Master's Thesis 2013

Candidate: Ensieh Taghimohammadi

Title: Remote Monitoring by Impulse Radar

Telemark University College
Faculty of Technology
Kjølnes
3914 Porsgrunn
Norway
Lower Degree Programmes – M.Sc. Programmes – Ph.D. Programmes

Telemark University College
Faculty of Technology
M.Sc. Programme

MASTER'S THESIS, COURSE CODE FMH606

Student: **Ensieh Taghi Mohammadi Khasheh**

Thesis title: **Remote Monitoring by Impulse Radar**

Signature: . . . .

Number of pages: 64

Keywords: Impulse Radar, Ultra Wideband (UWB), Clutter, Threshold, Tracking, Kalman filter, MATLAB

Supervisor: **Ole Mrius Lysaker** sign.: . . . .

2nd Supervisor: **Nils Olav Skeie** sign.: . . . .

Censor: sign.: . . . .

External partner: Novelda AS sign.: . . . .

Availability: open

Archive approval (supervisor signature): sign.: . . . . **Date**

Abstract:

This master thesis is centered on development of signal processing algorithms for an Ultra – Wideband (UWB) Radar system. The goal of signal processing algorithms is to identify components of radar received signal. Moreover, implementing algorithms for checking both static and moving objects, estimating the distance from an object, and tracking the moving object.

In this thesis we use a new type of Novelda UWB radar for indoor applications. It consists of two compact directional UWB antennas and a CMOS transceiver for the purpose of signal transmission and reception.

Theoretical analysis for the provided algorithms and their evaluations by practical measurements are presented in this thesis. All the signal processing algorithms are performed in MATLAB 2012a.

Telemark University College accepts no responsibility for results and conclusions presented in this report.

Contents

Overview of Figures

Nomenclature

List of symbols and abbreviations used in the thesis are included here.

Symbol	**Description**
$\boldsymbol{V}$	Measurement Vector
$\boldsymbol{V_C}$	Clutter Vector
$\boldsymbol{V_t}$	Object Vector
$\boldsymbol{e}$	Noise
$\boldsymbol{A}$	Measurement Matrix
$\boldsymbol{A_C}$	Clutter Matrix
$\boldsymbol{A_t}$	Object Matrix
$\boldsymbol{E}$	Noise Matrix

Abbreviations

Abbreviation	Description
A/D	Analogue to Digital
CTBV	Continuous Time Binary Valued
CMOS	Complimentary Metal Oxide semiconductor
DC	Direct or continuous current
FCC	Federal Communications Commission
GHz	GigaHertz
I/O	Input- Output
MATLAB	Matrix Laboratory
NVA	Novelda
PCB	Printed Circuit Board
RADAR	RAdio Detection And Ranging
RX	RiecIver
SPI	Serial Peripheral Interface
SVD	Singular Value Decomposition
TX	Transmitter
USB	Universal Serial Bus
UWB	Ultra-Wideband

Preface

This thesis is submitted to the Department of Faculty of Technology at Telemark University College (TUC), in partial fulfillment of the requirements for the award of Master of Science (MSc.) degree in Systems and Control Engineering. The thesis is written under the supervision of Associate Professor Ola Marius Lysaker.

The main focus of this thesis is working on development of signal processing algorithms for a new type of Ultra-Wideband Novelda radar in MATLAB.

I give a sincere gratitude to all the people who made it possible for me to write this thesis. I also want to thank to the Associate Professor Ola Marius Lysaker for his guidance, support, comments and arranging weekly meetings, which helped me to have good control on the progress of the thesis. I would like to thank to Morten Borg who helped me at the beginning time that I become familiar with Impulse Radar systems. Finally, I would like to thank Novelda AS Company, especially Kjell-Ivar Hosum for his valuable suggestions on different aspects.

Porsgrunn
04-06-2013
Ensieh

1 Introduction

The objective of this thesis is developing signal processing algorithms for an Ultra -Wideband radar system from Novelda AS. More specific information about Novelda radar system can be found in [1].

1.1 Background

The idea of measuring distance with electromagnetic waves goes back to many years ago. In 1904 Christian Hülsmeyer (Huelsmeyer), a German inventor patented a device that he named Telemobiloscope. Telemobiloscope could emit radio waves to objects with metallic surface for detecting the presence of them. Later he improved Telemobiloscope such that it could detect both presence of objects and also measure the distance to an object. Hülsmeyer's Telemobiloscope is what we call RADAR today. From the second half of the 1930s Germany, U.K., USA and Russia independently developed radar systems for detection and tracking of objects. The radar has been continuously progressed since world war two (WWII) and it has become a crucial equipment in air defense, early warning system for incoming airplanes, shipping, geological surveying and more. In the last years, Ultra-Wideband (UWB) technology has introduced a new type of radar. UWB radar technology has made enormous advances in industry and created a growing way for development of daily life applications. An example of such a system can be found in [2-5].

1.2 Objective of the Thesis

Novelda UWB radar system which is used in this thesis is shown in Figure *1.2*. It consists of a set of two directional Vivaldi antennas, a radar chip which is based on CMOS technology and an USB interface cable. The radar chip is mounted on a PCB board with a microcontroller and it can communicate serially with a PC via USB. The microcontroller handles all communication between the computer and the radar chip. On the PCB there are two connections which are provided for connecting the transmitting and receiving antennas to the radar chip.

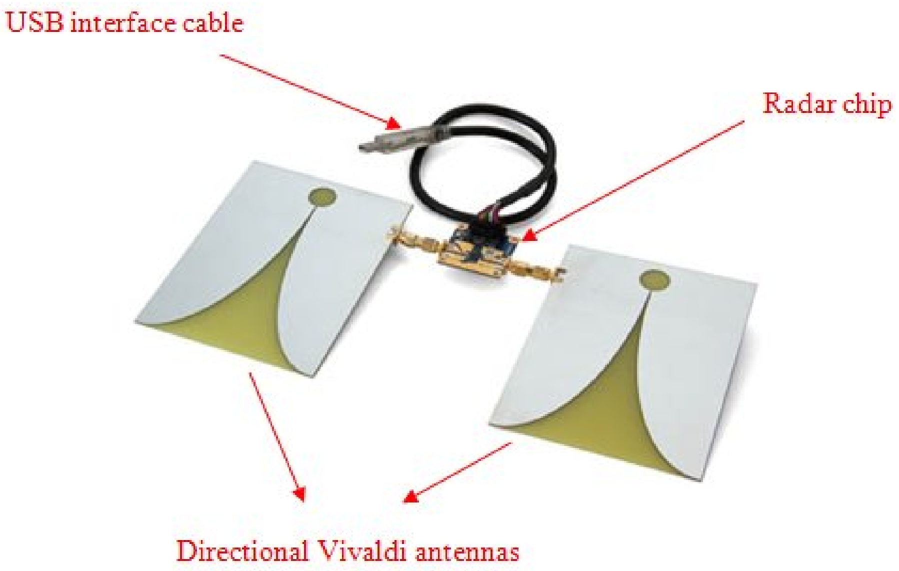


Figure 1.1: *Novelda UWB radar system with a pair of directional Vivaldi antennas.*

A basic principle of UWB Impulse Radar is shown in Figure *1.2*. UWB radar generates and transmits short pulses through the transmitter antenna TX and then the signal propagates in an environment. When the signal meets an object, some part of the electromagnetic energy is reflected from the object and propagates back to receiver antenna RX. The time delay between the transmitted and received signal represents the distance between the radar and the object. This concept is shown in Figure 1.3. Radar received signal has some data about presence of an object in the environment, as well as object characteristics, such as range and velocity. Besides, radar received signal consists of three components including clutter, the reflections from the object, and noise. We need to retrieve the signal of interest by eliminating the clutter and noise. Then we can develop our signal processing algorithms for the UWB radar system.

The main focus of this work is divided into the following parts:

UWB radar system analysis: Describing the particular behavior and properties of the Novelda UWB radar system.

Filtering the radar received signal: Identifying different component of radar received signal and removing the clutter and noise as the unwanted parts of the received signal.

Algorithm design: Developing signal processing algorithms to find a relationship between radar received signal amplitude and distance, detecting both static and moving objects, and also tracking moving object.

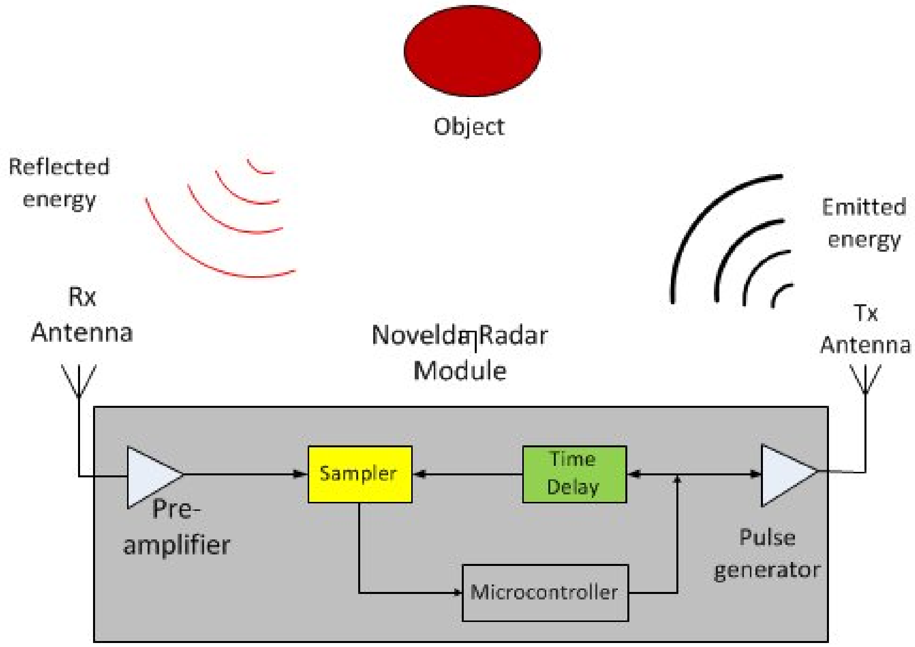


Figure 1.2*: Basic principle UWB radar system.*

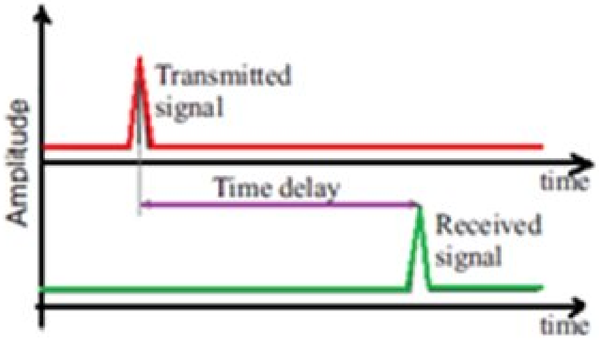


Figure 1.3*: Time delay principle in UWB radar.*

1.3 Structure of the Thesis

Chapter 2 explains the Basic principle of Novelda’s UWB impulse radar and fundamental parameters of Novelda UWB radar are also introduced.

