hardware related problems. The limit h_{min} will in reality also be degraded since the beat signal is multiplied with a window. The Taylor window with its default parameters doubles the limit and equation (3.3) then becomes

$$h_{min} = \frac{c}{B}. \tag{3.4}$$

The radar used in this thesis if nothing else is stated, has a bandwidth of approximately 3 GHz which corresponds to a resolution limit of 100 mm.

3.1.3 Interpolation

When calculating the FFT it produces values at the FFT frequencies which are equally spaced with $\frac{F_s}{N}$ Hz. This introduces an error when using peak picking if the true frequency is not located at an exact multiple of $\frac{F_s}{N}$. Instead the true frequency is in the range

$$\left[n\frac{F_s}{N} - \frac{F_s}{2N}, n\frac{F_s}{N} + \frac{F_s}{2N},\right], \qquad 0 \leq n < \frac{N}{2}. \tag{3.5}$$

To reduce this error different methods have been suggested over the years. The idea behind them is best described by examining Figure 3.1. The figure presents the values from an FFT operation where the maximum peak is located around 2.05 m. By examining the two closest frequencies one can see that the frequency to the right has a larger amplitude than the frequency to the left. It is then reasonable to assume that the true maximum is located somewhere in between the maximum frequency and the frequency to the right. This is what is called interpolation, combining the discrete frequencies close to the maximum and thus receive a better estimate of where the true maximum is located.

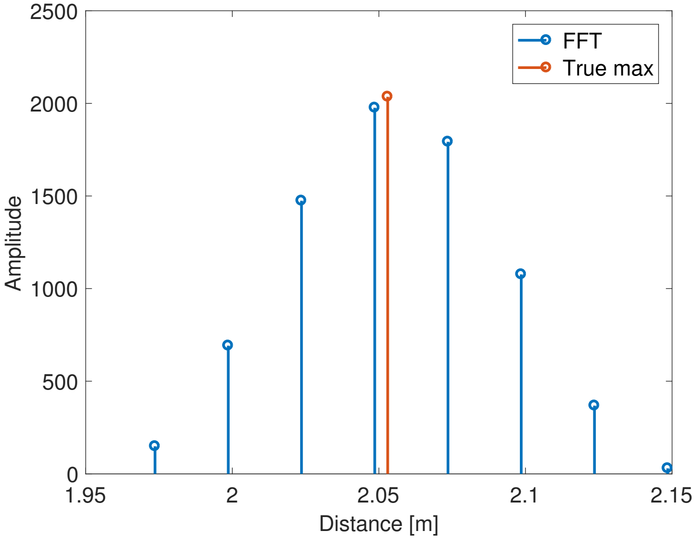


Figure 3.1: *The idea of interpolation explained. The FFT provides the blue lines and the maximum frequency is located at an offset from the true maximum, the orange line.*

3.2 Leakage compensation

The standard FFT based method works well when the FMCW signal only consists of one or a few sinusoids which are clearly separated in the frequency domain. The problem arises when the signal consists of two close frequencies, which occurs from the surface approaching a beam or the bottom. Then the disturbing echo will interact with the echo from the surface and depending on how close they are both will be detected or only one of them. To handle this problem a new improved algorithm is proposed below, which consists of a second step after the FFT based algorithm. The algorithm is best motivated by investigating Figure 3.2. The FFT from a single sinusoid is plotted together with the FFT of two sinusoids with close frequencies. For the double peak only one local maximum is found but visually one can see that more than one frequency is present. The main idea for the improved algorithm is to assume that each peak which is observed consists of two peaks instead of one.

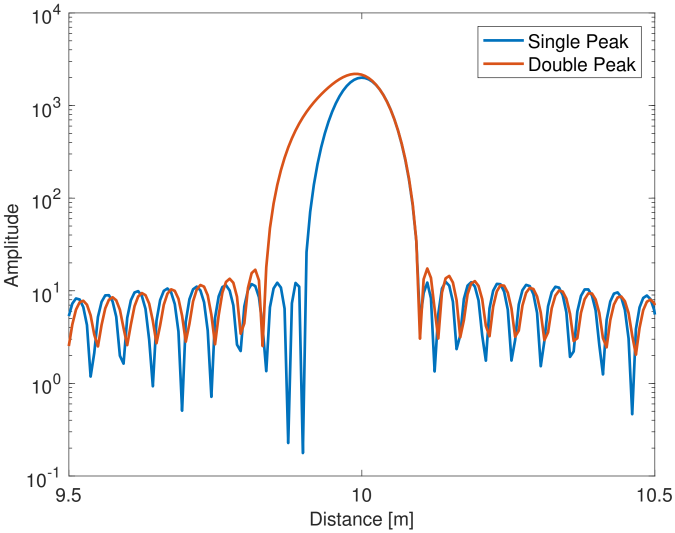


Figure 3.2: *Plot of the FFT from one single sinusoid together with two sinusoids with close frequencies. The double peak only has one local maximum but it is clearly seen that it consists of more than one frequency.*

3.2.1 Fitting of windows

To distinguish frequencies close to each other it is assumed that each peak in the observed frequency domain consists of two peaks. Since the FMCW signal is multiplied with a Taylor window before the FFT is applied, the window fitting contains of fitting two Taylor windows to the true periodogram. To receive the fit, the two Taylor windows are allowed to move around and be compared to the true periodogram.

3.2.2 Amplitude relation

Since the amplitude relation between the two assumed Taylor windows is unknown, it needs to be estimated. One way of doing this is to assume that the sum of the energy in the two windows, P_{W1} and P_{W2} is equal to the energy in the true periodogram P_Y. This provides the relation

$$P_Y = P_{W1} + P_{W2}. \tag{3.6}$$

Then it is assumed that the second window has amplitude A which is smaller than the amplitude of the first window. With this assumption equation (3.6) can be written according to

$$P_Y = P_{W1}(1 + A^2) \Longrightarrow A = \sqrt{\frac{P_Y}{P_{W1}} - 1}. \tag{3.7}$$

This simplification will introduce a few errors. The closer the frequencies are in the observed frequency domain the less accurate the estimation will be, because then will the two windows interact more with each other.

3.2.3 Minimization

To receive a numerical value of the fit, a minimization criterion is needed. One of the most common ways is to calculate the Mean Squared Error (MSE) of the fit according to

$$\epsilon = \frac{1}{N}\sum_{i=0}^{N}(|y_i - \hat{y}_i|^2), \tag{3.8}$$

where y_i denotes one sample of the true periodogram and $\hat{y}_i$ is one sample from the estimated periodogram. The number of samples,N, is chosen large enough such that the entire peak plus some extra in the surrounding are within the minimization. If N is chosen too large, peaks from other echos will affect the minimization.

When the minimum is found one has to determine if the minimum corresponds to moving an already existing peak, insert an extra peak or change nothing. To provide an understanding of how the decision structure works a flow chart is presented in Appendix C. The values used in the decision structure is developed with trial and error to receive the best possible result.

One of the main concerns in the window fitting is how the phase of the FMCW signal will affect the observed periodogram. This is easiest examined by taking two sinusoids with close frequencies (about 70 mm apart) and then change the phase of one of them and plot the corresponding periodogram. Repeating the procedure, one will end up with a result according to Figure 3.3. As there can be seen the phase affects the result in a way that might be a concern since the minimization is performed by comparing the sum of two Taylor windows with the true periodogram provided from the FFT. The question is basically if it will give a totally different minimum or just a slight offset in the minimization.

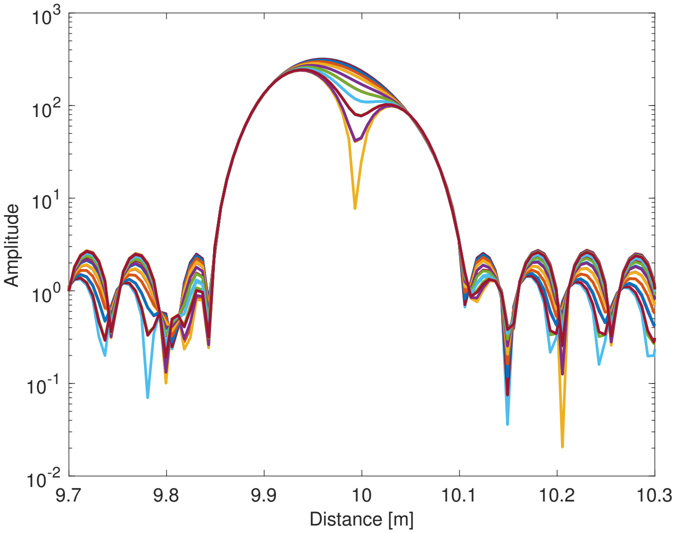


Figure 3.3: *The FFT of two sinusoids with fix frequency but the phase is changed in one of them. The phase is changed in steps and the results are plotted.*

3.3 Deconvoluition

The second method that will be evaluated is deconvolution. Its basic idea is that there is a recorded signal $y(t)$ which consists of a convolution between a signal $x(t)$, which one wants to recover, and a signal $g(t)$ which often is called the Point Spread Function (PSF). This is expressed in more mathematical terms by

$$y(t) = (x * g)(t), \tag{3.9}$$

where $*$ denotes the convolution. To extract the signal $x(t)$ from equation (3.9) one has to estimate a filter $h(t)$ such that

$$\hat{x}(t) = (y * h)(t) = (x * g * h)(t) \approx x(t). \tag{3.10}$$

Finding the filter $h(t)$ in the time domain is in general hard. What one could do then is to solve the problem in the frequency domain instead. The problem is then easier to solve since a convolution between two signals in the time domain corresponds to multiplication in the frequency domain. Expressing equation (3.9) in the frequency domain one gets

$$Y(f) = X(f)G(f) \Longrightarrow X(f) = \frac{Y(f)}{G(f)}, \tag{3.11}$$

where $Y(f)$, $X(f)$, $G(f)$ corresponds to the Fourier transform of the corresponding time domain signals. The signal $x(t)$ can then be received by doing an inverse

Fourier transform on the signal $X(f)$. In practice, the signal $y(t)$ not only consists of the convolved signal $(x * g)(t)$, but also of noise which complicates matters.

The above reasoning also applies if the signal is convoluted in the frequency domain since a convolution in the frequency domain corresponds to multiplying the signals in the time domain.

3.3.1 Deconvolution in FFT periodogram

The leakage observed in the FFT periodogram originates from the truncation of the Discrete Time Fourier Transform [8] and its main problem is the separation of close frequencies. If the time signal is not truncated the result from a pure sinusoid would be a Dirac impulse in the frequency domain. With this insight, one can see the FFT periodogram as a convolution between the true periodogram $X(f)$ and the window function $W(f)$. If this effect could be reversed with the idea of deconvolution the true periodogram, $X(f)$ could be reconstructed. In practice, a few problems occur, first the periodogram is frequency sampled and therefore will Dirac impulses be impossible to reconstruct. The second problem is that a physical signal commonly consists of noise which makes it more complicated to reconstruct $X(f)$.

The first problem is hard to come around, in discrete time the periodogram will always be frequency sampled. What one could do is to zero-pad the signal before the FFT is calculated to provide a denser frequency grid and therefore a more narrow peak when the deconvolution is applied. The second problem is the noise and many different methods have been developed in the literature over the years. The method which will be evaluated is the Lucy Richardson deconvolution which is an iterative method. In Section 3.3.2, the algorithm will be described in detail. Other methods that exist are the inverse filter which is calculated in equation (3.11) which assumes zero noise or the Wiener filter approach. The Wiener filter deconvolution provides optimal reconstruction in the MSE sense under the assumption that the noise is Gaussian. Since the noise is assumed to be Gaussian in the thesis the Wiener filter deconvolution should be the natural choice of method. However, it is proven by simulations that it does not work in this application. The result from a noise free case can be examined in Figure 3.4, where it clearly can be seen that the side lobes are amplified. This is for an undisturbed case, with only one echo. With two echoes close to each other the result would look even worse with many peaks with the same amplitude as the main echoes. This could be compared with the result from Lucy Richardson deconvolution in Figure 3.5, where two distinct peaks are present and the side lobes from the leakage are removed. Thus, the Lucy Richardson deconvolution is a better option than the Wiener deconvolution in this application even though Lucy Richardson is derived from a different noise realization.

Some further notes about the noise can be done. The noise is assumed to be additive and Gaussian in the time signal. When calculating the FFT spectrum the noise will become complex and Gaussian. However, since it is assumed that it is the periodograms $W(f)$ and $X(f)$ which is convolved, the noise will become exponentially distributed in the FFT periodogram $Y(f)$. This further motivates that the Wiener filter deconvolution is not necessarily the natural choice of method.

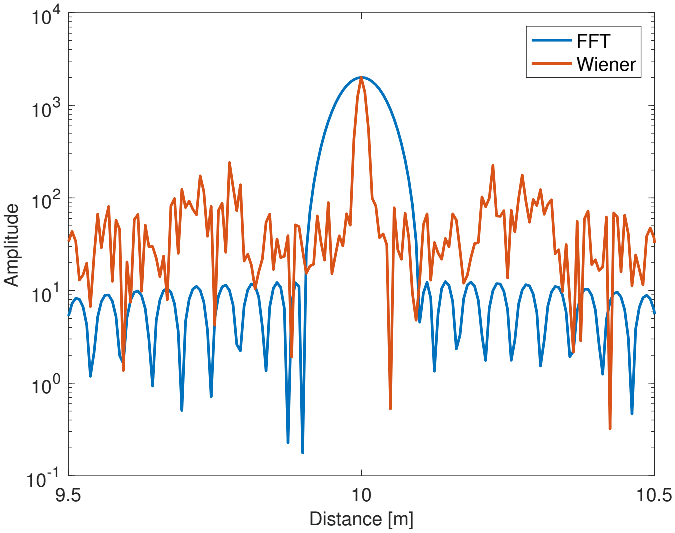


Figure 3.4: *The result from the Wiener deconvolution with zero noise plotted together with the periodogram from the FFT as a reference.*

3.3.2 Lucy Richardson deconvolution

The Lucy Richardson algorithm was developed in the 1970's by William Richardson [12] and Leon Lucy [11] from the Bayesian theorem. It was developed to receive better result than Fourier-Transform methods within the area of image restoration. The method has been proven to get better result compared to the Fourier based standard methods especially when the noise levels are high.

The main assumption for the algorithm is that the true periodogram $X(f)$ is convolved with the window function $W(f)$ and the noise which is added originates from a Poisson distribution and thus forms the FFT periodogram $Y(f)$. The Bayesian theorem in this context says with the same notation as in Section 3.3.1

$$p(X(f)|Y(f)) = p(Y(f)|X(f))\frac{p(X(f))}{p(Y(f))}, \tag{3.12}$$

where $p(X(f)|Y(f))$ is the a posteriori probability, $p(Y(f)|X(f))$ the likelihood probability, $p(X(f))$ a prior model of the true periodogram and $p(Y(f))$ the model provided by the FFT. If the posteriori probability is maximized with respect to $X(f)$ the best estimate of $X(f)$ is received. Maximizing $p(X(f)|Y(f))$ would be the same as maximizing the likelihood probability with respect to $X(f)$ because $p(Y(f))$ and $p(X(f))$ are known. Since the noise is assumed to be Poisson distributed in the algorithm the likelihood probability can be expressed as

$$p(Y(f)|\hat{X}(f)) = \prod_{f=0}^{n\frac{F_s}{2}} \frac{[(W * \hat{X})(f)]^{Y(f)} e^{-(W*\hat{X})(f)}}{Y(f)!}, \qquad 0 < n \leq \frac{N}{2}, \tag{3.13}$$

where $\hat{X}(f)$ is an estimate of the true periodogram and N the number of samples from the continuous signal $y(t)$. $Y(f)!$ should be interpreted as the gamma function, which is the generalization to all real number of the usual factorial function. Maximizing equation (3.13) with respect to $\hat{X}(f)$ is the same as minimizing

$$\hat{X}_{ML} = \arg\min_{\hat{X}(f)} (\sum_{f=0}^{n\frac{F_s}{2}} ((W * \hat{X})(f) - Y(f) \ln W(f) * \hat{X}(f) + \ln Y(f)!)), \tag{3.14}$$

which can be simplified, because the last term is not dependent of $\hat{X}(f)$, to

$$\hat{X}_{ML} = \arg\min_{\hat{X}(f)} \sum_{f=0}^{n\frac{F_s}{2}} ((\hat{X} * W)(f) - Y \ln (\hat{X} * W)(f)). \tag{3.15}$$

Calculating the gradient of equation (3.15) and solving it equal to zero one will end up with

$$W(-f) * \frac{Y(f)}{(W * X)(f)} = \bar{1}, \tag{3.16}$$

where $\bar{1}$ is a vector with ones of suitable size. For a more detailed derivation of the gradient see for example [5]. Then it is assumed that the algorithm in convergence fulfills

$$\frac{X_{k+1}(f)}{X_k(f)} = \bar{1}, \tag{3.17}$$

and inserting that into equation (3.16) one will receive the famous Lucy Richardson algorithm with k denoting the iteration index

$$\hat{X}_{k+1}(f) = \hat{X}_k(f)\Big(W(-f) * \frac{Y(f)}{(W * \hat{X}_k)(f)}\Big). \tag{3.18}$$

Equation (3.18) in its simplest form may lack of slow convergence and therefore much time is spent in later years on speeding up the algorithm to converge faster. The version which is used in MATLAB is based on an accelerated Lucy-Richardson algorithm [3] which guarantees faster convergence. The main difference is how the step size k is chosen.

Theoretically, the more iterations used the closer will the estimated periodogram come to the true one. However, this is not always the case because the signal always includes noise and individual noise spikes in the FFT periodogram might be amplified when the number of iterations is increased [3]. This is however in our application, not a problem as long as the amplified noise spikes are not in the area where the surface echo is located.

A result of the deconvolution can be seen by investigating the periodogram with the double peak in Figure 3.2 and see how the Lucy Richardson algorithm performs. The result can be seen in Figure 3.5. Ideally, there would only be two Dirac impulses left, that is not the case. However, the two peaks are now well resolved and possible to distinguish with peak picking. The plot might look somewhat strange since the periodogram from the deconvolution is not plotted for all values. That is because the y-axis is in logarithmic scale and all other values in the deconvoluted periodogram are equal to zero.

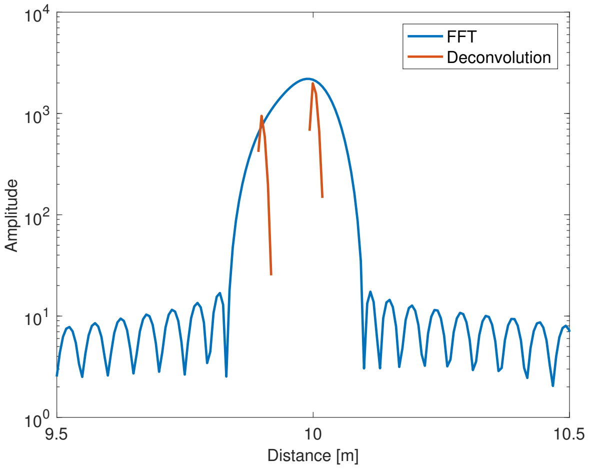


Figure 3.5: *The FFT periodogram of two close frequencies with only one maximum together with the deconvolved periodogram which separates the two frequencies.*

3.4 SURE

The methods described in Section 3.1–3.3 are in some sense evaluated in the frequency domain. There also exist methods that are evaluated in the time domain. A similar work [2] which was done some time ago focused on only time domain methods. One of the algorithms that performed the best in that work is chosen to be evaluated as well. The time domain methods were developed to overcome the FFT resolution limit and this is possible by using a priori information of the signal, in this case, the number of sinusoids in the signal. The method chosen is called SURE (SUbspace Rotation Estimation) and uses the regular structure in the covariance matrix for the data. The derivation follows the same structure as in [2] and [15].

3.4.1 Data model

The assumption for the derivation is that there is a signal consisting of p complex sinusoids with frequencies $f_1, \ldots, f_p$ with real amplitudes $A_1, \ldots, A_p$. Then can the data be written as

$$x[n] = \sum_{k=1}^{p} A_k e^{j(2\pi f_k + \phi_k)}, \quad n = 0, 1, ..., N-1, \tag{3.19}$$

where N is the number of samples and ϕ_k the phases. The additive noise $w[n]$ is assumed to be complex white Gaussian noise with mean 0 and variance σ^2. The received data can then be written according to

$$y[n] = x[n] + w[n], \quad n = 0, 1, ..., N-1. \tag{3.20}$$

To simplify matters later the following definitions are done,

$$\mathbf{\Phi} = \begin{bmatrix} e^{j2\pi f_1} & & \cdots & 0 \\ & e^{j2\pi f_2} & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & e^{j2\pi f_p} \end{bmatrix}, \tag{3.21}$$

$$\begin{aligned} \mathbf{a} &= \{a_1 \quad \cdots \quad a_p\}^T, \\ a_k &= A_k e^{j\phi_k}, \\ \mathbf{c} &= \{1 \quad 1 \cdots \quad 1\}_{1\times p}. \end{aligned} \tag{3.22}$$

With the definitions above equation (3.20) can be rewritten according to

$$y[n] = \mathbf{c}\mathbf{\Phi}^n\mathbf{a} + w[n], \quad n = 1, \cdots, N. \tag{3.23}$$

3.4.2 Auto correlation matrix

The auto correlation matrix R_{yy} with dimension $M \times M$ contains the M first elements in the Auto Correlation Function (ACF) $r_{yy}[m]$. Thus can R_{yy} for the data be defined by

$$R_{yy} = \begin{bmatrix} r_{yy}[0] & r_{yy}[-1] & \cdots & r_{yy}[-(M-1)] \\ r_{yy}[1] & r_{yy}[0] & \cdots & r_{yy}[-(M-2)] \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}[M-1] & r_{yy}[M-2] & \cdots & r_{yy}[0] \end{bmatrix}. \tag{3.24}$$

The ACF is calculated by

$$\begin{aligned} r_{yy}[m] &= \mathbb{E}[(x[n]+w[n])(\overline{x}[n+m]+\overline{w}[n+m])] \\ &= \mathbb{E}[x[n]\overline{x}[n+m]] + \mathbb{E}[w[n]\overline{w}[n+m]] \\ &= \sum_{k=1}^{p} A_k^2 e^{j2\pi f_k m} + \sigma^2\delta_m, \end{aligned} \tag{3.25}$$

where $\mathbb{E}[\,\cdot\,]$ denotes the expected value and δ_m is defined by

$$\delta_m = \begin{cases} 1, & m = 0, \\ 0, & m \neq 0. \end{cases} \tag{3.26}$$

Then inserting (3.25) into (3.24) one will get

$$R_{yy} = \begin{bmatrix} \sum_{k=1}^{p} A_k^2 & \sum_{k=1}^{p} A_k^2 e^{-j2\pi f_k} & \cdots & \sum_{k=1}^{p} A_k^2 e^{-j2\pi f_k (M-1)} \\ \sum_{k=1}^{p} A_k^2 e^{j2\pi f_k} & \sum_{k=1}^{p} A_k^2 & \cdots & \sum_{k=1}^{p} A_k^2 e^{-j2\pi f_k (M-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{p} A_k^2 e^{j2\pi f_k (M-1)} & \sum_{k=1}^{p} A_k^2 e^{j2\pi f_k (M-2)} & \cdots & \sum_{k=1}^{p} A_k^2 \end{bmatrix} + \sigma^2 \mathbf{I}_M. \tag{3.27}$$

Further one can rewrite equation (3.27) by introducing the matrices

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j2\pi f_1} & e^{j2\pi f_2} & \cdots & e^{j2\pi f_p} \\ e^{j2\pi 2 f_1} & e^{j2\pi 2 f_2} & \cdots & e^{j2\pi 2 f_p} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j2\pi (M-1) f_1} & e^{j2\pi (M-1) f_2} & \cdots & e^{j2\pi (M-1) f_p} \end{bmatrix}, \tag{3.28}$$

and

$$\mathbf{P} = \begin{bmatrix} A_1^2 & & \cdots & 0 \\ & A_2^2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & A_p^2 \end{bmatrix}. \tag{3.29}$$

Then equation (3.27) can be written as

$$\mathbf{R_{yy}} = \mathbf{EPE}^H + \sigma^2 \mathbf{I}, \tag{3.30}$$

and from this one can realize that the auto correlation matrix contains all information to estimate the frequencies of the sinusoids. Now only the estimation of the auto correlation matrix remains. This can be done by creating the matrix

$$\mathbf{Y} = [y[0] \quad y[1] \quad y[2] \quad \cdots \quad y[N-M]]$$
$$= \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_{N-M} \\ y_1 & y_2 & y_3 & & y_{N-M+1} \\ \vdots & & & \ddots & \vdots \\ y_{M-1} & y_M & y_{M+1} & \cdots & y_{N-1} \end{bmatrix}, \tag{3.31}$$

which can be created directly from the data. Then an estimate of the auto correlation matrix can be generated by

$$\hat{\mathbf{R}}_{yy} = \mathbf{Y}\mathbf{Y}^H. \tag{3.32}$$

3.4.3 Subspace

Denoting the eigenvalues to $\mathbf{R}_{yy}$ by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$, then equation (3.30) yields,

$$\lambda_k > \sigma^2 \quad \text{for} \quad k = 1, \cdots, p, \tag{3.33}$$

$$\lambda_{p+1} = \cdots = \lambda_M = \sigma^2. \tag{3.34}$$

Then by denoting the eigenvectors associated with $\{\lambda_1, \cdots, \lambda_p\}$, $\{s_1, \cdots, s_p\}$ and in the same manner $\{g_1, \cdots, g_{M-p}\}$ denoting the eigenvectors corresponding to $\{\lambda_{p+1}, \cdots, \lambda_M\}$. Then what is called the signal subspace $\mathbf{S}$ and the noise subspace $\mathbf{G}$ can be defined as

$$\mathbf{S} = [s_1 \cdots s_p], \tag{3.35}$$
$$\mathbf{G} = [g_1 \cdots g_{M-p}].$$

Further the matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ are defined as

$$\mathbf{S}_1 = [I_{M-1} \quad 0]\mathbf{S}, \tag{3.36}$$

$$\mathbf{S}_2 = [0 \quad I_{M-1}]\mathbf{S}, \tag{3.37}$$

where I_{M-1} denotes the identity matrix with size $M - 1 \times M - 1$. The last step would then be to calculate a matrix $\mathbf{\Psi}$ such that

$$\mathbf{S}_2 = \mathbf{S}_1\mathbf{\Psi}, \tag{3.38}$$

is fulfilled. The matrix $\mathbf{\Psi}$ eigenvalues will be $e^{i2\pi f_k}$ for $k = 1, \cdots p$. A more detailed explanation of this can be seen in [15] or [2]. Calculating $\mathbf{\Psi}$ can be done by

$$\mathbf{\Psi} = (\mathbf{S}_1^H\mathbf{S}_1)^{-1}\mathbf{S}_1^H\mathbf{S}_2, \tag{3.39}$$

where $(\mathbf{S}_1^H\mathbf{S}_1)^{-1}\mathbf{S}_1^H$ denotes the pseudo inverse of $\mathbf{S}_1$. What remains are to calculate the eigenvalues of $\mathbf{\Psi}$ and the angular argument of those. Then estimates of the p frequencies will be calculated.