Chapter 3 includes filter's algorithms to remove clutter and noise form radar received signal [6]. It includes DC offset Removal filter to remove the offset from radar received signal and SVD-based filter to reject the clutter and noise.

In Chapter 4 some signal processing algorithms are implemented in order to work with radar technology. These algorithms include:

Deriving a relationship between radar received signal amplitude and distance as an empirical equation.

Implementing an algorithm using the radar signals to check static objects.

Extending the algorithm to check both moving and static objects.

In Chapter 5 Kalman filter method as a fast and reliable algorithm is employed for tracking the moving object to measure and predict the correct position and velocity of the moving object [7]. This application is based on one dimensional radar tracking.

Chapter 6 includes conclusions of the work has been done and recommendations for future work for remote monitoring by impulse radar.

Chapter 7 includes list of the references used in this report.

Chapter 8 includes Appendix. It contains MATLAB code used for simulation in this thesis.

2 UWB Impulse Radar Chip

This chapter will present introducing of Ultra-Wideband (UWB) technology and taking advantages of this technology in Impulse Radar. In addition, functionality of Novelda Impulse Radar and important parameters of Novelda radar is introduced. At the end connection of Impulse radar to PC is shown and discussed.

2.1 Introduction to UWB Technology

In February 2004 the Federal Communications Commission (FCC) released wide spectrum allocation for unlicensed application. This spectrum allocation is from 3.1–10.6 GHz which giving a broad bandwidth of 7.5 GHz. The allocation of this unlicensed spectrum to Ultra-Wideband (UWB) technology made enormous advances in industry and created a growing way for development of daily life applications based on UWB technology.

Recently UWB technology has gained significant interest in research and development in different applications. One of the applications of UWB radar system is in automobile traffic control systems to keep from collision while driving and parking. UWB radars are also used in medical applications which they can provide remote measuring of heart beats and other vital signs of the patient. Another usage of the UWB radar can be measuring distances to an object with high accuracy [8-10].

Figure *2.1* illustrates the comparison between conventional Narrowband signals with Ultra-Wideband signals in both time and frequency domain. The conventional Narrowband radio signal can transmit and receive periodic sinusoidal signals with a very narrow frequency spectrum in both transmission and reception. However, an Ultra-Wideband radio uses pulses with very short duration time and very wide frequency band [11, 12]. Figure *2.1* shows the reference Frequency in Narrow-Band signals is 2.4 GHz, however the spectrum allocation in UWB signal is from 3.1–10.6 GHz which gives a broad bandwidth of 7.5 GHz.

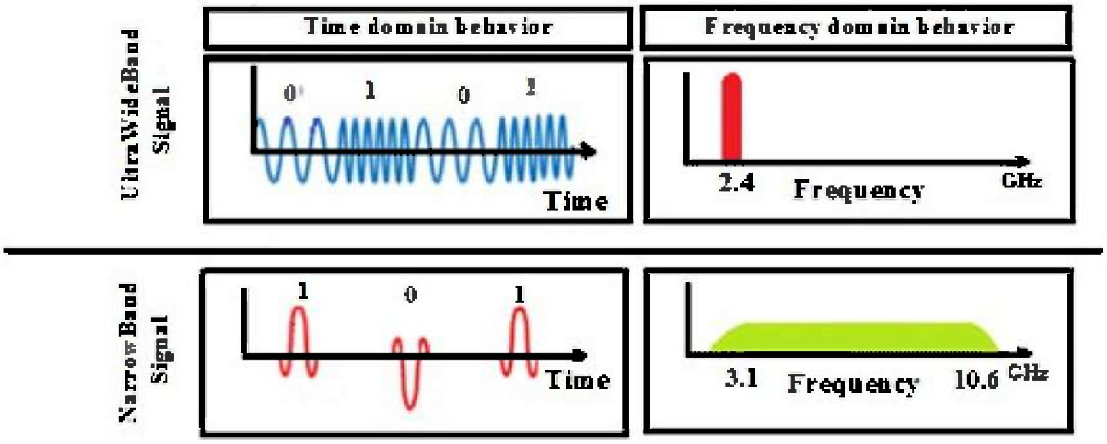


Figure 2.1*: Narrowband vs. UWB in time and frequency domain.*

2.2 Impulse Radar Based on UWB Technology

The basic principal of radar technology, which was originally an acronym for "**RA**dio **D**etection **A**nd **R**anging", has a history goes back to Heinrich Hertz's classical experiments in the 1880's [13]. He showed that electromagnetic waves have the ability to reflect from the surface of the objects. An Impulse Radar is a device can measure the distance to an object by sending and receiving electromagnetic pulses. The radar antenna transmits electromagnetic wave pulses to an object and then the object returns some part of the wave is reflected from the object [14].

As Impulse radars were initially being developed, they were based on working by Narrowband pulses. Impulse Radar technology based on Narrowband pulses did not have any practical application. Since electronic circuits of this impulse radar were very simple, generating and receiving such Narrowband pulses were not possible. But after some improvement in integrated circuits, it was possible to use this technology with both Narrowband and UWB signals.

Unique attributes of UWB impulse radar such as high penetration abilities, high accuracy and low power level has caused such systems are more practical and efficient than traditional Narrowband systems [15].

Today, impulse UWB radar transceivers are becoming available in practical applications.

One such UWB radar transceiver has been developed by a Norwegian company is Novelda AS [1]. In this thesis, Novelda NVA-641 impulse radar chip has been used which is based on CMOS technology.

2.3 The Chip Architecture of the Novelda Impulse Radar

Novelda Impulse Radar works based on Strobed Sampler concept which shown in *Figure* 2.2. The main functionality of a sampler is to capture an analogue input signal and store it as digital values. The Strobed Sampler which used in Novelda Impulse Radar operates in a way that a signal at several gigahertz is converted from continuous time to discrete time mode. As can be seen in *Figure* 2.2, for each analogue pulse transmitted from TX antenna, the backscattered analogue signal is sampled after a given specified time offset. Time offset represents the time-of-flight of the signal relative to the time of transmission, which in turn can be used to represent the distance to the object.

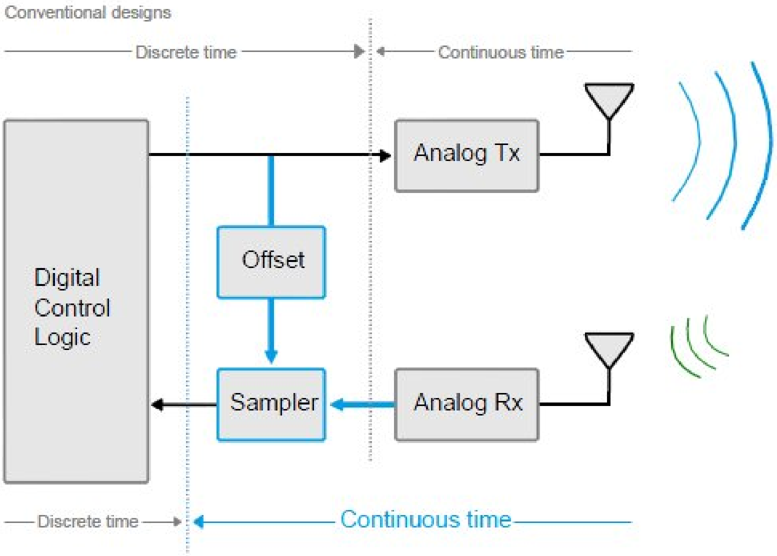


Figure 2.2*: Impulse Radar using Strobed Sampling [16].*

The Novelda Impulse Radar has improved the basic Srobed Sampler by taking advantages of Continuous Time Binary Valued (CTBV) method. In this case each analogue signal is converted to CTBV code by a high speed 1-bit A/D quantizer. In other words each analogue received signal is converted to a digital sequence values like (11001011001…). Then this triggered signal is connected to the parallel samplers, where each sampler has a slightly different time offset t. Therefore by using this method, the samplers sample the incoming CTBV and convert them as normal digital values [16]. Figure 2.3 shows block diagram of Novelda Impulse Radar based on the CTBV method.

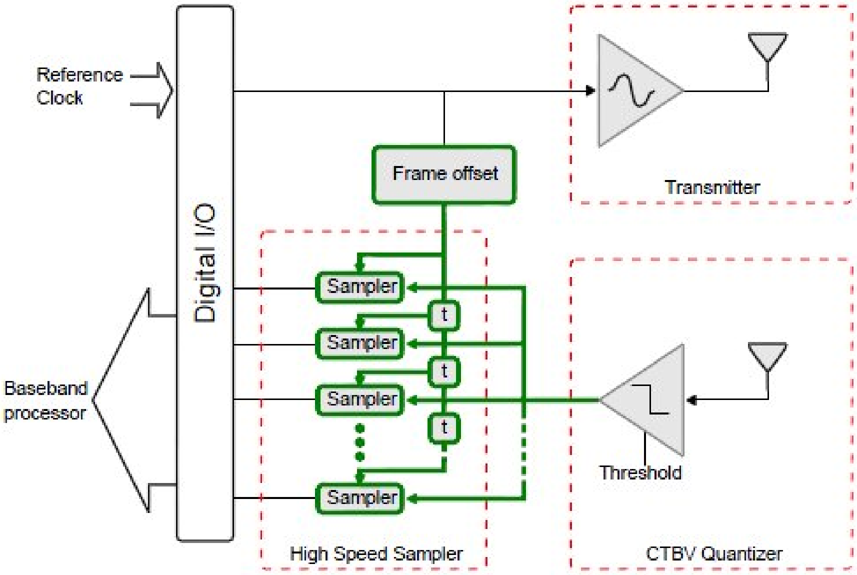


Figure 2.3: Impulse Radar based on CTBV method [16].

These digital value are referred to as samplers and the collection of the samplers are read from the radar chip is called Frame. NVA R-641 used in this project, received digital signal consists of 512 individual samplers.

The data stored in the frame represent the voltage or strength of the radar received signal. It is assumed that the input voltage range of the receiver is between 0 V to 1.04 V. These values are normalized by Pre-Amplifier gain that is shown in *Figure* 1.1. The signal voltage is normalized between 0 to 100 (i.e.100 means 1.04 V, 0 means 0 V and 50 means 0.502 V etc). Then these normalized values are stored in frame. *Figure 2.4* shows an example of the normalized received signal from NVA R-641. This received signal has 512 samplers.

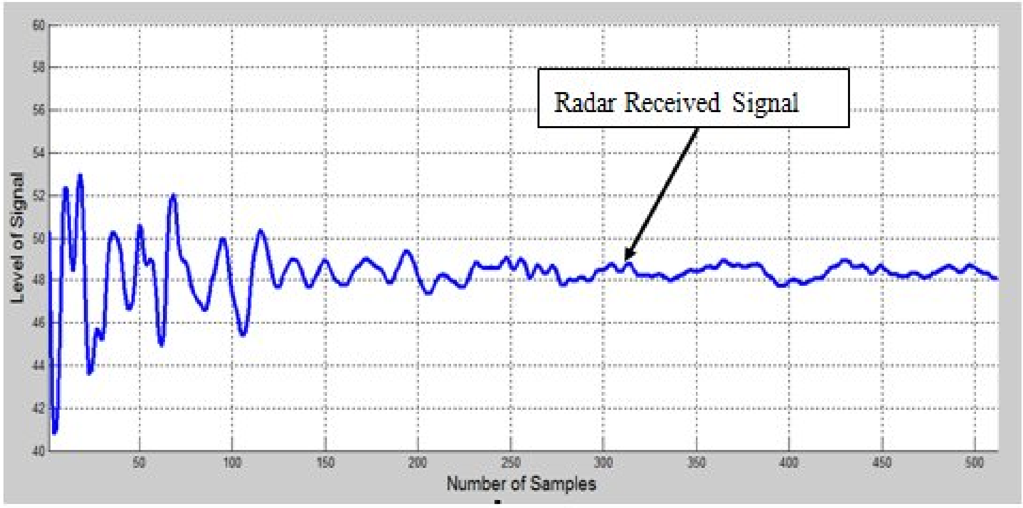


Figure 2.4: Normalized recieved signal with 512 samplers.

2.4 Basic Functionality of Novelda Impulse Radar

One of the basic concepts of the Novelda Impulse Radar is time duration takes the radar signal is stored in a frame. This time duration is related to the spatial domain in UWB Impulse Radar.

The conversion from time to space in radar gives us the ability to detect the distance of the object from radar and tracking object. Figure 2.5 shows how reflections from two objects are stored in the frame. As can be seen object1 is close to the Impulse Radar appears in the begining of the frame and object 2 is located further away from the radar apears in further part of the frame.

The recorded frame can be buffered at different distances relating to the Impulse Radar by setting the frame offset parameter. Frame offset parameter is a distance that radar can sense the objects that are located outside the frame offset range.

Figure 2.5 shows the recorded frame is located in a distance that can cover the reflection from object 1 and object 2. Any reflections coming from objects outside the range of the recorded frame are ignored by the impulse radar. For example if object1 is located in a distance between the impulse radar and recorded frame (in frame offset duration) the reflected signal from object1 is ignored by the Impulse Radar. In other words frame offset can be considered as a blind distance that t radar cannot sense the objects are located in this distance.

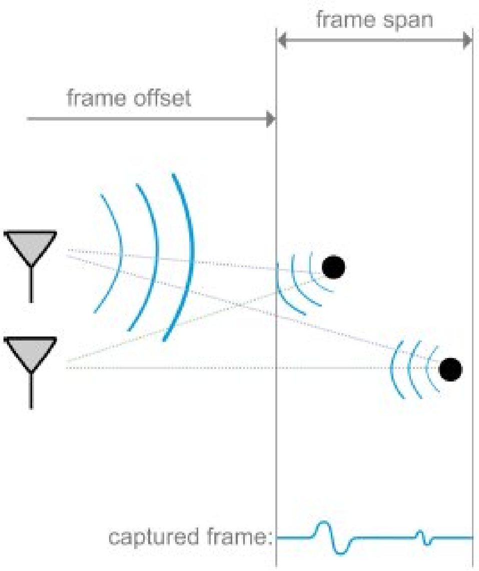


Figure 2.5: Relationship between times and space in UWB radar[16]

As previously described each frame has 512 samplers. The time delay between every two consecutive sampler is defined by (2.1). Normal values for Samples per Second is about 30 – 37 GHz depending on Pulse Repetition Frequency (PRF), temperature etc. Consequently in this case the time between every sampler is approximately about 0.027 ns.

$$Time\ between\ samplers\ =\ 1/SamplesPerSecond = 1/37 = 0.027ns \qquad (2.1)$$

By converting this time to spatial domain, 0.027 ns delay represents approximately 4.2 mm. therefore, in each frame the time duration of the signal is $512 * 0.027ns\ =\ 13.82\ ns$ and approximately 2.2 m in spatial domain.

Frame offset parameter in this project is set to 80 samples. In space, it is approximately 34 cm. Accordingly, in spatial domain one received signal spans the range from 0.34 m to0.34 $+ 2.2\ m$.

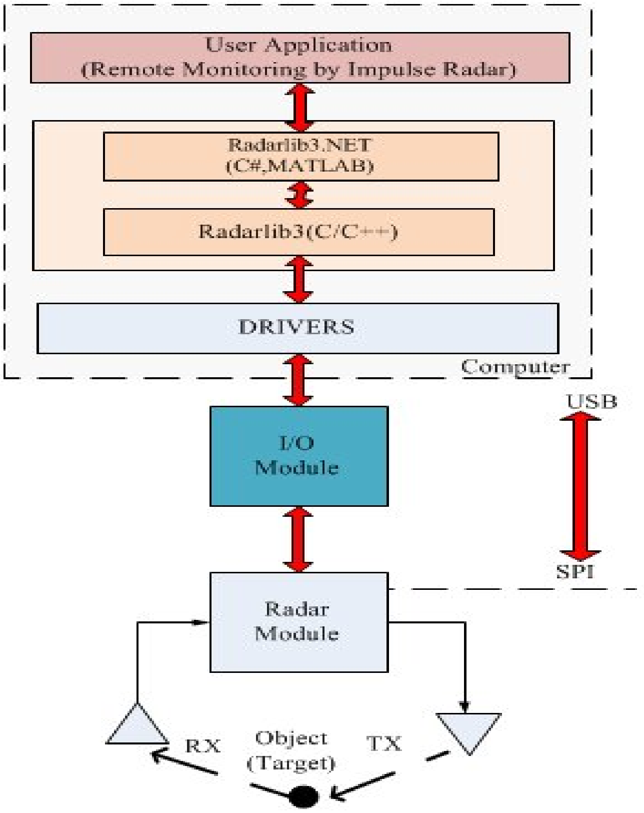
2.5 Development Kit Architecture

Novelda NVA-R641 development kit is built up of both hardware and software modules in a layered architecture. The hardware part of the NVA-R641 development kit consists of three main parts as illustrated in **Error! Reference source not found.**.

- Two Vivaldi antennas
- Impulse Radar chip
- I/O-module (USB to SPI adapter)

The software part of the NVA-R641 development kit also consists of three main parts.

The first software layer consists of a driver which is located at the bottom of the software stack and makes the I/O-module visible and accessible for the operating system on the computer. The middle software layer is called Radarlib3/Radarlib3.Net and handles communication between the I/O-module and the user application. On top of Radarlib3/Rdarlib3.Net the user application will receive, process and present the data which are received from the radar module.



2.6: Connection of Impulse Radar to computer.

User application is a tool for building demo application of the Impulse Radar. For example in this project, "Remote Monitoring by Impulse Radar" is a software application which is written in MATLAB software in user application. If software application is written in C/C++, it uses the functions are defined in Radarlib3 for running the program and if is written in MATLAB or C# Rdarlib3.Net is used.

The basic MATLAB code for connecting the Impulse Radar to computer is listed in appendix 8.1. Figure 2.7 shows the connection of the Impulse Radar to the computer while the programm is running in MATLAB.

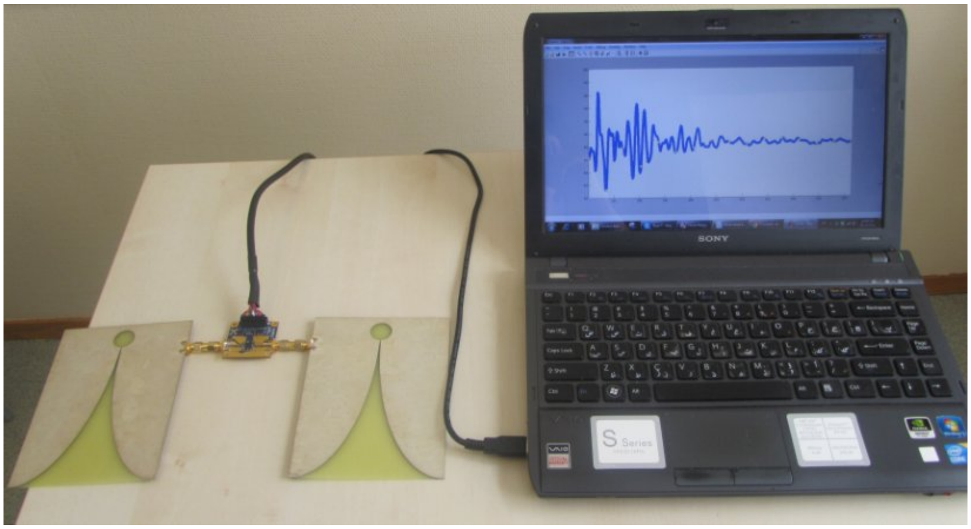


Figure 2.7: connection of Impulse Radar to the computer.

3 Filtering

In signal processing, filtering is an algorithm that removes unwanted components or features from signal. Each reflected signal from the Impulse Radar NVA-R641 consists of 512 samples per frame. Each sample has the components of clutter and noise other than the object's signal. Removing these unwanted parts is one of the most important processes which is done in order to remove the noise and clutter from the received signal and retaining useful data. In this chapter DC removal, and singular value decomposition (SVD) as useful methods for filtering the clutter and noise are used.

3.1 Signal Model

Figure 3.1 shows one example of the received signal from Impulse Radar. This signal consists of three components of the reflections from object, clutter and noise [6].