3.4.4 Amplitude and phase estimation

When the frequencies are estimated with the SURE algorithm the amplitude and phase can easily be estimated with a least squares estimate. By using the same notation as in Section 3.4.1 the data can be written as

$$\underset{\mathbf{y}}{\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[N-1] \end{bmatrix}} = \underset{\mathbf{O}}{\begin{bmatrix} \mathbf{c} \\ \mathbf{c}\boldsymbol{\Phi} \\ \mathbf{c}\boldsymbol{\Phi}^2 \\ \vdots \\ \mathbf{c}\boldsymbol{\Phi}^{\mathbf{N-1}} \end{bmatrix}} \mathbf{a} + \begin{bmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ w[N-1] \end{bmatrix}, \tag{3.40}$$

where all the terms except the noise and **a** are known. Therefore a least squares estimate of **a** can be computed as

$$\hat{\mathbf{a}} = (\mathbf{O}^H\mathbf{O})^{-1}\mathbf{O}^H\mathbf{y}. \tag{3.41}$$

3.4.5 Adaption to real-valued data

The algorithm is developed for estimation of complex sinusoids but in practice, the beat signal consists of real-valued data. The algorithm works the same except that two changes have to be made. First, twice as many frequencies have to be estimated. This is because a real-valued sinusoid can be written according to

$$A\cos 2\pi f_0 + \phi = A\frac{e^{i(2\pi f_0+\phi)} + e^{-i(2\pi f_0+\phi)}}{2}, \tag{3.42}$$

where A is the amplitude, f_0 the frequency and ϕ the phase. A real sinusoid can thus be rewritten as two complex sinusoids with different signs of the frequencies and the phases. Second, the amplitude needs to be multiplied by a factor of 2, since the complex sinusoid will only have half the amplitude of the real-valued sinusoid.

4 Performance evaluation

In this chapter, the suggested algorithms are evaluated and compared to each other, mostly by simulated data but also by data from a real FMCW radar. In Section 4.1, the definitions of the performance measurements are presented together with how the test case is generated. In Section 4.2, the three chosen methods are evaluated against each other and against the FFT based algorithm. In Section 4.3, a further evaluation of the deconvolution algorithm for simulated data is presented. Section 4.4 presents some results when the deconvolution is evaluated on some real data and compared with the FFT based algorithm. The chapter ends with a summation of the main results in Section 4.5.

4.1 Performance measurements

The methods will be evaluated by simulations in Matlab. To receive a fair comparison of the algorithms, they will be evaluated by Monte Carlo simulations. This means that signals with known parameters are generated and the algorithms then estimate these parameters. The estimated parameters are then compared with the true ones for a large number of simulations with different realizations of the additive noise. The main focus will be to evaluate the accuracy of the estimates of the surface and the resolution performance. The accuracy will be evaluated by calculating the Root Mean Squared Error (RMSE) according to

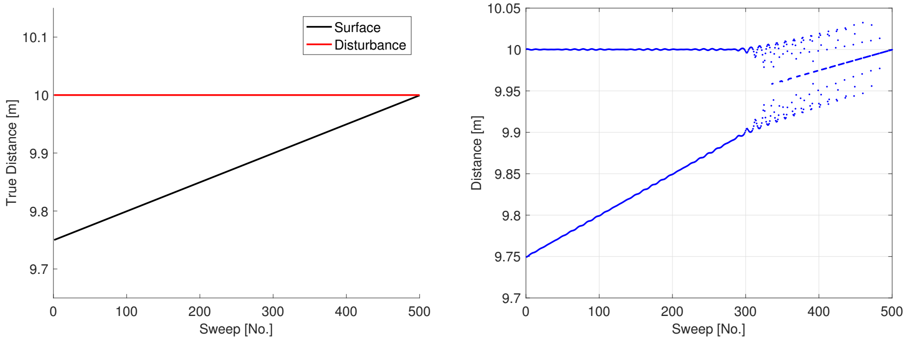
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{4.1}$$

where y_i is the true value of the surface, $\hat{y}_i$ is the estimated position and N is the number of samples. To associate estimates of the echoes with the true positions, a Global Nearest Neighbour (GNN) will be used. As a maximum association limit the FFT resolution limit will be used $\frac{c}{2B}$, thus a bandwidth of 3 GHz corresponds to 50 mm. Consequently, if the true position is at 10 m a valid association is in the range [9.95, 10.05] m. In the case, only one echo is detected the closest of the surface and disturbance will be associated with that single echo. The resolution performance will be determined by finding at what distance between the surface and the disturbance one of them will disappear for the first time.

Throughout this chapter, the FFT is referred to as the combined method with windowing, FFT calculation and interpolation. The leakage compensation algorithm is referred to as leakage in the figures.

4.1.1 Test case

For the evaluation one main setup will be used. In this setup, one main source of disturbance is introduced which is fixed at 10 m with an amplitude of 2000. The surface starts at 9.75 m and slowly (0.5 mm/s) approaches the disturbance. However, the surface positions are not to be seen as a series of measurements one after another, rather as static measurements at a certain distance from the disturbance which decreases with 0.5 mm per sweep. Ideally, the result would look like in Figure 4.1a. Both the surface echo and the constant disturbance echo are well separated all the time until they intersect. With the FFT based algorithm, the result looks more like in Figure 4.1b. When the surface is about 100 mm from the disturbance the two echoes start to interact and sometimes only one peak is found. The first time this happens is what is determined as the resolution performance. The RMSE will then be the average deviation between the black line in Figure 4.1a and the peaks associated with that line from Figure 4.1b.



(a) *(b)*

Figure 4.1: *In (a) the result in a perfect world, in (b) what the result looks like in a realistic case with peak picking from the FFT based algorithm.*

If nothing else is stated in the tests, the following will hold

- SNR: 20 dB,

- Amplitude relation: 1:1 (surface same amplitude as disturbance),
- Auto correlation lags in SURE: 40,
- Number of iterations for deconvolution: 20 (determined from simulations),
- Monte Carlo iterations: 100.

4.1.2 SNR

The SNR in the simulated data is defined as the ratio between the surface amplitude squared and the noise variance i.e.

$$SNR = \frac{A_{surf}^2}{\sigma^2}. \tag{4.2}$$

Expressed in decibel it becomes

$$SNR = 10\log_{10}\left(\frac{A_{surf}^2}{\sigma^2}\right). \tag{4.3}$$

4.1.3 Residual

For the real data, the residual will be calculated. The residual will be calculated by comparing the true value for the surface y and the estimated surface $\hat{y}$ according to

$$Residual = y - \hat{y}, \tag{4.4}$$

where the estimated value is the result from either the FFT or the deconvolution. The residual will then illustrate the estimation error for the method. One problem that occurs when creating the residual is that no true measurement of the surface exists. Using the fact that the FFT provides an accurate estimate in the undisturbed case and that a constant pump speed is used when increasing/decreasing the surface level, it is possible to approximate the surface as a straight line. Therefore a true value will also be available in the area where the disturbances are located. The undisturbed case for real data refers to the case when the surface and the bottom/beam are well separated and the FFT provides accurate estimates under these circumstances.

4.2 Comparison between algorithms

In some cases, the disturbing echo will have a larger amplitude than the surface and the question is if the amplitude difference affects the accuracy and resolution performance. Note that the disturbing echo has larger amplitude when the surface is approaching the bottom especially if there is oil in the tank since the reflection factor $\rho << 1$ and the bottom has $\rho \approx 1$. The result for the resolution performance and accuracy can be seen in Figure 4.2 and 4.3. The FFT, deconvolution and SURE algorithms are calculated for relative amplitude $[0.25 \quad 0.26 \quad ...1]$

and the leakage compensation is calculated for [0.25 0.30 ...1]. A value of 0.25 corresponds to the surface amplitude being 25 % of the disturbing echo's amplitude.

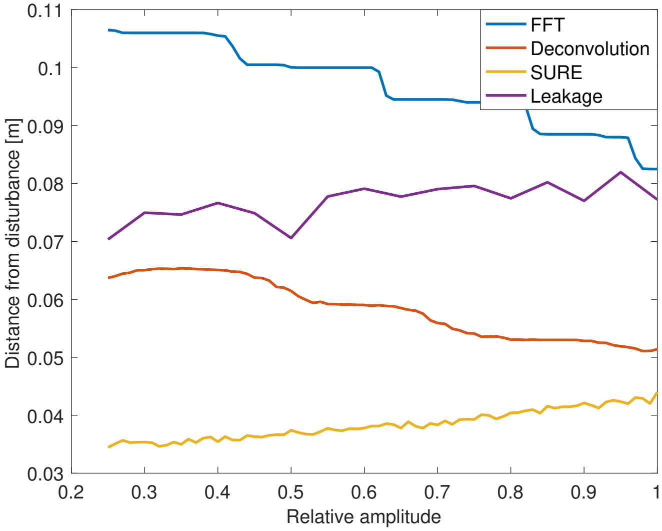


Figure 4.2: *The resolution performance for the four methods for different amplitude relations between disturbance and surface.*

As Figure 4.2 suggests, all the three methods outperform the FFT based method in terms of resolution. The subspace-based method is the one that performs the best, but one has to keep in mind that the algorithm is feed with the true number of ingoing sinusoids. The accuracy, in Figure 4.3, is somewhat the same for the FFT, SURE and deconvolution algorithm and it improves when the relative amplitude approaches one. For the FFT method, the accuracy does a jump when the surface has the same amplitude as the disturbance. That is the case because more peaks after the resolution are lost the first time will be associated with the surface. In general, the associations which are done after the double peak is lost for the first time have a larger offset from the true position. Thus, if more peaks are associated after the first single peak the average error increases which is the case when the amplitude reaches the same level as the disturbance.

The leakage compensation algorithm looks to have a better resolution performance than the FFT but with less accurate estimates. The result is somewhat misleading. By investigating the result in Figure 4.4b, which is from a single Monte Carlo simulation for the leakage compensation, one can see a more realistic case. The same case is also presented for the FFT based algorithm in Figure 4.4a. A few extra peaks are found in the leakage compensation algorithm which provides the extra resolution, but also a few triple peaks are found. This is some-

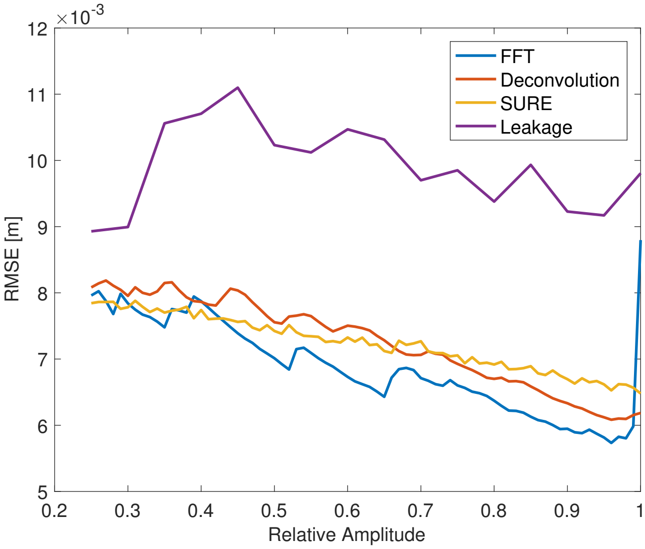


Figure 4.3: *The average RMSE for all four methods for different amplitude relations between disturbance and surface.*

what problematic since so-called ghost peaks is not desired. This is for ideal sinusoids embedded in Gaussian noise. The result would look even worse when the disturbances are added because the energy in the periodogram will then be more spread out, and therefore provide a more inaccurate estimate of the amplitude which will give a worse result.

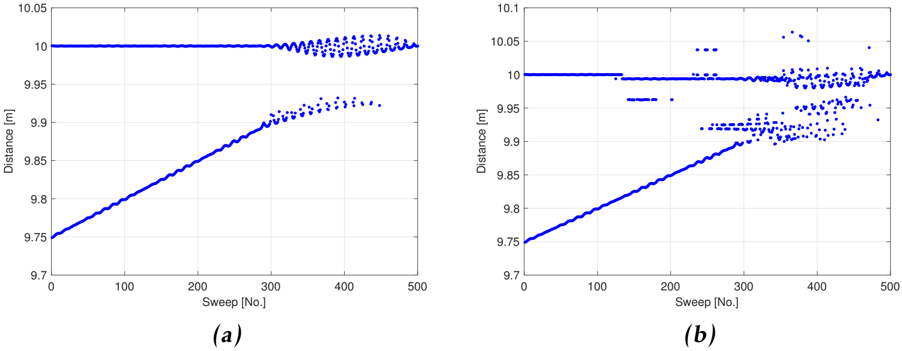
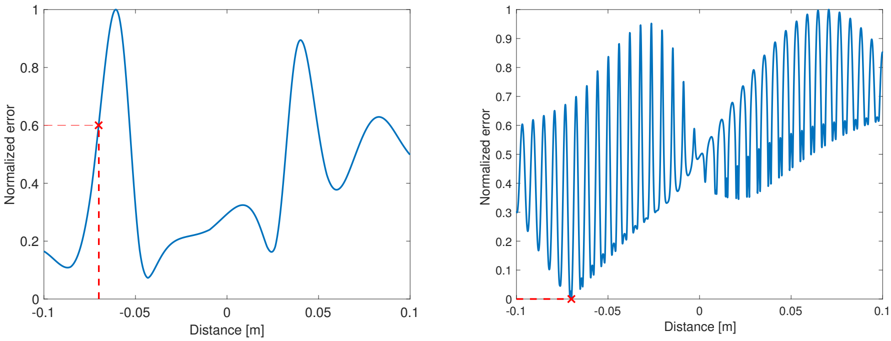


Figure 4.4: *(a) The result from the FFT by peak picking, (b) the result by performing the leakage compensation*

The extra peaks are not desired and it is of interest to find out where they come from. It is best realized by examining the minimization criterion, from equation (3.8), by plotting the criterion in Figure 4.5a. To be sure nothing is missed, the minimization criterion is performed at a denser grid than the leakage compensation algorithm runs. As there can be seen the position where the true minimum is located is not nearby a minimum value in this case. Since it is believed that it is the phase difference between two sinusoids that is the main problem, the minimization function is also plotted when the true periodogram is fitted with the FFT of phase correct sinusoids. The result can be examined in Figure 4.5b. The true minimum is found as a minimum but the error is oscillatory. It oscillates with 6 mm in between every maximum which is the distance it takes for the phase to go from $0-2\pi$. By the insight from these two figures one can realize that even though there is no phase in the FFT periodogram, it still affects the way it looks which will make it hard to do a graphical comparison with estimates of the periodogram. Finding minimums a bit away from the true minimum, are what will add ghost peaks.