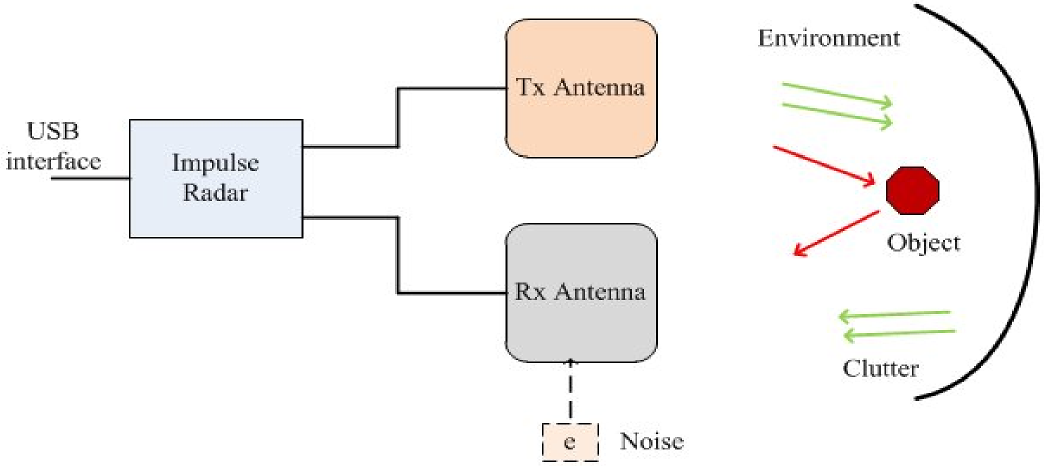


Figure 3.1: Radar system and the components of the received signal

Clutter: Clutter is defined the reflections from the surrounding environment of the object in addition the desired object. In other words, the received signal from radar is the summation of the desired object and the environment. For indoor application like our case clutter is reflections from the things around the object such as chairs or walls. Therefore, we need to eliminate theses unwanted signals from the radar received signal in order to retrieve the real signal for the next signal processing [17].

Noise: Noise is defined as the random part of the radar received signal and we need to eliminate noise from the radar signal as much as possible [17].

Each received signal v consists of three parts:

$$v = v_c + v_t + e\,. \tag{3.1}$$

Therefore the received signal v and its subcomponents of v_c , v_t and e can be expressed in a 512 dimensional vectors.
Figure 3.2 shows one example of received signal v from the radar such that each sampler is superposition of reflected signal from object, clutter and noise.

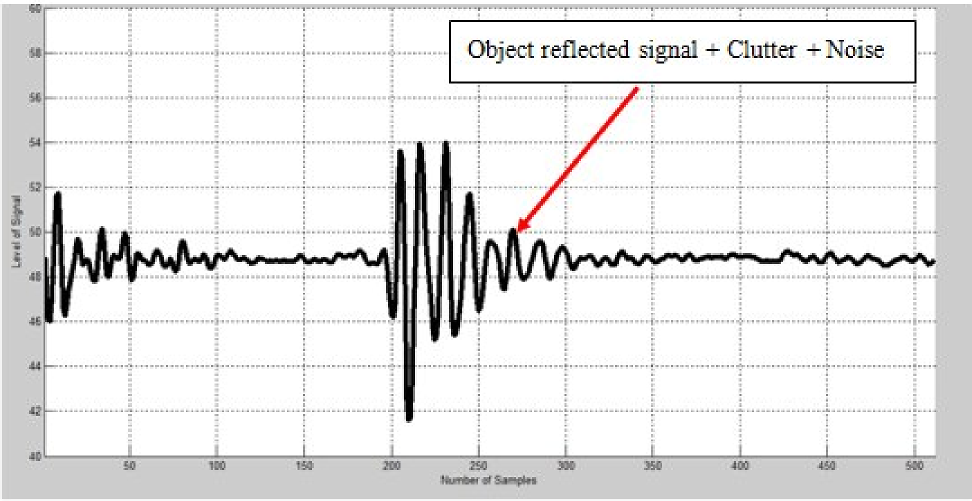


Figure 3.2: Radar received signal v with its subcomponent

The object of interest in this figure is a mirror **500* 500 mm2** area. The surface of the mirror provides a strong reflection in comparison of other reflected signal from the environment.
The reflected signal at each sample number is summation of the reflected energy from both the mirror and the surrounding environment.
Where the vectors v^1,v^2, ……, v^n denote the consecutively received signals in a time duration of n and so the measurement matrix in this specific time can be written as matrix A:

$$\mathrm{A} = [\, \mathrm{v}^1, \mathrm{v}^2, \mathrm{v}^3, \ldots, \mathrm{v}^n]. \tag{3.2}$$

Where the measurement matrix A can be expanded and written as:

$$A = A_c + A_t + E = [v_c^1, v_c^2, ..., v_c^n] + [v_t^1, v_t^2, ..., v_t^n] + [e^1, e^2, ..., e^n]. \tag{3.3}$$

Where $\mathrm{A_c}$, $\mathrm{A_t}$, and E are referred to as the clutter matrix, the object matrix, and the noise matrix, respectively. Figure 3.3 shows measurement matrix A with size of (N, 512), where N denotes to the time of measurement.

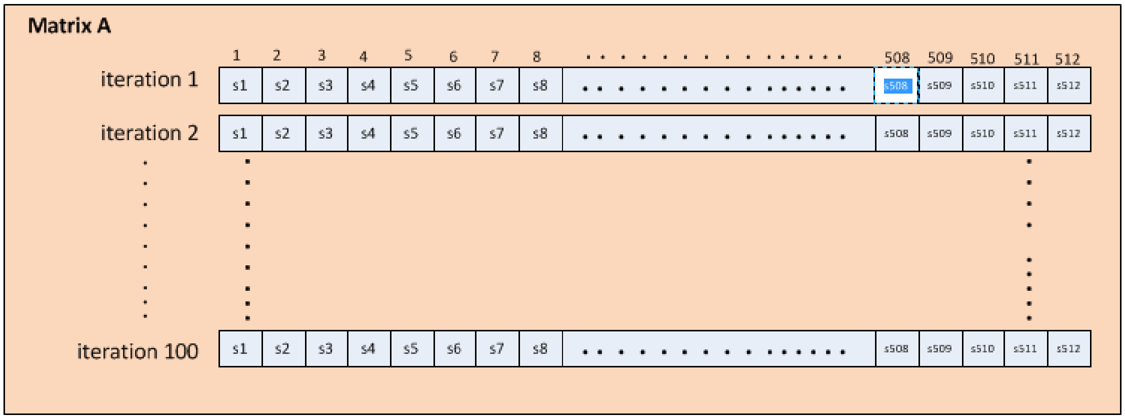


Figure 3.3:Measurement matrix A.

3.2 Clutter Matrix

The clutter parameter v_c can be modeled by a linear equation of with a rescaling factor α_i and a shifted bias c_i

$$v_c^i = a^i v_c + c^i, \tag{3.4}$$

so the clutter matrix can be written as:

$$A_c = [v_c^1, v_c^2, ..., v_c^n]. \tag{3.5}$$

The clutter matrix $A_c$can be expanded as:

$$A_c = [a^1 v_c + c^1, a^2 v_c + c^2, a^3 v_c + c^3, ..., a^n v_c + c^n] = [a^1 v_c, a^2 v_c, a^3 v_c, ..., a^n v_c] + [c^1, c^2, c^3, ..., c^n] = V_c^i + C^i, \tag{3.6}$$

where a^i is the scalar and v_c represents the clutter vector, and c^{i} is a vector of ones scaled by a scalar for each measurement i. That is $\mathrm{c}^{\mathrm{i}} = [1\ 1\ , \ldots ,1]\mathrm{c}^{\mathrm{i}}$ and c^{i} is a constant number.

Besides, form the regarded model it can be said that the clutter matrix is a rank two matrix since the columns of the matrices both V_c^i and C^i are linearly dependent [6].

3.3 Noise Matrix

Noise is defined as a random part of the reflected signal. Noise is modeled as zero-mean random variable and it can be represented by a matrix which has random direction and low power.

3.4 Singular Value Decomposition (SVD)

Due the property of clutter's definition in equations (3.5) and (3.6), method of Singular Value Decomposition can be used as a suitable method for removing the clutter. Singular Value Decomposition is a mathematical method that works according to matrix factorization methods [18].

The measurement matrix $\mathrm{A} \in \mathrm{R}^{\mathrm{m} \times \mathrm{n}}$ is a full rank matrix that can be factorized by Singular Value Decomposition rule. The Singular Value Decomposition is given by:

$$A = UWV^T, \tag{3.7}$$

where $U \in R^{m \times m}$ and $V \in R^{n \times n}$ are orthogonal matrices, and $W \in R^{m \times n}$ is a matrix whose off-diagonal entries are all zeros and whose diagonal elements satisfy:

$$\begin{array}{ccc} \mathrm{W}_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mathrm{W}_\mathrm{n} \\ 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{array} \tag{3.8}$$

$$W_1 > W_2 > \cdots > W_n > 0, \tag{3.9}$$

W_i is unique and called the singular values of matrix A. The column matrix of U (also called the left singular matrix of A) is eigenvectors of the matrix $A\,A^T$. The columns of V (also called the right singular vectors of A) are eigenvectors of the matrix $A\,A^T$.

As we described before in formula (3.3) the measurement matrix A is combination of reflections from the targets, the clutter and the noise. The clutter is made of two rank-1 matrices that we assume that these two rank-1 matrices have the strongest weight in the measurement matrix A. Using the compact form of singular value decomposition of A matrix is a useful strategy in to estimate the clutter in this project. To show how the Singular Value Decomposition compression is used to removing the clutter part, the measurement matrix of $\mathrm{A = UWV^T}$can be written as formula (3.11).

$$A = \partial_1 U_1 V_1^T + \partial_2 U_2 V_2^T + \ldots..+\partial_n U_n V_n^T. \tag{3.11}$$

According to formula (3.9) the singular values of the measurement matrix A are stored in descending order. Also we have assumed that clutter has the strongest weights in the measurement matrix A and the rank of clutter is equal 2. Consequently the first two element of the Singular Value Decomposition of matrix A in formula (3.11) represents the clutter that will be removed. So the clutter can be written as:

$$Clutter = \partial_1 U_1 V_1^T + \partial_2 U_2 V_2^T,. \tag{3.12}$$

Therefore, the measurement matrix A after removing the clutter shown like (3.13):

$$A - clutter = \partial_3 U_3 V_3^T + \partial_4 U_4 V_4^T + \ldots..+\partial_n U_n V_n^T. \tag{3.13}$$

In order to estimate the clutter matrix, we take measurement in our place without any object. Figure *3.4* shows the room that the object is not put. So the measurement matrix is consists of only clutter and noise:

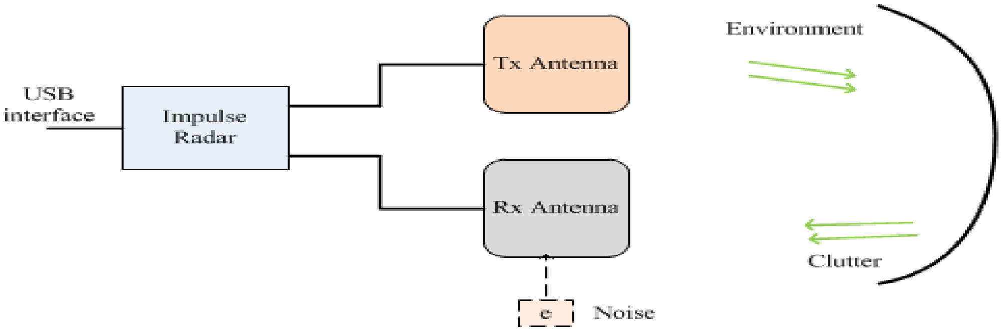


Figure 3.4*: Room without object.*

So measurement matrix A can be written as:

$$A = clutter + noise. \tag{3.14}$$

By using method of Singular Value Decomposition of measurement matrix A and removing the first two elements of the singular values as a clutter and remaining part is approximation of noise. Figure *3.5* and Figure *3.6* are general view of the clutter and noise by taking measurement in a place without desirable object.