(a) *(b)*

Figure 4.5: *The minimization function for the leakage compensation with the red cross being where the true minimum is located, (a) what it looks like when fitted with sum of 2 Taylor windows, (b) what it looks like when phase correct sinusoids are used.*

Due to the poor performance in accuracy, the problem with the phase and the extra ghost peaks the leakage compensation algorithm will not be investigated any further.

The FFT is known to be insensitive to high noise levels, both for accuracy and the resolution performance. Here it will be tested if the other two remaining algorithms also are insensitive to noise. The test case will be the same as described in Section 4.1.1 with the SNR to iterate over. The results can be observed in Figure 4.6 and 4.7.

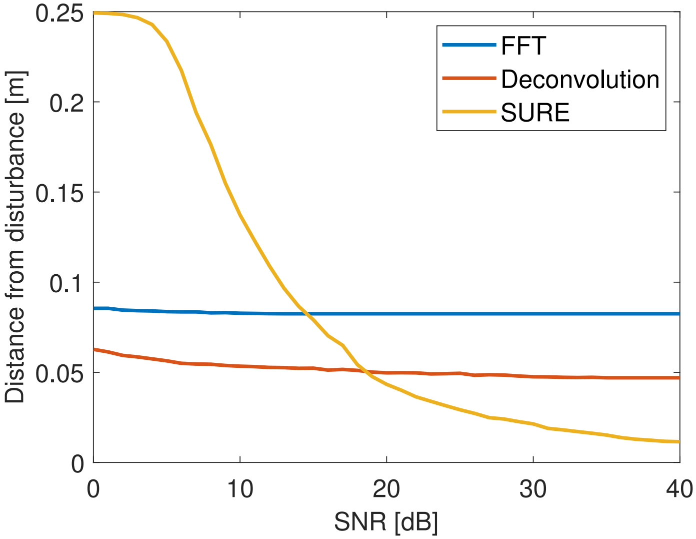


Figure 4.6: *The resolution performance for different SNR.*

In Figures 4.6 and 4.7 one can see that the FFT and the deconvolution algorithms are almost insensitive to noise. There is a slight increase in resolution performance for high SNR both for the FFT and the deconvolution. The subspace algorithm SURE, on the other hand, requires high SNR to provide accurate estimates and acceptable resolution.

The subspace based algorithm SURE outperforms the other algorithms in resolution as long as the SNR is not too low. Even though this is the case, it will not be evaluated any further. This is due to a number of reasons. First of all the problem of choosing the number of ingoing frequencies is neglected. The most obvious idea would be to calculate the FFT and then find the number of peaks, say p, which overcomes a threshold and then estimate $p + 1$ or $p + 2$ frequencies. The estimation of some extra frequencies is because one peak from the FFT can contain more than one echo. Or one could do as suggested in [2] which first estimates a large number of frequencies and then calculates the corresponding amplitudes with equation (3.41). From this, the number of amplitudes which are larger than some threshold is once again estimated. Another problem which is not considered is that the signal, in reality, has passed through a set of filters and how that affects the estimation might be a problem. A third problem occurs when calculating the pseudo inverse in equation (3.39). It happens that the ma-

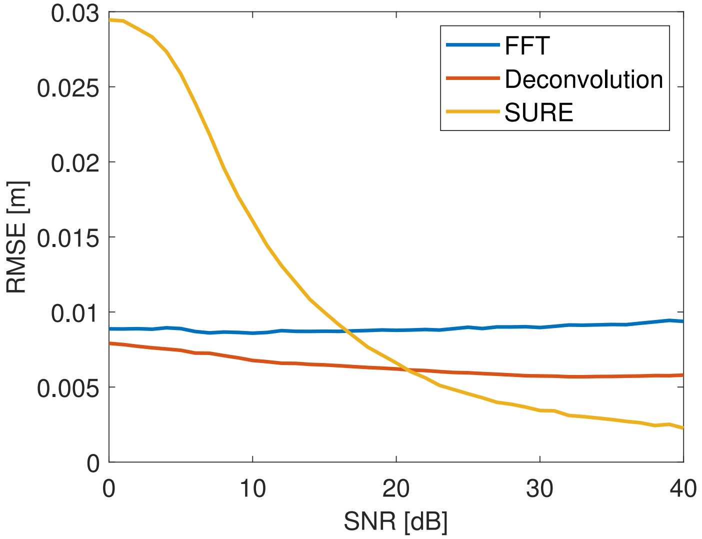


Figure 4.7: *The accuracy for different SNR.*

trices are almost singular which gives inaccurate estimates. The algorithm has also been tried out on real data and provided a worse result than the FFT and the deconvolution algorithm. With all these aspects in consideration the algorithm will not be further analyzed.

4.3 Deconvolution

The deconvolution algorithm is the algorithm that is chosen to be evaluated further. This is because no prior knowledge is needed as in the SURE algorithm and it is not that dependent on the phase differences as the leakage compensation.

The deconvolution algorithm is an iterative algorithm which means that the number of iterations can be changed and the question is if it affects the accuracy and the resolution performance. The result can be seen in Figures 4.9 and 4.8. The resolution and accuracy performance for the FFT are plotted as well to have something to compare with. The accuracy looks to be much better for the deconvolution than the FFT. This is however not the case, by looking back at Figure 4.3 one can observe that the RMSE increases when the surface and disturbance have the exact same amplitude. The accuracy for the FFT when the surface has slightly less amplitude than the disturbance is about the same as what the deconvolution provides in Figure 4.9. The resolution performance in Figure 4.8 on the other hand becomes better and better for increasing number of iterations. The largest increase in resolution is made up to about 20 iterations. By increasing the number of iterations to 100 another 10–15 mm is gained.

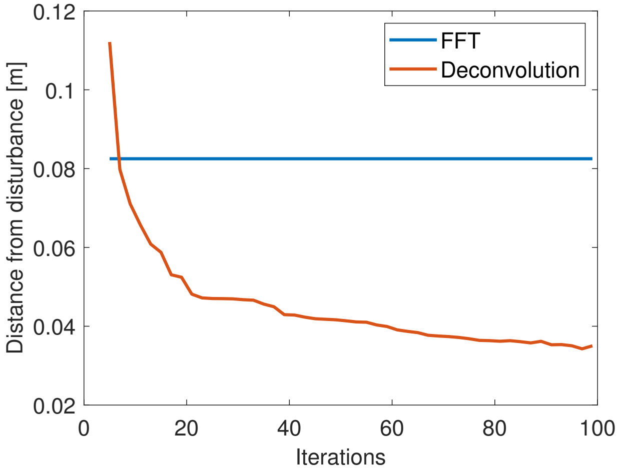


Figure 4.8: *The resolution performance for different number of iterations in the deconvolution algorithm. The resolution performance for the FFT for the same test setup is plotted as a reference.*

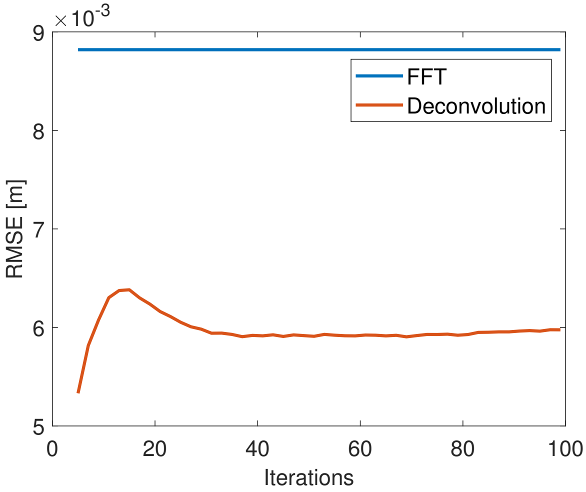


Figure 4.9: *The accuracy for different number of iterations in the deconvolution algorithm. The accuracy for the FFT for the same test setup is plotted as a reference.*

The data this far has been from ideal sinusoids embedded in white Gaussian noise. As mentioned in Chapter 2 some extra disturbances are introduced due to imperfections in the radar unit. In the same way as in Figures 4.3 and 4.2, the amplitude relation will be changed, but instead of comparing the algorithms, the deconvolution will be compared when different disturbances are active. The frequency dispersion adds a time delay in the signal which will lead to a distance offset. The offset will be compensated for by first examining how large offset it introduces. The result can be seen in Figures 4.10 and 4.11. As there can be seen the disturbances have an influence on both the accuracy and the resolution performance. The amplitude modulation affects the result the least which also is expected since it almost does not change the look of the FFT periodogram. The frequency dispersion tends to be the one that provides the least accurate estimates. This is most likely because it blurs the periodogram and provides a wider main lobe which leads to more inaccurate estimates when the surface and disturbance peak interfere with each other. The frequency dispersion also seems to have the same resolution performance regardless of the relative amplitude.

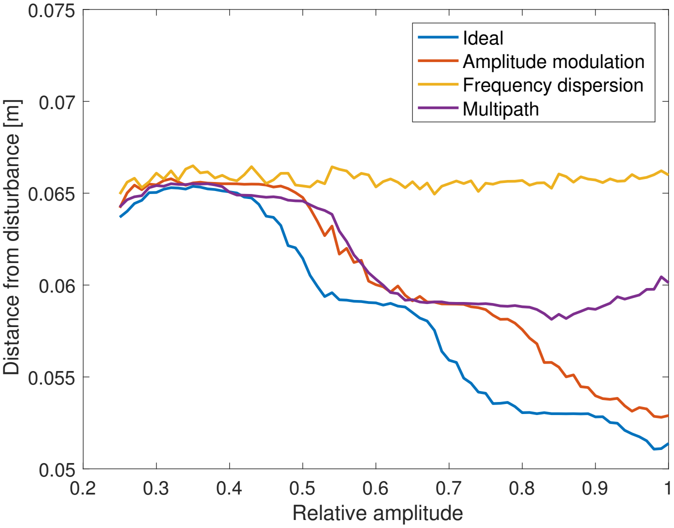


Figure 4.10: *The resolution performance for the deconvolution algorithm when different disturbances are active.*

For an algorithm to be considered useful it also must provide accurate estimates in the undisturbed case i.e., when only one main echo is present which originates from the surface. This is tested by still having the disturbing echo at 10.00 m and then letting the surface go from 8.75–9.00 m. Then investigate the accuracy and compare with the same for the FFT. The result can be investigated in Figure 4.12. It is clearly visible that the deconvolution is less accurate in the undisturbed case than the FFT. However, if the relative amplitude is not too small the precision is within submillimeter level.

Another typical case which is a problem is when there is a thin layer of liquid above the liquid surface, typical oil above water. If the FFT is applied and the

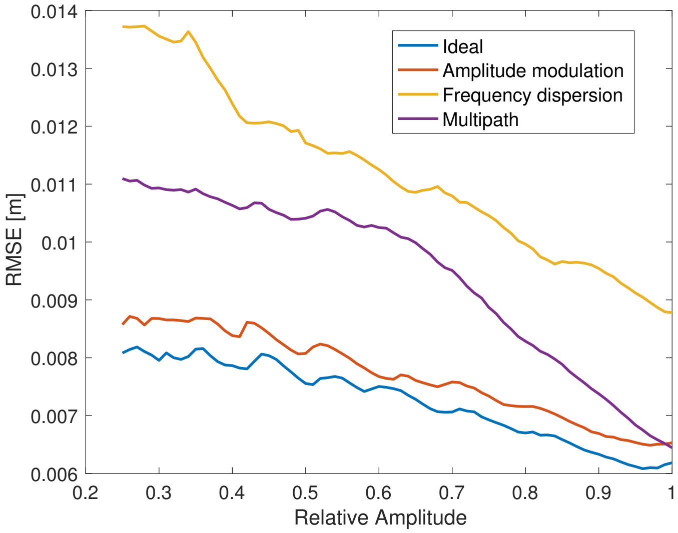


Figure 4.11: *The accuracy for the deconvolution algorithm when different disturbances are active.*

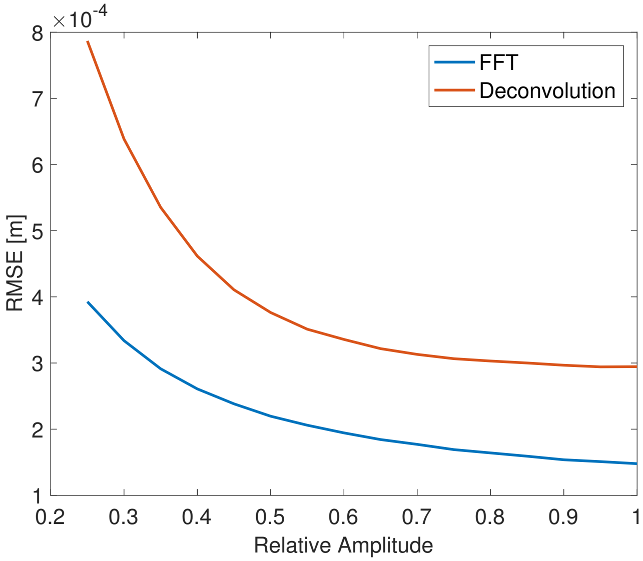


Figure 4.12: *The accuracy for different amplitude relations when the disturbing echo is about 1 m away from the surface echo.*

thickness of the layer is below the resolution limit an error of the surface echo estimate is introduced. This is simulated by having one main echo and one echo moving along with the main echo at a constant distance. The distance is then varied and the RMSE is calculated in the cases when both peaks are well resolved. The result can be seen in Figure 4.13. To have something to compare with, the same plot for the FFT algorithm can be seen in Figure 4.14. For a value to be plotted the two peaks have to be resolved in 99 % of the sweeps. To make a realistic case, oil above water, the echo corresponding to oil will have a relative amplitude of 25 % of the main echo that symbolizes water. This is because the reflection constant ρ is larger for water than for oil.