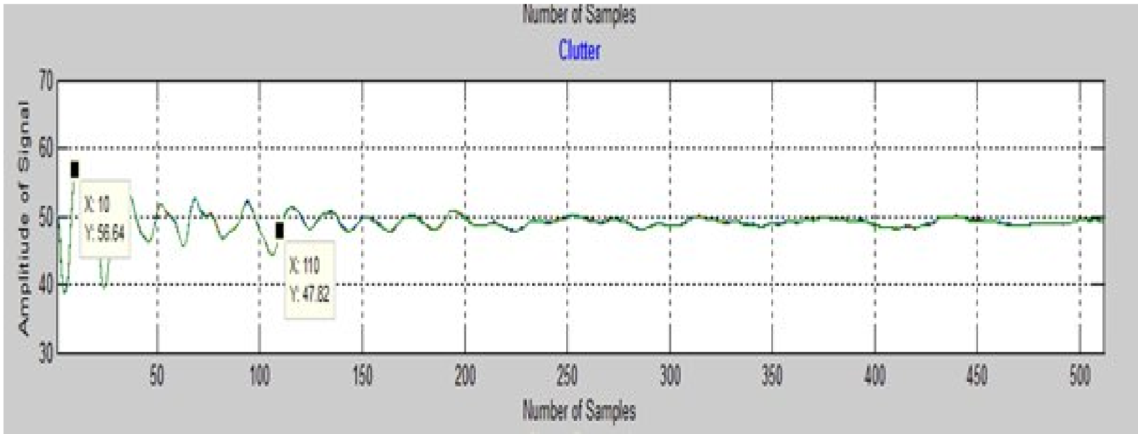


Figure 3.5*: Clutter estimation in a room without any object.*

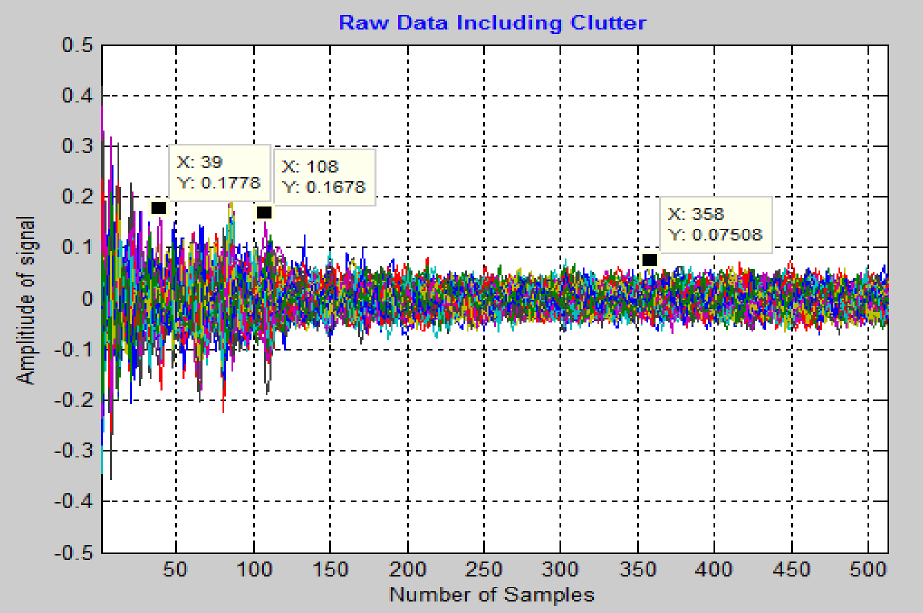


Figure *3.6: Noise estimation in a room without any object.*

Comparison between these two figures shows that estimated clutter have very higher amplitude in comparison with estimated noise. Therefore it tried to remove and reject clutter in radar signal

A general view of the measurement matrix (raw data including clutter), clutter and removing clutter from the raw data is shown in Figure 3.7. The measurement matrix consists of 100 consecutive measurements ($n = 100$) of a moving mirror.

Since clutter has the strongest weights in the measurement matrix, clutter seems as a thick line in the raw data. Besides, Subtraction of clutter from raw data results the reflected signal from moving mirror. Sample number 10 and 110 are examples that show removing clutter from the raw data has decreased the amplitude of the radar received signal.

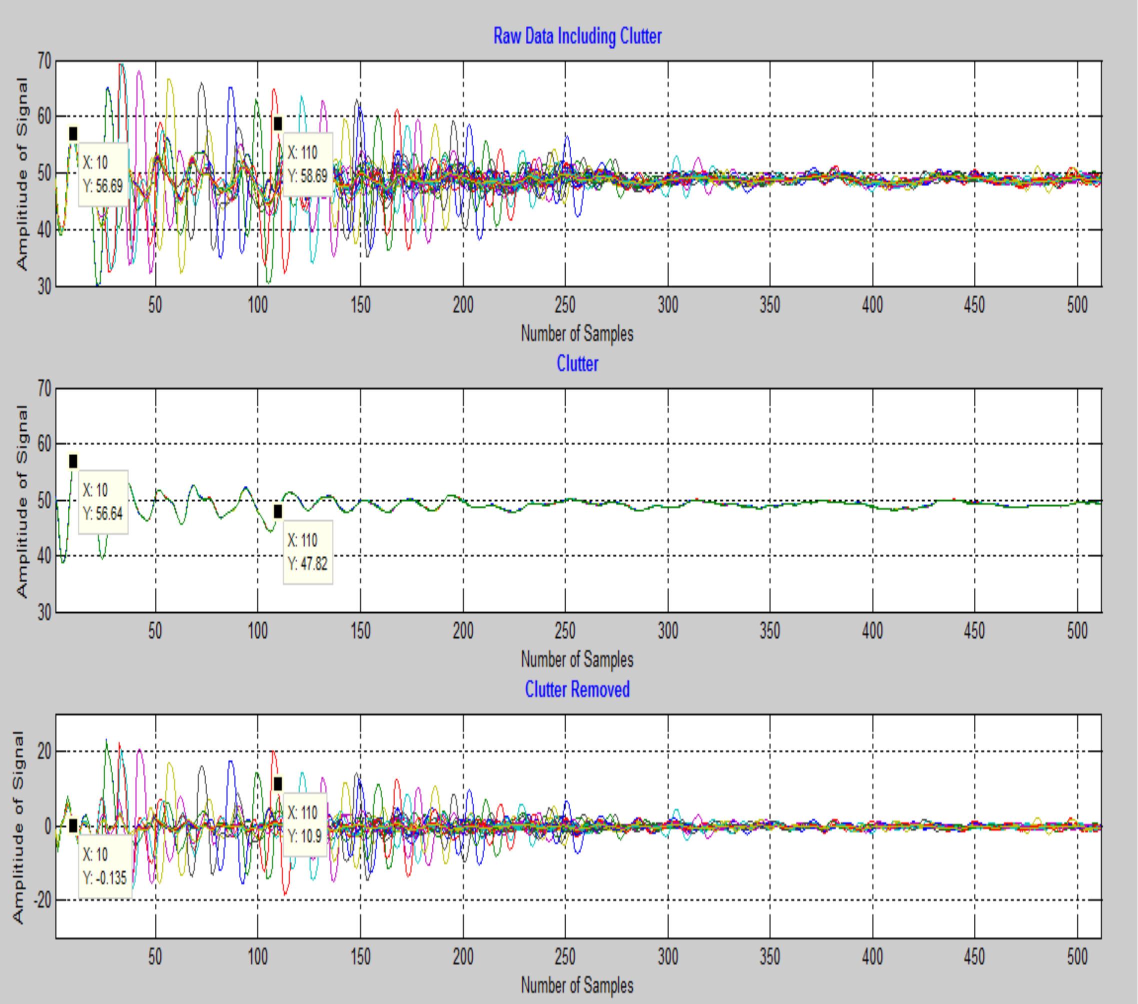


Figure 3.7: Raw signal vs. Clutter removed signal.

3.5 DC Removal Filter

DC offset is an offsetting of a signal from zero. It means that the waveform in default view appears not to be centered on 0 around horizontal lines. This offset can be seen in Figure 3.7.

The removal of the offset from a signal is a common requirement in signal processing [19]. This filter is applied by finding the mean value of all sample points in measurement matrix A and then subtracting it from all sample points. Figure 3.8 is a general view of the measurement matrix (raw data including clutter), clutter and removing clutter from the raw data after removing DC offset. Figure 3.8 shows that radar signal amplitude has decreased significantly after removing DC offset. Corresponding MATLAB code for filtering with SVD and DC removal are in Appendix 8.2.