By comparing Figure 4.13 with 4.14, it can clearly be seen that the deconvolution performs better in terms of resolution. The first time the deconvolution misses the two peaks is when they are about 65 mm apart. The FFT, on the other hand, misses the double peak at 105 mm for the first time. The overall accuracy is worse for the deconvolution but not larger than 30 mm. As expected, the accuracy of the water echo is better than for the oil echo due to the water having a larger amplitude. It was also investigated if the resolution would be better if a lower percentage than 99 % were used. The resolution would be better but only with a few mm in case of lowering it to 50 %. The interesting thing that can be observed is that the average error oscillates and has a maximum peak every 6:th mm. This is due to the influence from the phase which spins one lap around the unit circle each 6:th mm.

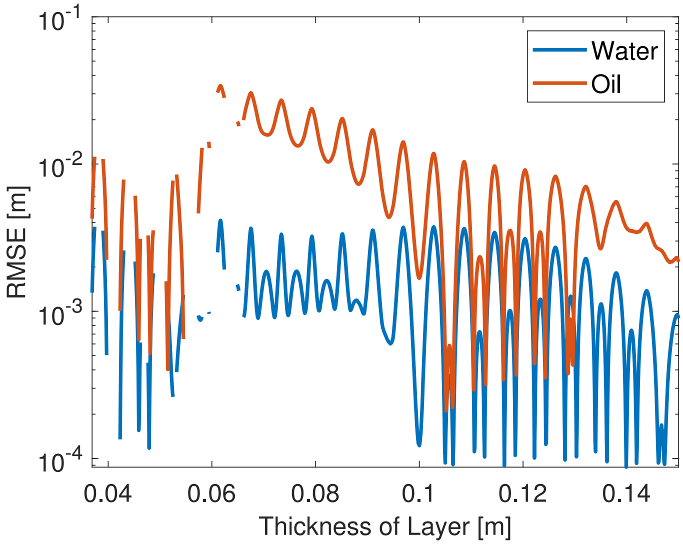


Figure 4.13: *The accuracy for oil and water for the deconvolution algorithm when the resolution requirement is 99 % of the sweeps.*

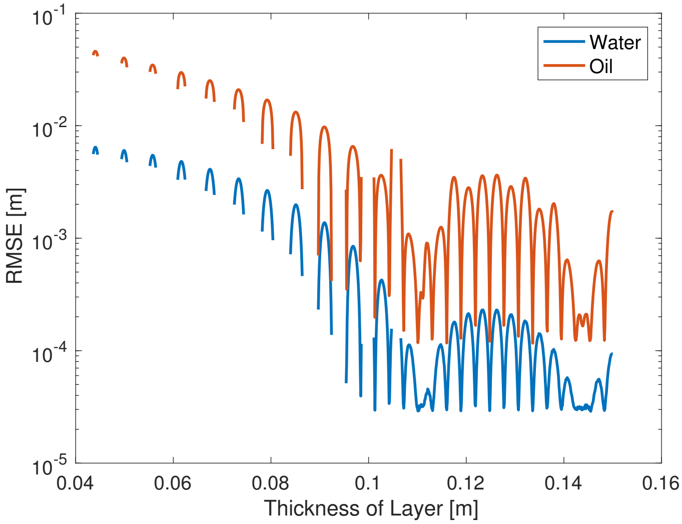


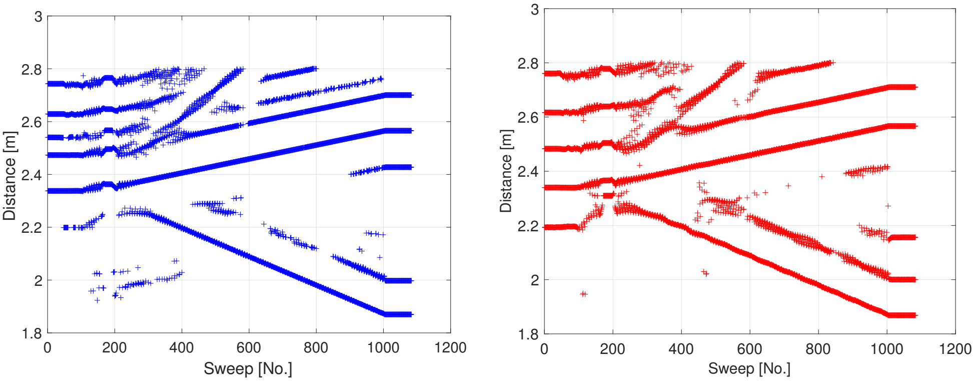
Figure 4.14: *The accuracy for oil and water for the FFT algorithm when the resolution requirement is 99 % of the sweeps.*

4.4 Real data

Finally, some real data are evaluated as well. A variety of data is used, some are from an oil tank and some other are from a measuring path. The data from the measuring path uses a moving metal plate to simulate the surface and fix ones will be the disturbances. To get a variation of data different sensors will be used with different bandwidths. The data with a bandwidth of 3 GHz has a FFT resolution of 100 mm meanwhile the sensor with 1.5 GHz bandwidth has 200 mm. Also, the center frequencies differ and therefore affects how fast the phase changes. The 3 GHz radar has a center frequency of 25 GHz and the 1.5 GHz radar has 10 GHz.

In the data set from the 3 GHz radar, the resolution performance is investigated when the surface is close to the bottom. At first, there is no oil in the tank and then after a while, the tank will start to fill up. The echo from the bottom is approximately four times as strong as the surface echo. To receive an idea of what the data looks like the result from the peak picking in the FFT is presented in Figure 4.15a.

A variation of different numbers of iteration are tested and the best is presented in Figure 4.15b together with the residual of the fit in Figure 4.16. Some of the other results can be investigated in Appendix A.1. By comparing the result one can see that the deconvolution provides continuous estimates approximately 50 mm longer than the FFT. However, the accuracy of the gained resolution varies from a few mm to about 50 mm. By examining the residuals in Appendix A.1 one can see that the accuracy when surface and bottom is well separated becomes worse the more iterations used. Also, the resolution is the same for 20–50 iterations. From this one can realize that more iterations used, does not provide a better result, neither in resolution nor accuracy.



(a) *(b)*

Figure 4.15: *(a) the result from the FFT by peak picking, (b) the result from 20 iterations with deconvolution.*

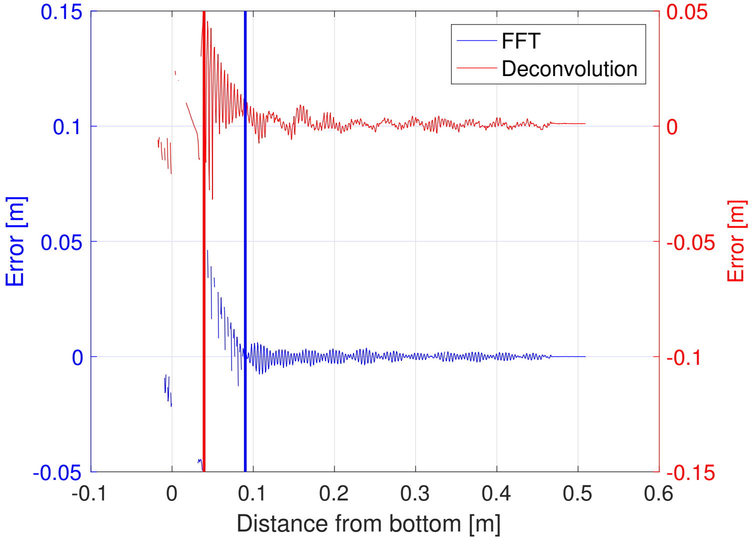
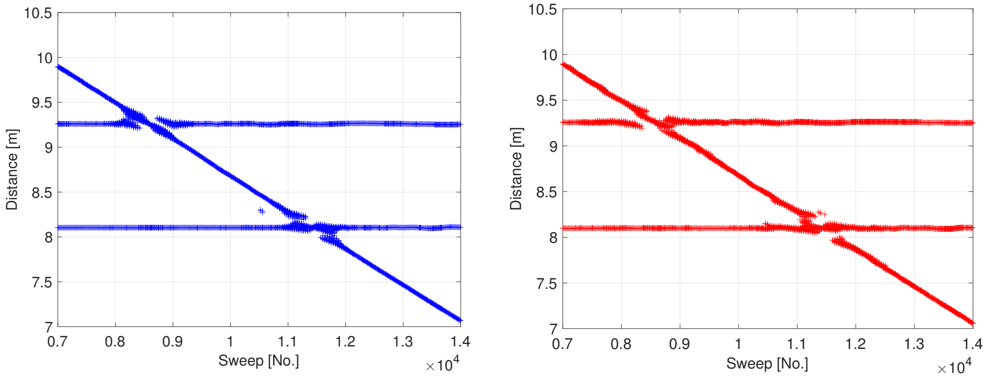


Figure 4.16: *The residual for both the FFT and the deconvolution algorithm. The blue vertical line is at what distance from the bottom the FFT loses the double peak for the first time. The red is the same but for the deconvolution.*

The next data set is from a measuring path where two different passages are done. One that has less amplitude than the metal plate corresponding to the surface and one that has a larger amplitude. The bandwidth in this radar is 1.5 GHz, thus a resolution on the FFT of 200 mm. To get an understanding of what the data looks like, the result from peak picking with FFT is presented in Figure 4.17a. Also this time a number of different iterations are evaluated, the best is presented in Figure 4.17b together with its residual in Figure 4.18. The rest can be examined in Appendix A.3. The first observation which can be done from Figure 4.17b, is that the result is less accurate in the undisturbed cases than the FFT. This is the case no matter how many iterations used. The second observation which is done is that the resolution has increased both at the left and right side of the disturbance with large amplitude in the residual. The gain is about 30-40 mm, thus only about 15-20% of the FFT resolution limit meanwhile for the 3 GHz radar about 50 % is gained. The amplitude relation between the surface and the disturbing echoes are about the same in the data sets, thus is the comparison made applicable. The passage with smaller amplitude looks to have about the same accuracy for the deconvolution as for the FFT.

Another data set is being investigated in the same way as the above in Appendix A.2 which has 1.5 GHz bandwidth. The trend for that data is the same as for the data from the measuring path i.e., worse accuracy over the entire data set but better resolution nearby the disturbance.



(a) *(b)*

Figure 4.17: *(a) the result from the FFT by peak picking, (b) the result from 30 iterations with deconvolution zoomed in at the disturbances.*

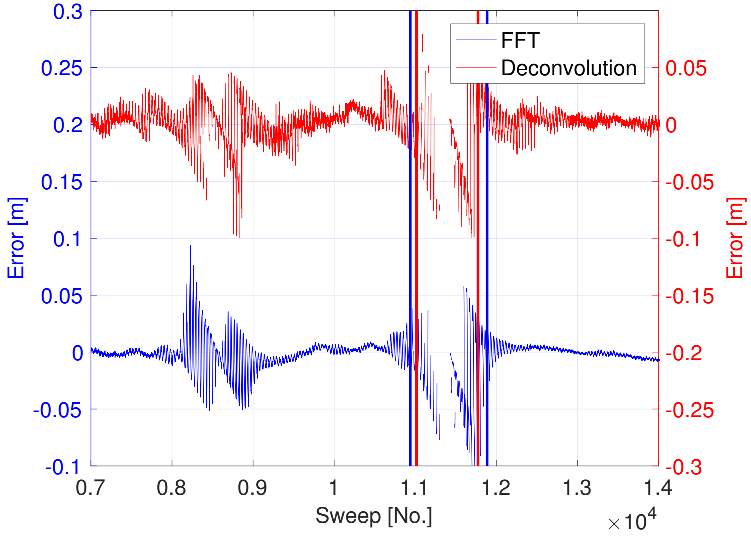


Figure 4.18: *The residual for both the FFT and the deconvolution algorithm. The left blue vertical line is when the surface echo is lost for the first time approaching the disturbance at 8 meter and the right is the last lost peak when moving away from the disturbance. The red lines are the same but for the deconvolution.*

4.5 Summary and discussion

In this section, a summary of the investigated methods are presented together with a discussion of the pros and cons of them.

4.5.1 Leakage compensation

The leakage compensation algorithm is considered to be the algorithm that gives the overall worst result. It is better than the FFT in terms of resolution but the accuracy of the surface estimates is poor. Ghost peaks appearing due to false minimums found in the minimization, is a large problem. The main reason for this is believed to be because of the phase influence. The phase makes it hard to create a minimization criterion which does not oscillate. Also, the zero-pad length has a role in this. In the algorithm, the minimization is performed every 6.25 mm (zero-padded to 8:th times the original data length). This is also a reason for not finding the true minimum in the cases when the true minimum is possible to find. Even if the minimization would be performed at a denser grid the true minimum would not always be a minimum or even close to a minimum as proven in Figure 4.5a.