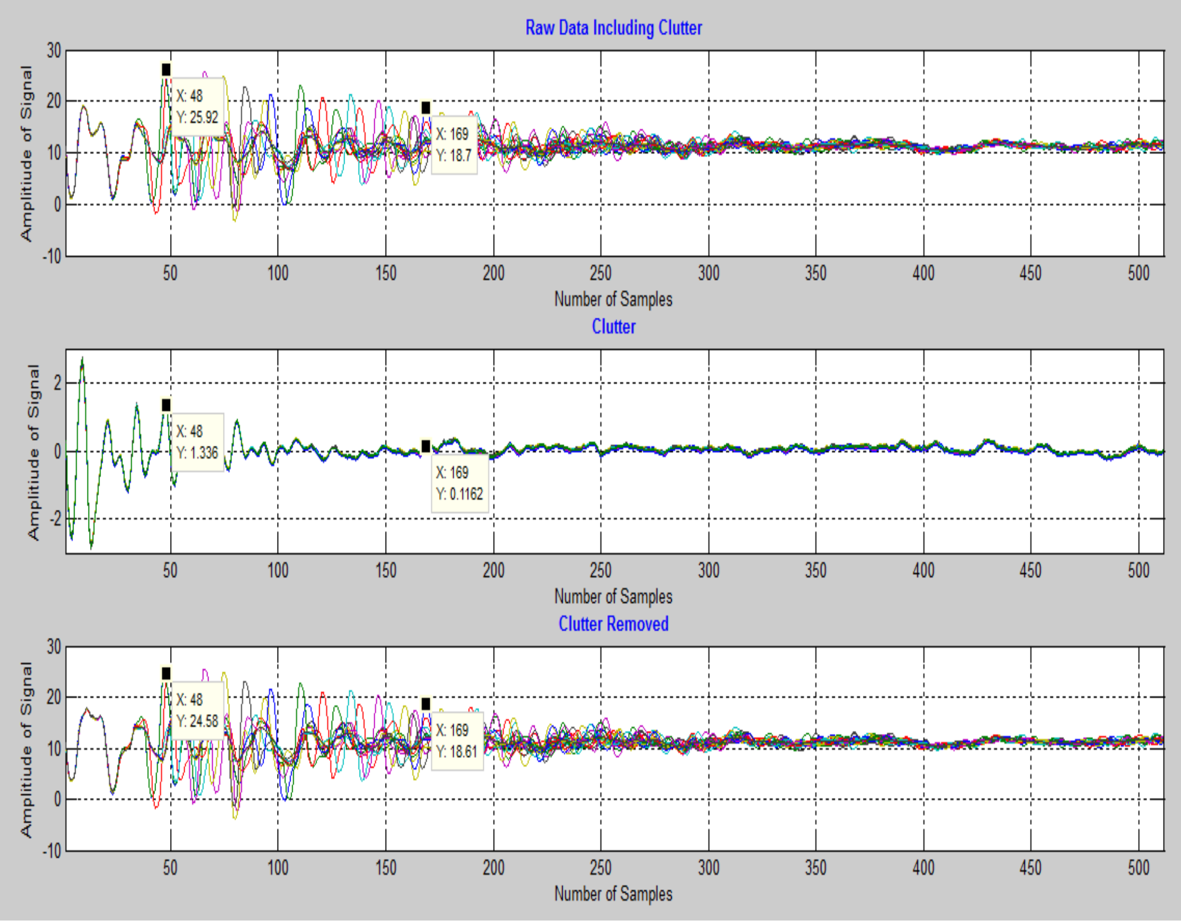


Figure 3.8: Raw signal vs. Clutter removed signal after removing DC offset

4 Detecting Static and Moving Objects

As previously described in chapter 1 and chapter 2 radars is an object detection system which uses electromagnetic waves to detect the presence of objects, determine the distance and or speed of objects. Therefore one of the important usages of NVA R-641 is detecting the object that is located in the radar field and identifying the distance of object from radar. The objective of this chapter is providing an algorithm to detect static and moving objects.

UWB signals have the characteristic that the signal amplitude decreases as distance increases. Therefore, in order to determine the distance of object from radar, threshold method is used. Threshold method has been developed empirically for our case to determine the relative position of an object according to signal amplitude.

4.1 Radar Range Equation

Radar range equation helps to determine the distance of the object from radar according to amplitude of reflected signal. Strength of radar received signal is a function of distance from radar and other parameters. Equation (4.1) defines the relationship between transmitted power $\mathrm{P_t}$ and received power $\mathrm{P_r}$ for far-field application [20].

$$P_r = P_t\, G_t\; A_{eff} \frac{1}{(4\pi d)^2} \tag{4.1}$$

Where:

G_t = Gain of the radar transmit antenna (dimensionless)

d = Distance from radar to the object (meter)

A_{eff} = Effective area of the radar receiving antenna (meters squared)

P_t = Power transmitted by the radar (watts)

$\mathrm{P_r}$ = Power received back from the target by the radar (watts)

The radar equation (4.1) provides a relationship between the received power, the characteristics of the target, and characteristics of the radar itself. It also shows that the received power decreases by the square of distance from radar to the object.

Due the following factor determining radar signal amplitude over distance in near field is very difficult to do properly. The major factor is how the signal is distributed in the area. If the antenna is not directional, the signal will be sent in all directions and the signal strength will be reduced as any signal with spherical spreading. Also the surface of the object may vary and there will be different reflections [21]. So it is a bit difficult to calculate the strength of the signal over distance. Therefore, for our project we use the far field formulas and determine an equation for near field case empirically.

In order to find signal amplitude over distance empirically, the radar is put in a fixed point and the mirror is applied in line-of-sight of the radar. Then the signal strength of the radar is measured in 10 points i.e. 10 points between samples from 1 to 512. These measurements results are presented in Table 4.1. Figure 4.1 shows the location of mirror in front of radar in 10 different points.

Table 4. 1: Experimental data.

Position	51	65	126	160	194	233	288	331	383	495
Amplitude	63.56	61.08	59	56.68	55.49	53.86	52.49	52.26	52.25	51.29

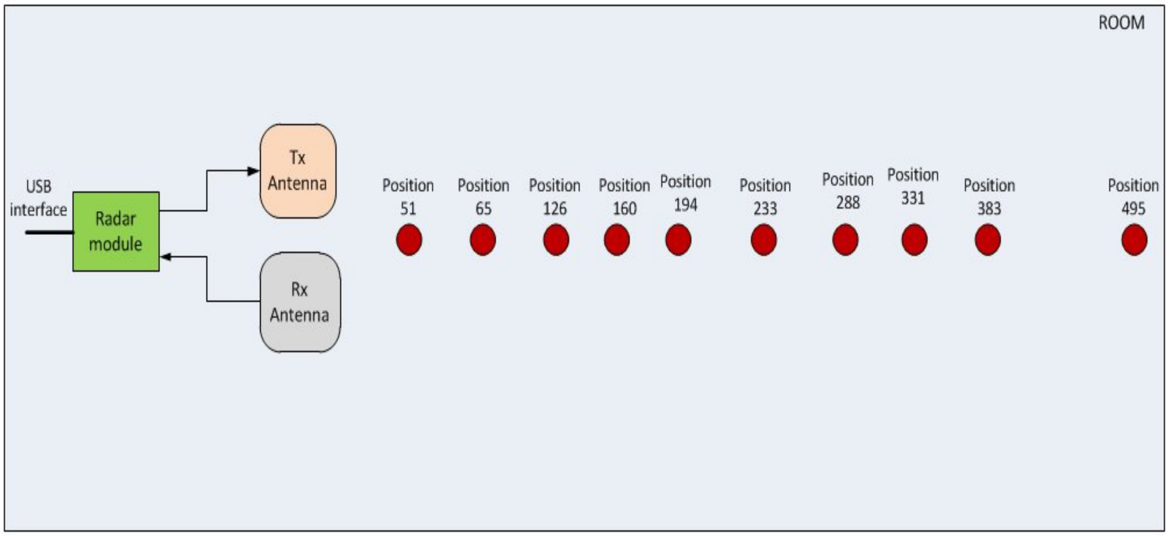


Figure 4.1: Object in front of the radar in different locations.

4.1.1Curve Fitting

Curve fitting is the process of constructing a curve, or mathematical function that has the best fit to a series of data points, possibly subject to constraints. It is desirable to represent the experimental data in *Table 4. 1* with a closest curve fitting expression. We use $y = a + \frac{b}{x}$ to approximate the given set of data in *Table 4. 1*. We want to fit a line $y = a + \frac{b}{x}$ for the known points (x_i,y_i), where i = 1, …, 10.

We suppose z =$^1/_x$. Therefore we can write $y = ax + z$ and we have 10 equations as below: These set of equation can be written as the following matrix equation.

$$\begin{bmatrix} 1 & \frac{1}{51} \\ 1 & \frac{1}{65} \\ 1 & \frac{1}{126} \\ 1 & \frac{1}{160} \\ 1 & \frac{1}{194} \\ 1 & \frac{1}{233} \\ 1 & \frac{1}{288} \\ 1 & \frac{1}{331} \\ 1 & \frac{1}{383} \\ 41 & \frac{1}{495} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 63.56 \\ 61.08 \\ 59 \\ 56.68 \\ 55.49 \\ 53.86 \\ 52.49 \\ 52.26 \\ 52.25 \\ 21.29 \end{bmatrix} .$$

Since we are trying to find a single line that passes through 10 different points, In general, we will write our matrix equation as $\boldsymbol{AC} = \boldsymbol{Y}$. where:

$$\begin{bmatrix} 1 & \frac{1}{d_1} \\ 1 & \frac{1}{d_2} \\ \vdots & \vdots \\ 1 & \frac{1}{d_n} \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} .$$

And we have

$$A=\begin{bmatrix}1 & \frac{1}{d_1}\\ 1 & \frac{1}{d_2}\\ \vdots & \vdots\\ 1 & \frac{1}{d_n}\end{bmatrix}\qquad C=\begin{bmatrix}\alpha\\ b\end{bmatrix}\qquad Y=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix}$$

As d_i denotes sample number and y_i is radar signal amplitude. $\mathbf{AC} = \mathbf{y}$ equation can be written as :

$(A^TA)C = A^TY.$

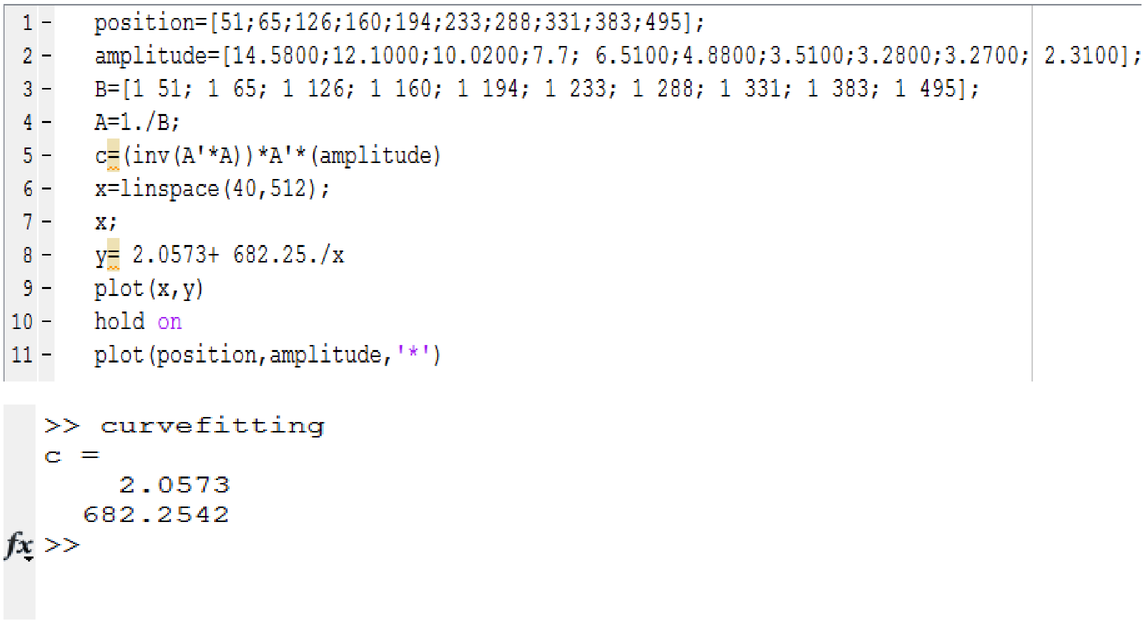
Where:

$C = (A^TA)^{-1}A^TY`.$

Solving this equation in MATLAB gives C = $\begin{bmatrix}2.057\\ 682.25\end{bmatrix}$. As a result the equation of regression line is:

$$y = 2.057 + \frac{685.25}{x} \tag{4.2}$$

x = linspace (40,512) generates 472 points linearly spaced between and including samples 40 and 512. Figure *4.2* shows some part of this curve fitting equation in MATLAB.



```
position=[51;65;126;160;194;233;288;331;383;495];
amplitude=[14.5800;12.1000;10.0200;7.7; 6.5100;4.8800;3.5100;3.2800;3.2700; 2.3100];
B=[1 51; 1 65; 1 126; 1 160; 1 194; 1 233; 1 288; 1 331; 1 383; 1 495];
A=1./B;
c=(inv(A'*A))*A'*(amplitude)
x=linspace(40,512);
x;
y= 2.0573+ 682.25./x
plot(x,y)
hold on
plot(position,amplitude,'*')
```

```
>> curvefitting
c =
    2.0573
  682.2542
>>
```

Figure *4.2: MATLAB code curve fitting*

In Figure 4.3 dots indicate the maximum strength of the measured signal and the solid line fits these measured values. As it is seen from the blue fitted curve for the first values of radar sample numbers, the signal amplitude approches to infinity. So the first 40 sample' value is replaced with the value of sample 41 which is equal to 14.58. The curve shows that the radar signal strength is reduced as distance increases. Therefore, the distance to the object can be roughly calculated from $y = 2.057 + \frac{685.25}{x}$ where y is signal strength and x denotes distance form radar.

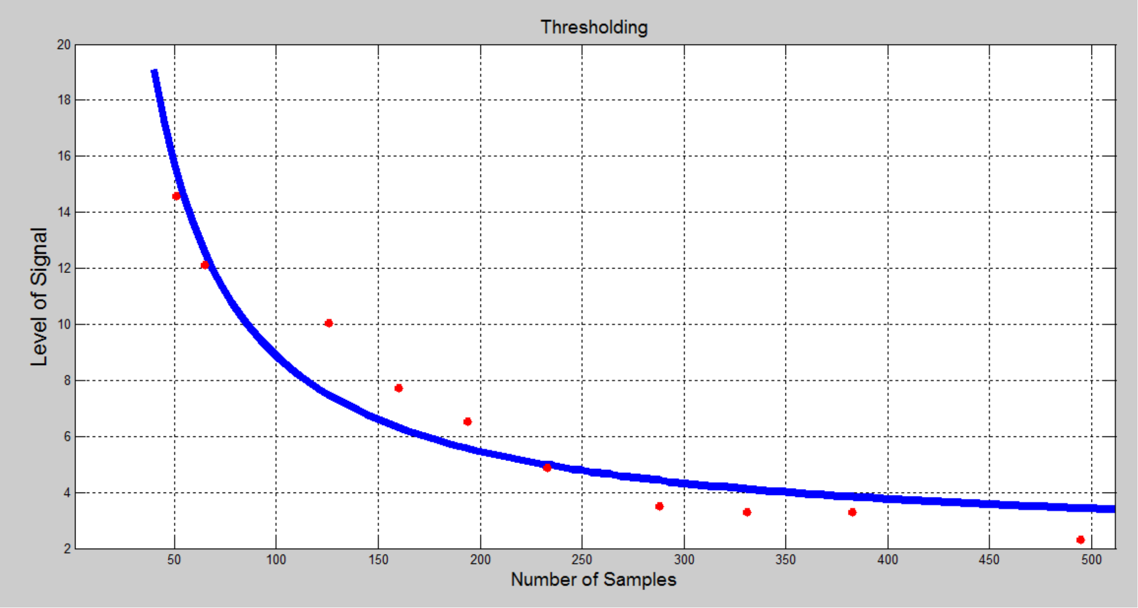


Figure 4.3: The relation between signal strength and distance.

4.2 Object Dtection

Figure 4.4 and Figure 4.5 show two streams of data from the Impuse Radar. **Error! Reference source not found.** shows the situation, in which there is no object in front of the radar. Figure 4.5 shows the case, where a mirror is placed in fornt of the radar. It shows that the reflected signal from the mirror is stronger than other signals reflected from the sourounding environment. Therefore, the presence of an objcet in fornt of the radar around sample 200 -250 is observed.

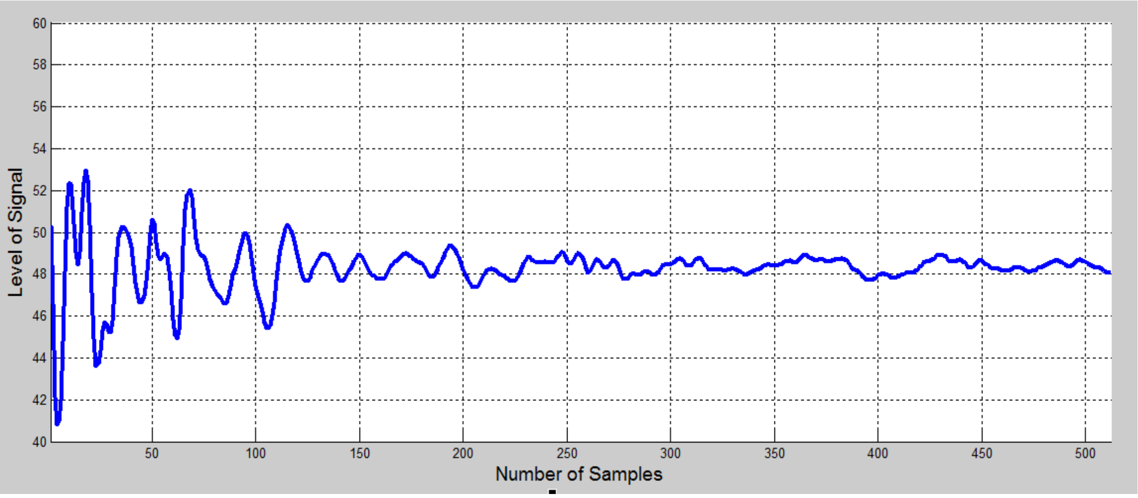


Figure 4.4: Radar signal when no object is in front of radar.

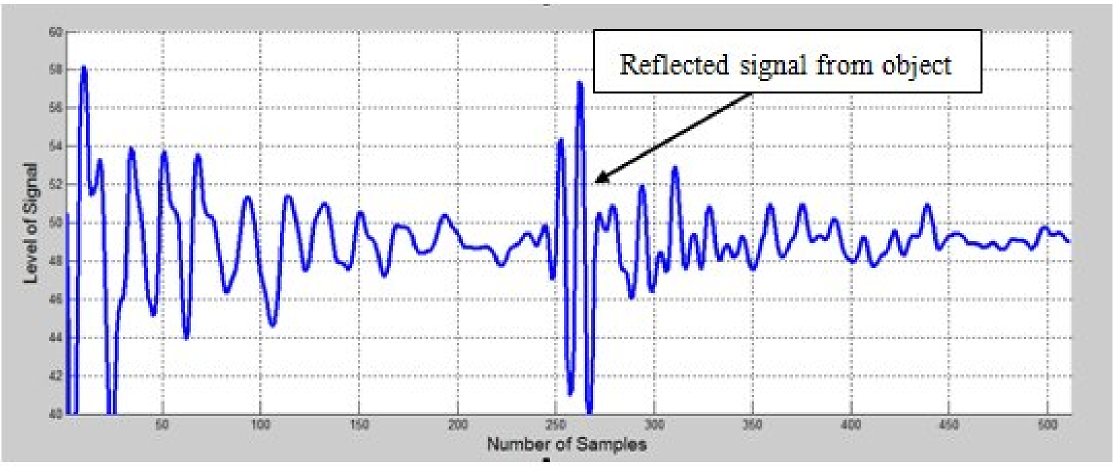


Figure 4.5: Radar signal when an object is placed around sample 200-250.

The result is that object detection is one important part in radar signal processing. Consequently in our signal processing algorithms we need to implement an algorithm to determine the presence and location of object. The solution of the target detection task is based on threshold detection algorithm.

4.2.1Threshold Detection Algorithm

Object detection is the process of evaluating the radar signal and determining whether a target is detected by radar. Radar received signal can be processed and analyzed by threshold detection algorithm. If the received signal exceeds the threshold, presence of the target is presented. Similarly, radar signal may fail to exceed the threshold, and thus there is no object in radar field. These are illustrated in

Figure 4.6 and

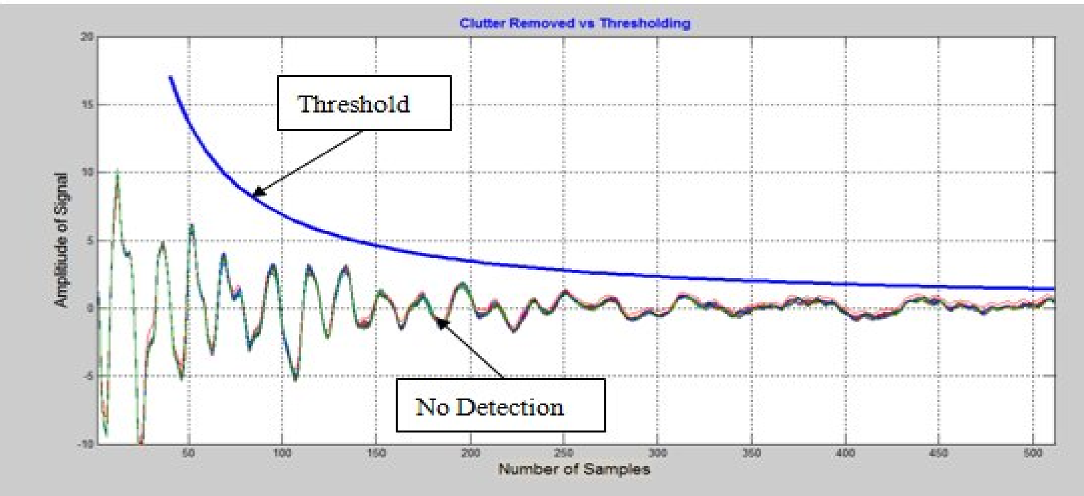


Figure 4.7. Figure 4.6 shows the radar signal is greater than the threshold around sample 250 - 280. So it is an indication of presence of an object. In contrast, in

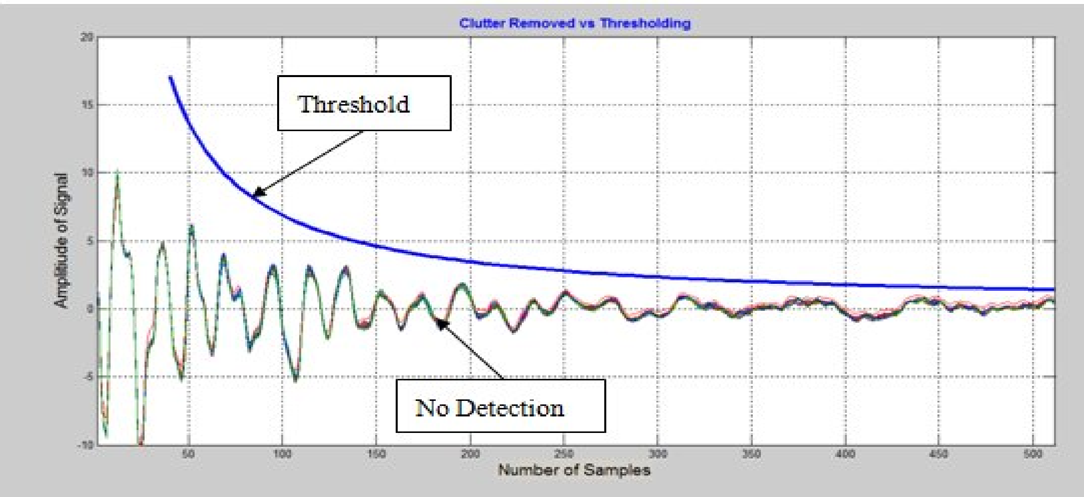


Figure 4.7 radar signal has crossed threshold in no sample point and it shows there is no object in front of radar.