One idea that came up when working with the leakage compensation was if it would have worked better if there was no phase in the signal. This is tried out but not presented and the result is decent if the minimization is performed denser than earlier. This proves that the influence from the phase is a big problem when trying to compensate for the leakage for close frequencies. The result is not perfect, still, some ghost peaks appear but fewer than earlier.

The amplitude estimation is also erroneous. The error in the estimation becomes larger the closer the frequencies are because the main lobes will interact and amplify/reduce the amplitudes of the observed main lobe. If the main lobes are amplifying or reducing each other depends on the phase difference between the sinusoids.

4.5.2 SURE

The SURE algorithm outperforms all the other algorithms in resolution performance as long as the SNR is not too low. In terms of accuracy, it performs about the same as the FFT and the deconvolution algorithm. However, it was not chosen to evaluate any further because of a number of reasons. The main problem is how to choose the number of frequencies in the signal. For simulated data the problem is not that large, a few ideas were presented earlier. For real data, on the other hand, there are quite a few disturbances, not just the ones that are modeled in the report but also double bounces, reflections from the wall of the tank etc. This makes it a difficult problem to choose the model order. The subspace methods were also evaluated in an earlier thesis and the overall conclusion after that thesis was that the correct model order is hard to choose for real data.

4.5.3 Deconvolution

The deconvolution algorithm is the one considered to be the best, all tests considered. It has the same robustness as the FFT to noise and amplitude relations but better resolution performance. The resolution performance increases with the number of iterations but the largest gain is up to about 20 iterations. Of the three modeled disturbances the amplitude modulation is the one which affects the result the least. The accuracy is most affected by the frequency dispersion. This is reasonable since it blurs the periodogram and provides a wider main lobe. With a wider main lobe, the deconvolution algorithm will believe that the FFT periodogram originates from more than one frequency and therefore find a double peak. The peaks will then interact and thus lead to a worse accuracy.

As a second step, the deconvolution algorithm is evaluated on some real data. The main erudition is that a suitable number of iterations in the algorithm is needed for it to perform optimally. About 20–30 iterations seem to be what works the best since the largest gain in resolution is up to about 20–30 iterations. With more iterations, the accuracy becomes worse than the FFT algorithm, but no further gain in terms of resolution. This is especially the case for the data from the radar with 3 GHz bandwidth. The reason for the accuracy being worse with more iterations is believed to be because more peaks are found nearby the surface echo. With more peaks close to the surface echo, they will interfere with each other and therefore contribute to less accuracy.

The accuracy being worse in the 1.5 GHz radar regardless the number of iterations is somewhat strange. However, the main lobe in the FFT is of 200 mm, which makes the first echo from the multipath imperfection to appear inside the main lobe. The guess is that the deconvolution algorithm finds this peak with a few iterations. It will then interfere with the main peak and thus provide an inaccurate estimate of the main echo. The peak from the multipath will necessarily not show up as a peak in the peak picking, either it is below the threshold or it will not show up as a clear peak and only affect the look of the deconvoluted periodogram.

The residuals created for the real data must also be treated with some skepticism since they are created by approximating the result from the FFT with a few straight lines since no ground truth exists. But it can not be a too poor approximation since it is known by experience that the FFT provides accurate estimates in the undisturbed case. Another thing that needs to be held in mind is that the accuracy of the deconvolution increases when the resolution is lost with the FFT. One can say that the gain is 40–50 mm (for 3 GHz radar) of continuous measurements but that the error in the worst case could be around 50 mm.

5 Conclusion

This chapter sums up the thesis and offers some suggestions for future work. In Section 5.1, the questions from Chapter 1 are answered. Section 5.2 ends the chapter by providing some suggestions for future work that might improve the performance.

5.1 Concluding remarks

The goal of this thesis work has been to investigate methods that overcome the FFT resolution limit for close frequencies. Three such methods have been evaluated and compared to each other and against the FFT based algorithm. The deconvolution algorithm was considered to be the best of them and was therefore evaluated more in depth than the other two algorithms. The leakage compensation algorithm largest problem was that it does not find the minimum at the correct place due to the influence of the phase in the periodogram. The subspace-based method SURE was also evaluated but was sensitive to noise and the fact that prior knowledge of the number of frequencies observed was needed, it was not seen as a useful method.

The resolution performance was affected by the amplitude relation between the surface echo and the disturbing echo in the deconvolution algorithm but not more than it affected the result in the FFT algorithm. The noise level did not affect the accuracy and resolution performance as long as the SNR was not too low. With low SNR, the noise spikes get above the threshold in the peak picking. In the real data, this was not a problem since the SNR was high. The deterministic disturbances were a larger problem especially in cases when they affected the appearance of the main lobe, which was the case for the frequency dispersion. When using radars with 1.5 GHz bandwidth the multipath imperfection also appeared within the main lobe and thus contributed to inaccurate estimates even in the undisturbed case. Hence, it is important to have large enough bandwidth so as few disturbances as possible gets within the main lobe.

The overall conclusion from this work is that it is a well-studied problem to resolve frequencies below the FFT resolution limit. Of the investigated methods, none outperforms the FFT based algorithm in all tests. The FFT provides accurate estimates of the surface position until the resolution is lost. The investigated methods provide about the same accuracy as the FFT as long as the FFT can resolve the frequencies meanwhile, the accuracy of the gained resolution is poor. The deconvolution which was considered to perform the best gained about 40–50 mm extra resolution (3 GHz radar) but the accuracy of those were poor. In the best case, an error of only a few mm was accomplished but in worst case about 50 mm.

5.2 Future work

The deconvolution algorithm is well tested especially for simulated data for a wide range of test setups, but there are a few ideas of how to develop it. One idea that is discussed during the thesis is if it would be possible to use a different PSF. At the moment the FFT of a Taylor window is used, but the idea is instead to use the FFT of the signal including all the disturbances as in [10]. For the simulated data that would mean to use a PSF looking something as the result in Figure 2.6 when all disturbances are added. For the real data, one has to use the FFT of an undisturbed signal with only one main echo and the radar imperfections. For the real data, two main problems appear. Firstly every radar will have their own so called impulse response (FFT of an undisturbed case), can all use the same PSF or must each radar unit first be calibrated? The second problem is that the radar unit operates in a wide range of environments. The imperfections will differ quite a lot when the temperature changes and the impulse response will change accordingly. Would it then be possible to model the temperature dependence in the impulse response?

Another aspect which is outside the scope of this thesis, but still of great importance is the computational burden. This is something that needs further investigations because the signal processor that performs the signal processing is not that powerful, the real-time requirements still need to be fulfilled. One thing that affects the computational time is the number of iterations which is performed. For real data, more iterations did not increase the resolution performance rather decreased the accuracy. Finding a suitable number of iterations is thus important. This might be improved by trying a Lucy Richardson algorithm with a stop criterion as in [1] instead of a fixed number of iterations.

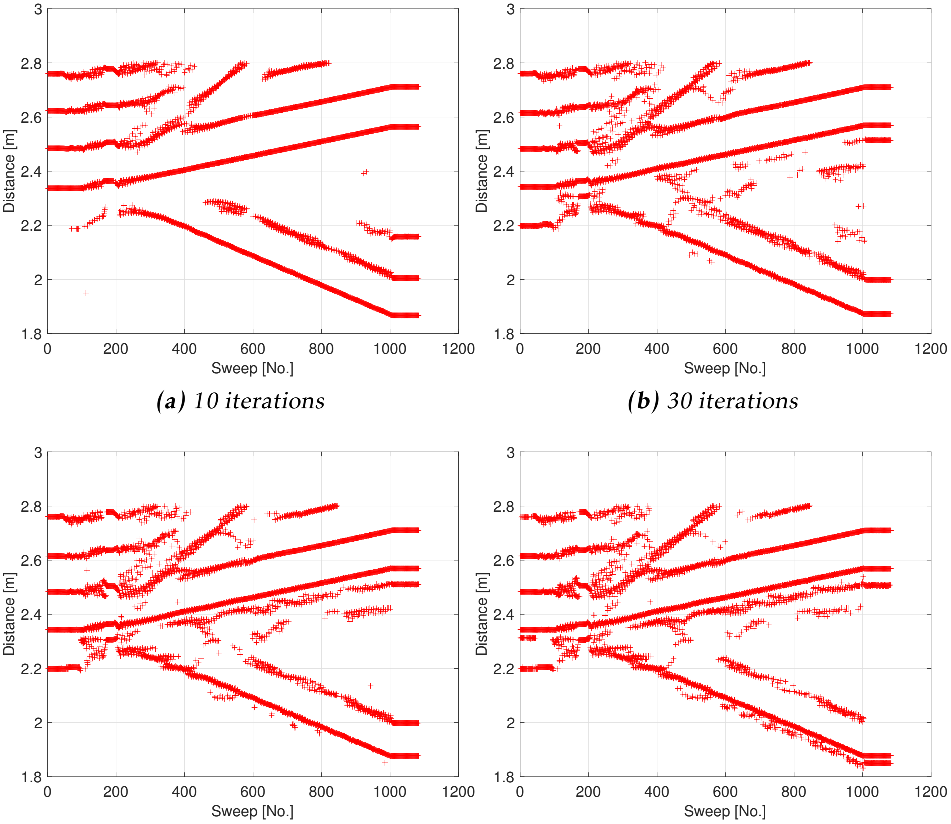
Appendix

A

Real data

A.1 Oil tank 3 GHz bandwidth

Here is the rest of the result from the oil tank with 3 GHz bandwidth presented with the result from peak picking and the corresponding residuals.



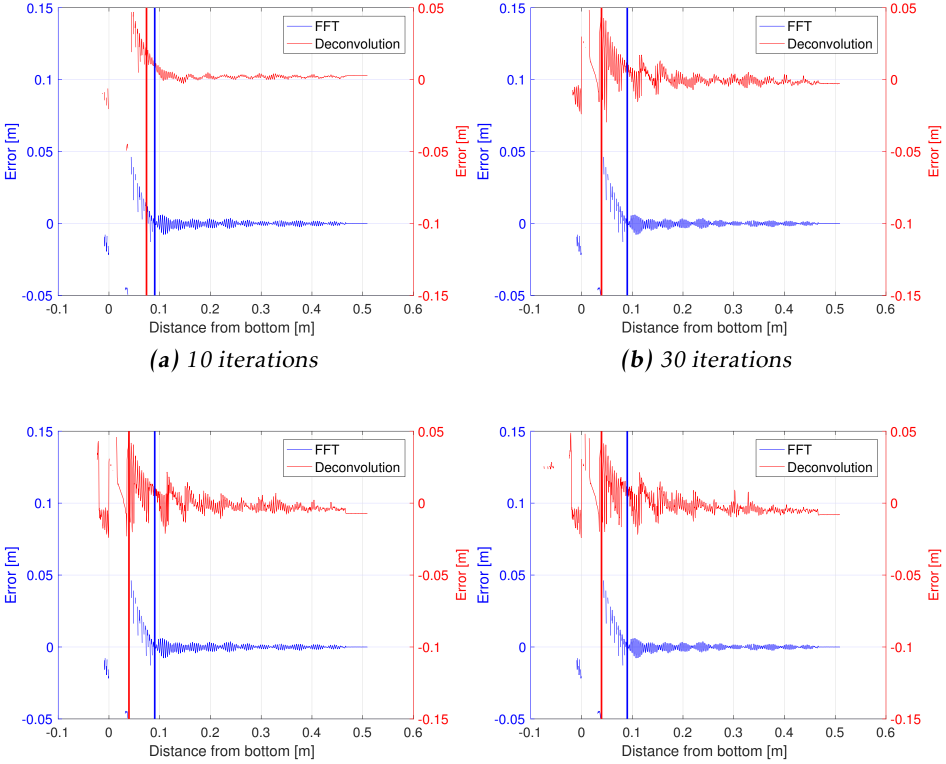
(a) 10 iterations

(b) 30 iterations

(c) 40 iterations

(d) 50 iterations

Figure A.1: *The result by peak picking from the deconvolution algorithm for different number of iterations.*



(a) 10 iterations

(b) 30 iterations

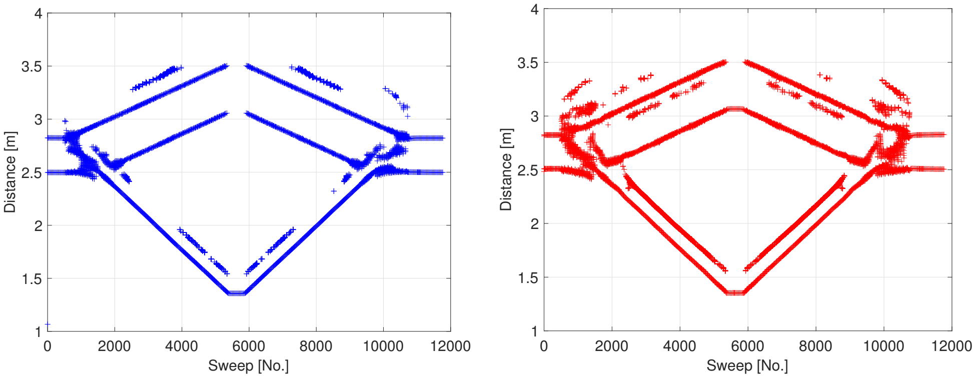
(c) 40 iterations

(d) 50 iterations

Figure A.2: *The residual for both the FFT and the deconvolution algorithm. The blue vertical line is at what distance from the bottom the FFT loses the double peak for the first time. The red is the same but for the deconvolution.*

A.2 Oil tank 1.5 GHz bandwidth

This data set is from a run where the oil level increases from the bottom and then passes a beam which is located about 300 mm from the bottom. The result from the FFT can be seen in figure A.3a. The best result from peak picking with deconvolution can be seen in figure A.3b which is with 20 iterations. The residual for the same number of iterations can be seen in figure A.4. Some results from different number of iterations can be seen in figure A.5 and A.6.



(a) *FFT*

(b) *Deconvolution 20 iterations.*

Figure A.3: *(a) The result from peak picking with the FFT algorithm, (b) the same but from the deconvolution with 20 iterations which is determined to be the best.*

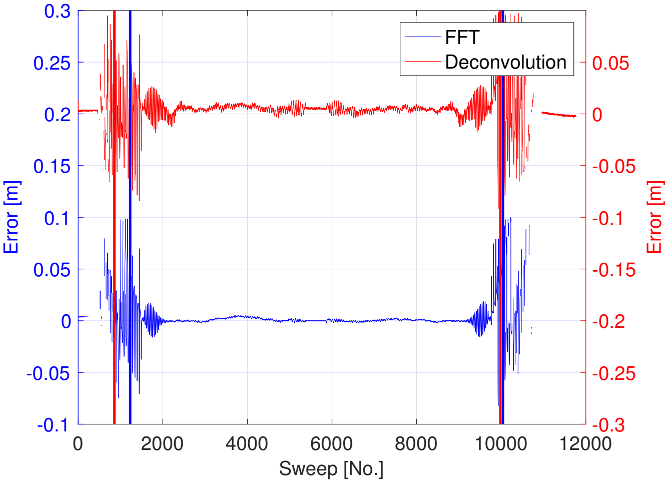
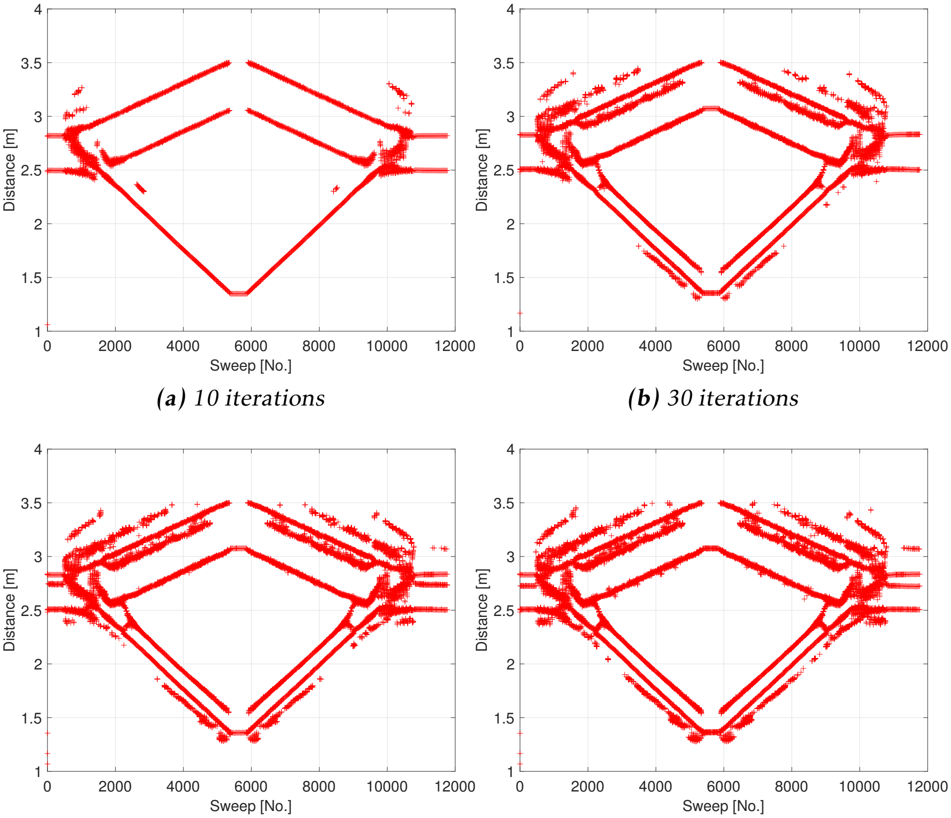


Figure A.4: *The residual for the FFT and the deconvolution algorithm with 20 iterations. The blue vertical lines are at what sweep from the bottom the FFT loses the double peak for the first time. The red ones are the same but for the deconvolution.*



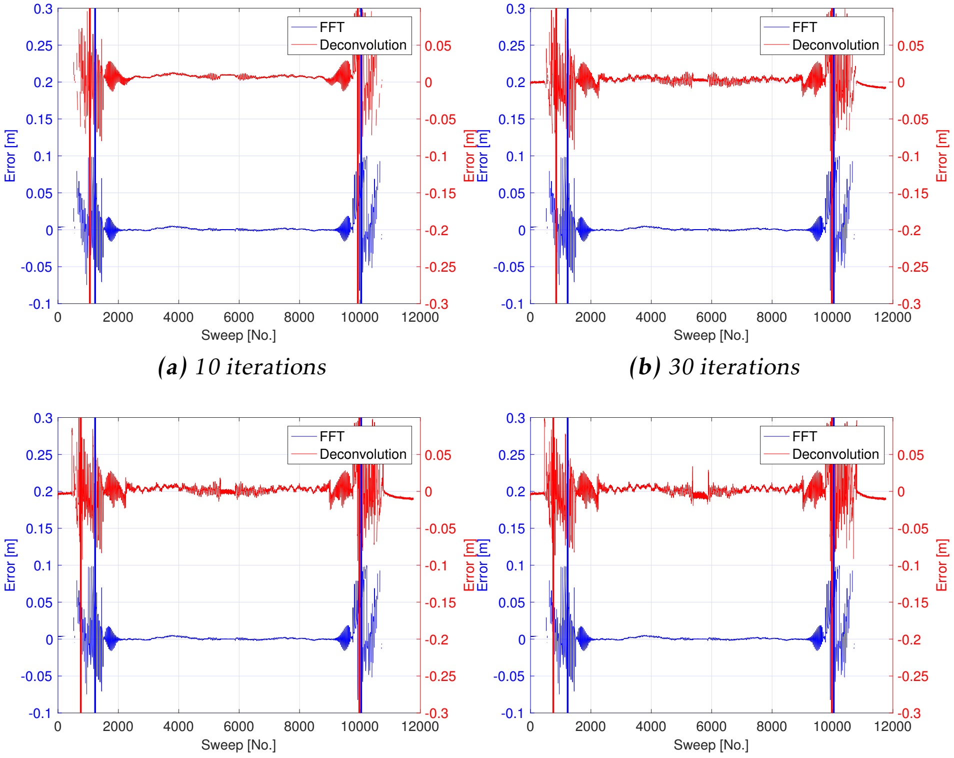
*(**a**) 10 iterations*

*(**b**) 30 iterations*

*(**c**) 40 iterations*

*(**d**) 50 iterations*

Figure A.5: *The result by peak picking from the deconvolution algorithm for different number of iterations.*



*(**a**) 10 iterations*

*(**b**) 30 iterations*

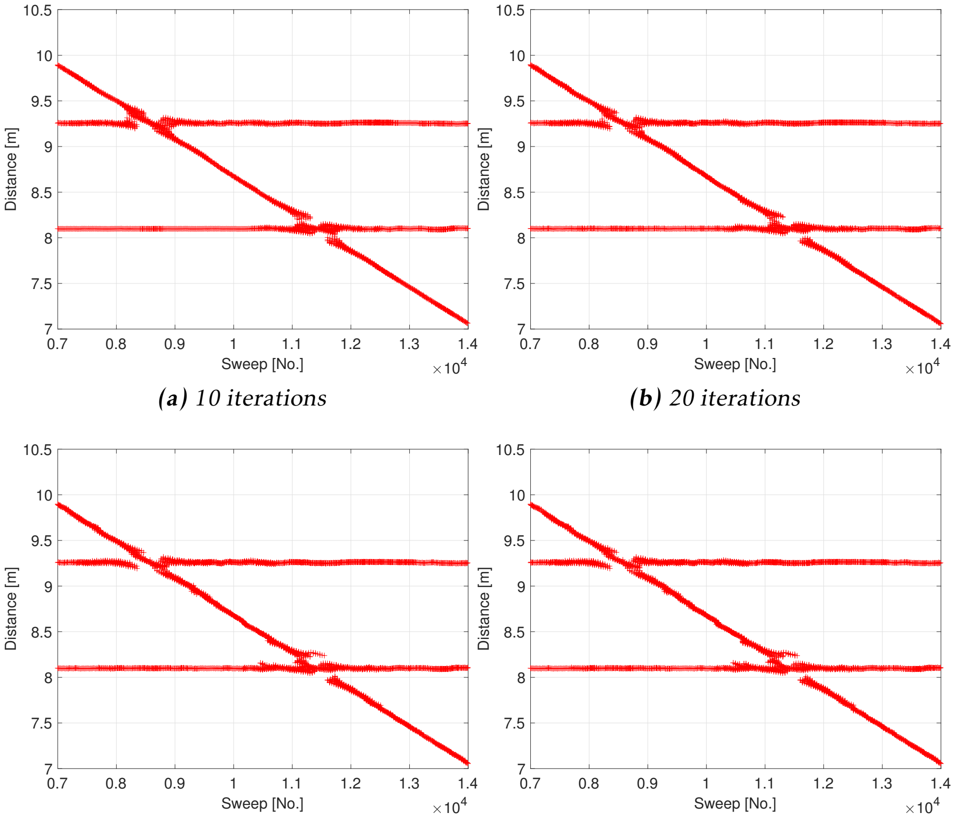
*(**c**) 40 iterations*

*(**d**) 50 iterations*

Figure A.6: *The residual for both the FFT and the deconvolution algorithm. The blue vertical lines are at what sweep from the bottom the FFT loses the double peak for the first time. The red ones are the same but for the deconvolution.*

A.3 Measuring path 1.5 GHz bandwidth

Here is the rest of the result from the measuring path presented with the result from peak picking and the corresponding residuals.



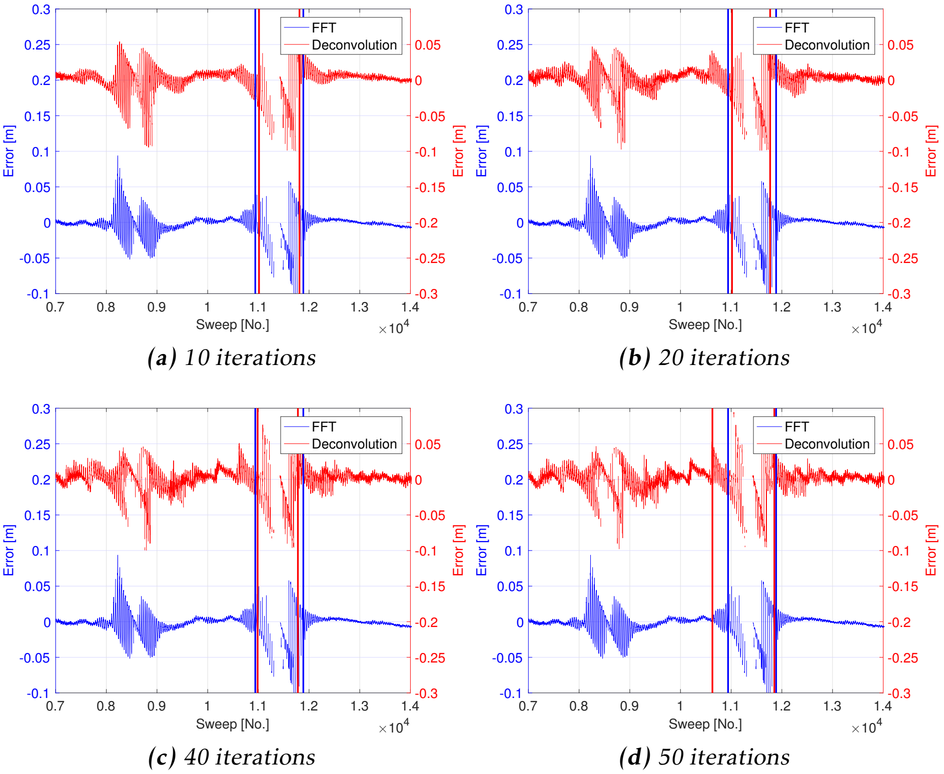
(a) 10 iterations

(b) 20 iterations

(c) 40 iterations

(d) 50 iterations

Figure A.7: *The result from peak picking with the deconvolution algorithm for different number of iterations.*



(a) 10 iterations

(b) 20 iterations

(c) 40 iterations

(d) 50 iterations

Figure A.8: *The residual for both the FFT and the deconvolution algorithm. The left blue vertical line is when the surface echo is lost for the first time approaching the disturbance at 8 meter and the right is last lost peak when moving away from the disturbance. The red lines are the same but for the deconvolution.*

B

Matlab code

B.1 Refined peak search

Hear is the code for the leakage compensation presented.

```
function [Peaks] = refined_peak_search(PowerSpectra, peaks,Setting,...
                                        Distance,AmplitudeScale)
% Function that fits the given FFT spectra with 2 taylorwindows.
% Peaks = Matrix containing information of the peaks found
%
% PowerSpectra = Powerspectra from FFT.
% peaks:  Matrix containing information of the peaks found from the
% FFT spectra
% Setting = Information about the signal, bandwidth etc.
% Distance= at what distances the Powerspectra is evaluated
% AmplitudeScale = scaling factor to get correct amplitude


% Peaks(px,1), Amplitude for peak
% Peaks(px,2), Peak Index
% Peaks(px,3), Peak Start index
% Peaks(px,4), Peak Stop index
% Peaks(px,5), Position [m]
% Peaks(px,6), Check if peak is already investigated

M = Setting.Conf.FFT_LEN;
delta_h = Distance(2)-Distance(1); % how dense the minimization is done

l = 1;
while (l <= Setting.Conf.MaxNrPeaks)
    % peaks are stored in order, peak positon = 0, same as no more peaks
        so
    % jump out of while loop
    if (peaks(l,5) == 0)
```

```
   break;
end
% if peak investigated jump to next
if peaks(l,6) == 1
    l = l+1;
else
    % Perform amplitude estimation accoording to equation 3.7 in
        report
    window1 = taylorwin(Setting.Sweep.Ns,8,-44);
    WIN1 = fft(window1,M);
    WIN1 = WIN1.*conj(WIN1);
    WIN1 = WIN1*max(peaks(l,1))^2/AmplitudeScale^2/max(WIN1);
    % sum up values from 10 peaks before the peak start to 10 values
    % after it ends
    PY = sum(PowerSpectra(max(peaks(l,3)-10,1):min(peaks(l,4)+10,M)));
    PW1 = sum(WIN1);

    % Amplitude shuld not be larger than 1 since all peaks with larger
    % amplitude is already investigated.
    A = min([sqrt(abs((PY-PW1)/PW1)),1]);

    % if amplitude is maller than 15 % of max peak in spectra, do
    % nothing since the peaks tends to end up random.
    if (A < 0.15*max(PowerSpectra)/((peaks(l,1)^2)/(AmplitudeScale^2))
        )
        l = l+1;
        continue;
    end

    window2 = A*taylorwin(Setting.Sweep.Ns,8,-44); % create the second
         window.

    % where to start withe second window
    distance_start = max(peaks(l,5) - 0.1,0);

    % assuming the peak with large amplitude will be almost fixed
    distance_fixed = peaks(l,5);
    % finding the corresponding index
    [~, ix] = min(abs(Distance-peaks(l,5)));

    % error matrix
    error = zeros(5,33);

    distance = distance_start;

    % finding index of where to start
    [~, ix1] = min(abs(Distance-distance));

    % searthing 2*delta_h in each direction for main peak
    for n = -2:2
        WIN1 = fftshift(fft(window1, Setting.Conf.FFT_LEN));

        k = ix  + n;
        % rotate the window so it end up at correct frequncy
        WIN1 =  [WIN1(M/2 -k + 1: M/2); WIN1(M/2+1:M); WIN1(1:M/2-k)];
        % loop for the the window with amplitude A
        for i = 0:32
            k1 = ix1 + i + n;
        WIN2 = fftshift(fft(window2, M));
        % Rotate second window to correct frequncy
        WIN2 =  [WIN2(M/2 - k1+ 1:M/2); WIN2(M/2+1:M); WIN2(1:M/2-
            k1)];

        % Add them together and scale them so they have the same
        % amplitude as the investigated peak
        WIN_tot = WIN1+WIN2;

        WIN_tot = WIN_tot.*conj(WIN_tot);

        WIN_tot = WIN_tot*peaks(1,1)^2/(AmplitudeScale^2)/max(
            WIN_tot);
        % calculate the error.
        error(n+3,i+1) = fit(PowerSpectra,WIN_tot,k);
    end
end
% find minimum error in error matrix
[~, I] = min(error(:));
[I_row, I_col] = ind2sub(size(error),I);

% calculate what value the minimum corresponds to in distance.
h_min = distance_start +   delta_h*(I_col-1) +(I_row-3) *delta_h;


% find closest peak except the one that is investigated
Q = abs(peaks(1,5)-peaks(:,5));
[p, ~] = min(Q(Q>0));
[~,closest_found_peak] = min(abs(p-Q));

% if peak already investigated and closer than 100 mm continue
    with
% next peak
if (peaks(closest_found_peak,6) == 1 && Q(closest_found_peak) < 0
    .1)
    l = l+1;
    continue;
end

% The following tries to figure out if a new peak should be
% inserted or correcting an already exsisting peak.
tmp_varible = 0;
% check if an old peak is being corrected
if (abs(h_min - peaks(1,5)) > 0.01) && (abs(h_min - peaks(1,5)) <
    0.1)...
        && (abs(peaks(1,5)-peaks(closest_found_peak,5)) < 0.1)
    peaks(closest_found_peak,5) = h_min;
    peaks(closest_found_peak,6) = 1;
    tmp_varible = 1;
end

% In case no correction of old peak is done insert a new peak
a = find(peaks(:,5)==0,1,'first');

if isempty(a)
    a = Setting.Conf.MaxNrPeaks;
end
```

```
        % new peak must be more than 30 mm away from the main peak to keep
        % the number of ghost peaks low. (This would be the resolution
        % limit if it would work perfect)
        if ((abs(peaks(l,5) -h_min) > 0.03)  && (tmp_varible == 0) && (a ~
            = l))
            peaks(a,5) = h_min;
            peaks(a,1) = peaks(l,1)*A;
            peaks(a,6) = 1;

        end

        % See if main peak needs to be moved.
        if (abs(I_row-3)*delta_h) > 0
            peaks(l,5) = distance_fixed +(I_row-3)*delta_h;
        end
        peaks(l,6) = 1;
        l = l+1;
    end
end
Peaks = peaks;
end
```

B.2 SURE

The same code as used in [2] but a few changes is made to update it against a new matlab version.

```
function [f, ampl, phase, r] = sure(y, M, p, fs)
%SURE   SUbspace Rotation Estimates of sinusoidal frequencies.
%   [f, ampl, r] = sure(y, M, p, fs)
%
%   Returns the frequencies of the p sinusoidal components of the signal
%   sequence y, using M autocorrelation lags. Sampling frequency fs.
%
%  y = signal sequence
%  M = number of auto correlation lags
%  p = number of complex sinusoids
%  fs = sampling frequency
%
%  f = estimated frequencies of the p sinusoidal components
%  ampl = estimated amplitudes
%  phase = estimated initial phases


[n, k] = size(y);
if (k > n)
  y=y';
  [n, k] = size(y);
end;

yc=y(1:M);
yr=y(M:n);
Y=hankel(yc,yr);                % Form the data matrix Y.
```

```
R_yy = Y*Y';                    % Covariance matrix

[V, D]= eig(R_yy);

D=diag(D);

[~, Index] = sort(D);
Index=Index(M:-1:1);

V = V(:,Index);

S = V(:,1:p);                       % Signal subspace
G = V(:,p+1:M);                     % Noise subspace

S1 = S(1:M-1,:);

phi = inv(S1'*S1)*S1'*S2;
eig_phi = eig(phi);

f = angle(eig_phi)/(2*pi)*fs;
r = abs(eig_phi);


% Estimation of amplitude
O = exp(1i*2*pi*(0:n-1)'*f'/fs);    % Observability matrix

x0 = inv(O'*O)*(O'*y);           % Least-squares solution
ampl = abs(x0);
phase = angle(x0);
```

C
Flow chart

A few definitions are done which is used in the flow chart in Figure C.1.

- p_{invest} the peak being investigated which is found in the peak picking from the FFT.
- p_{close} the closest peak to p_{invest} found by peak picking in the FFT.
- h_{min} the distance corresponding to the found minimum.

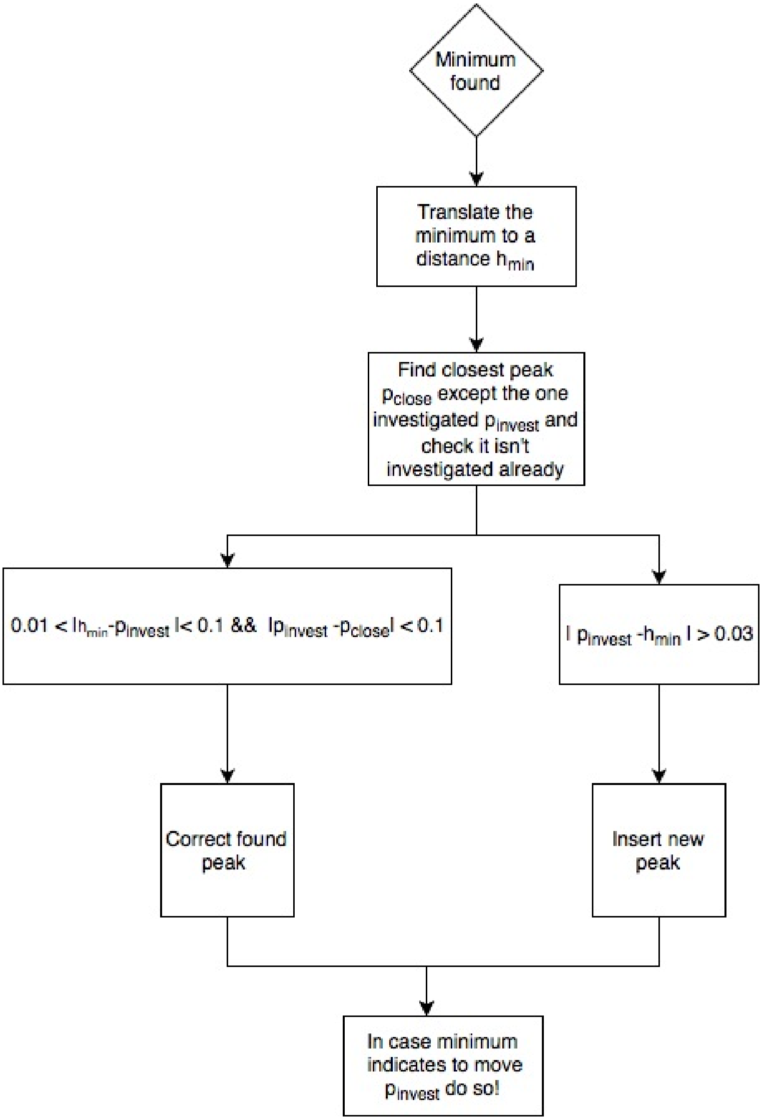

Minimum found
Translate the minimum to a distance h_{min}
Find closest peak p_{close} except the one investigated p_{invest} and check it isn't investigated already
$0.01 < |h_{min} - p_{invest}| < 0.1$ && $|p_{invest} - p_{close}| < 0.1$
$| p_{invest} - h_{min} | > 0.03$
Correct found peak
Insert new peak
In case minimum indicates to move p_{invest} do so!

Figure C.1: *Flow chart over the decision structure in refined peak search. This is to determine if the found peak corresponds to move an already found peak, insert a new peak or assume that the minimum is close enough to the investigated peak and thus do nothing.*

Bibliography

[1] G. L. Almedia, M. I. Silvani, E. S. Souza, and R. T. Lopes. A stopping criterion to halt iterations at the Richardson-Lucy deconvolution of radiographic images. *Journal of Physics: Conference Series*, 2015. Cited on page 48.

[2] M. Andersson. Högupplösande frekvensskattning i en FMCW-radar tillämpning, 1996. Cited on pages 24, 25, 27, 35, and 64.

[3] D. Biggs and M. Andrews. Acceleration of Iterative Image Restoration Algorithms. *Applied Optics*, vol. 36, 1997. Cited on pages 23 and 24.

[4] K. Dae-Bong and H. Sun-Mog. Multiple-target tracking and track management for an FMCW radar network. *EURASIP Journal on Advances in Signal Processing*, 2013(1), Oct 2013. Cited on page 2.

[5] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J-C. Olivo-Martin, and J. Zerubia. 3D Microscopy Deconvolution using Richardson-Lucy Algorithm with Total Variation Regularization. 2004. Cited on page 23.

[6] A. W. Doerry. Catalog of Window Taper Functions for Sidelobe Control, 2017. Cited on page 15.

[7] P-C. Gong, J. Li, W. Guo, and Z. Shao. A high resolution algorithm based on Chirp Z-transform for FMCW radar. In *2015 IEEE International Conference on Communication Problem-Solving (ICCP)*, pages 482–484, 2015. Cited on page 2.

[8] F. Gustafsson, L. Ljung, and M. Millnert. *Signal Processing*. Studentliteratur AB, 1:3 edition, 2010. Cited on page 21.

[9] The Mathworks Inc. Taylorwin, 2018. URL `https://www.mathworks.com/help/signal/ref/taylorwin.html?s_tid=doc_ta`. Cited on page 15.

[10] J. Li, F. Rodrigue-Morales, J. B. Yan, D. Gomez-Garcia, A. Masud, A. Patel, T. Gaynes, C. Leuschen, and S. Gogineni. Deconvolution of FMCW radars for Operation IceBridge missions. In *2014 United States National Committee of*

URSI National Radio Science Meeting (USNC-URSI NRSM), pages 1–1, 2014. Cited on page 48.

[11] L. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 70, 1974. Cited on page 22.

[12] W. H. Richardson. Bayesian-Based Iterative Method of Image Restoration. *Journal of the optical society of America*, 62, 1970. Cited on page 22.

[13] S. Scherr, S. Ayhan, B. Fischbach, A. Bhutani, M. Pauli, and T. Zwick. An Efficient Frequency and Phase Estimation Algorithm With CRB Performance for FMCW Radar Applications. *IEEE Transactions on Instrumentation and Measurement*, 64(7):1868–1875, 2015. Cited on page 2.

[14] S. Scherr, S. Ayhan, M. Pauli, W. Winkler, and T. Zwick. Parametric estimation of distances with a 122 ghz FMCW radar system. In *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, 2016. Cited on page 2.

[15] P. Stoica and T. Soderstrom. Statistical analysis of MUSIC and subspace rotation estimates of sinusoidal frequencies. *IEEE Transactions on Signal Processing*, 39(8):1836–1847, 1991. Cited on pages 25 and 27.