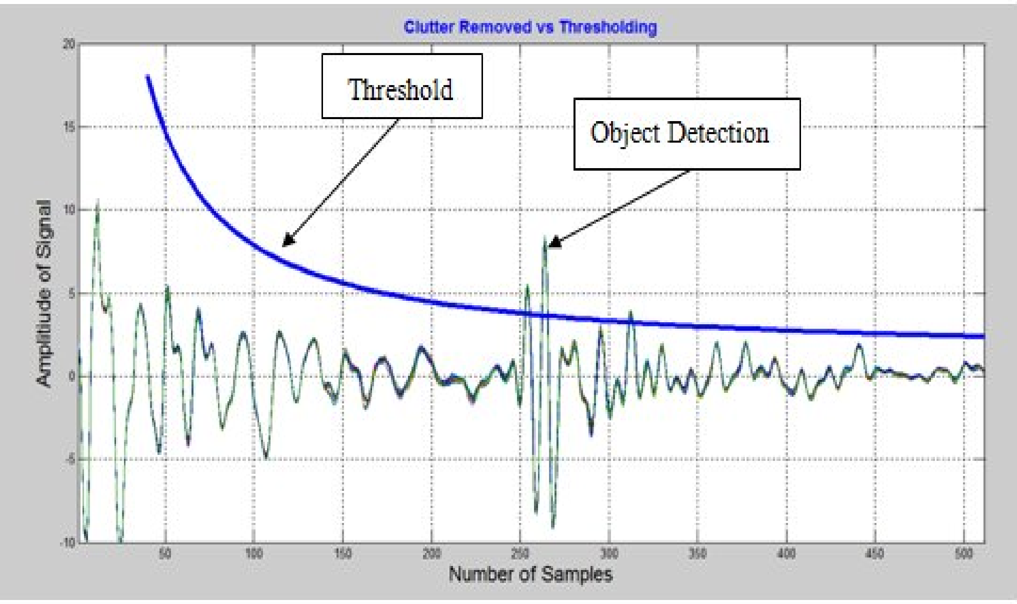


Figure 4.6: Radar signal vs threshold with presence of object**.**

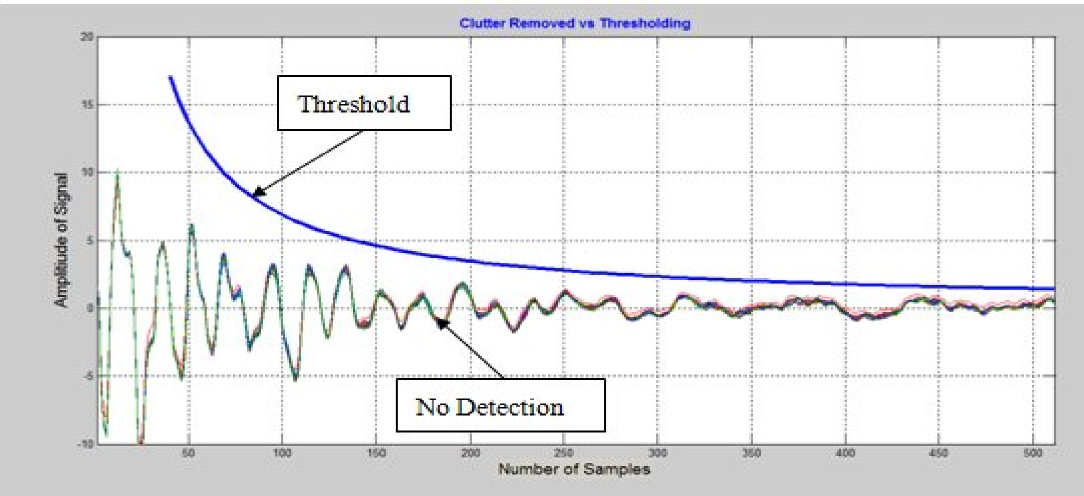


Figure 4.7: Radar signal vs threshold.

Figure 4.8 shows a general view of measurement matrix A $(100, 512)$. The number of rows in matrix A is proportional to the time of measurement. Target $(100, 512)$ is also an empty matrix with the same size of matrix A which is filled by data processing algorithm on matrix A.

Data processing algorithm is executed for each row of matrix A which is equivalent to one radar received signal. Data processing algorithm is a comparator which compare radar received signal in each sample with its equivalent threshold. When the radar signal goes over threshold, it yields 1 and it remains 0, until radar signal is below of threshold. The result for each sample is recorded in the Target matrix with the same index. Therefore, each cell in Target matrix has high value declares the presence of an object, such that the index of that cell equal to the location of the object.

As radar signal is affected by thermal noise, various interferences, the probability of object detection is affected by these random variables. I.e. noise signal will occasionally exceed the threshold, producing what is called a false alarm. Similarly, a target signal may fail to exceed the threshold, and thus not be detected. Therefore in order to decrease these random effects, and have precise detection about presence of an object and its location, we can check each sample along with neighbor samples. For this reason we use Sweep pointer. Size of sweep can be set optionally. It is supposed Sweep is 5. Sweep not only checks each sample but also checks the 5 samples that are located both in the left and right part of the desirable sample. Then the numbers of samples are more than threshold values in this duration (10 checked samples) are recorded in the Target matrix in the desirable cell. This procedure is carried out for each 512 samples in each row. Finding high values in each row in Target matrix help to find the presence and location of the object in radar received signal.

Another important point that should be declared about threshold algorithm is curve fitting equation. Equation (4.2) was derived for maximum value at each sample number. Therefore, we need to subtract from this curve fitting equation to detect the presence of object. In this ace we have subtracted about 10% from curve fitting equation.

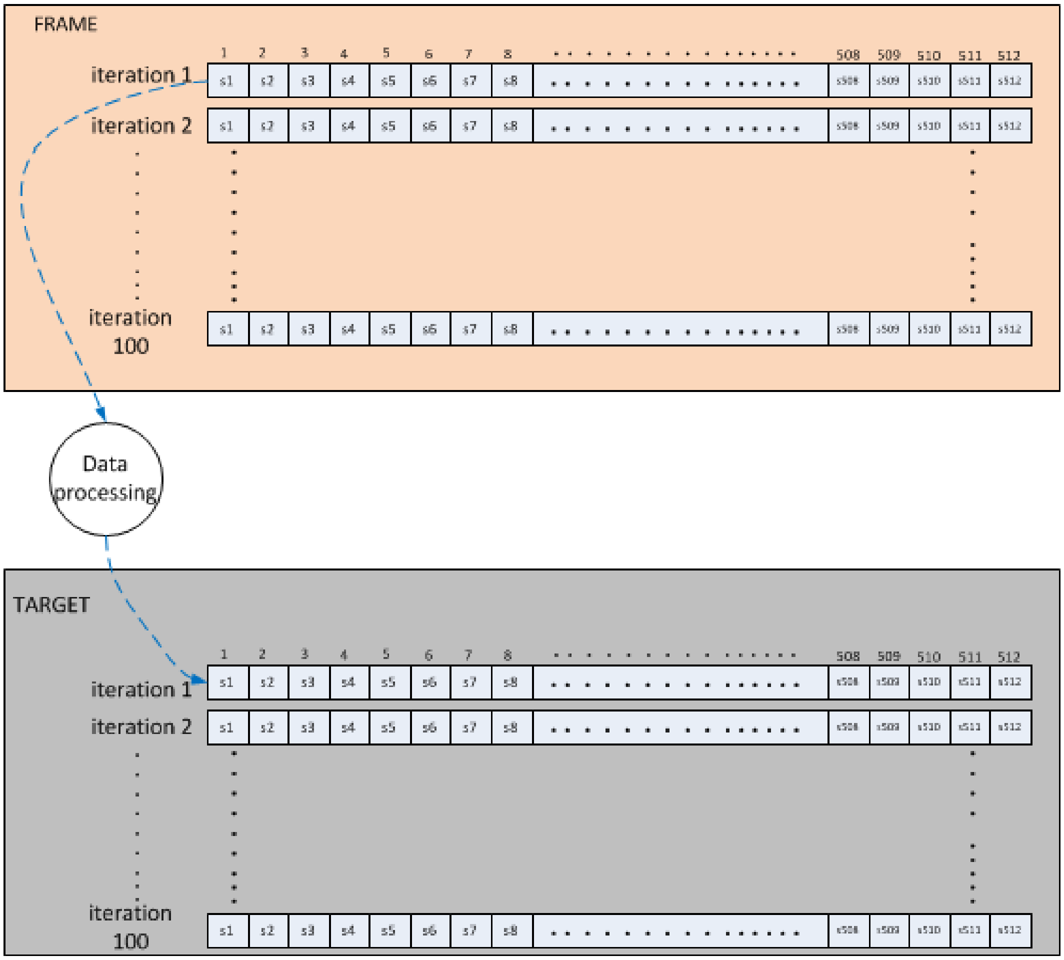


Figure 4.8: Data processing algorithm for detecting object.

4.2.2Detecting Static Object

Figure 4.9 illustrates refrigerator is located as a static object in front of the radar. Measurement by meter stick shows it is located in 75 cm of the radar. Figure *4.10* shows radar signal amplitude over distance of this measurement. It is seen from this figure radar signal has the highest level amplitude in sample number 180 to 200 in comparison with other samples. Therefore, it means a static object located in front of radar in sample number from 180 to 200.

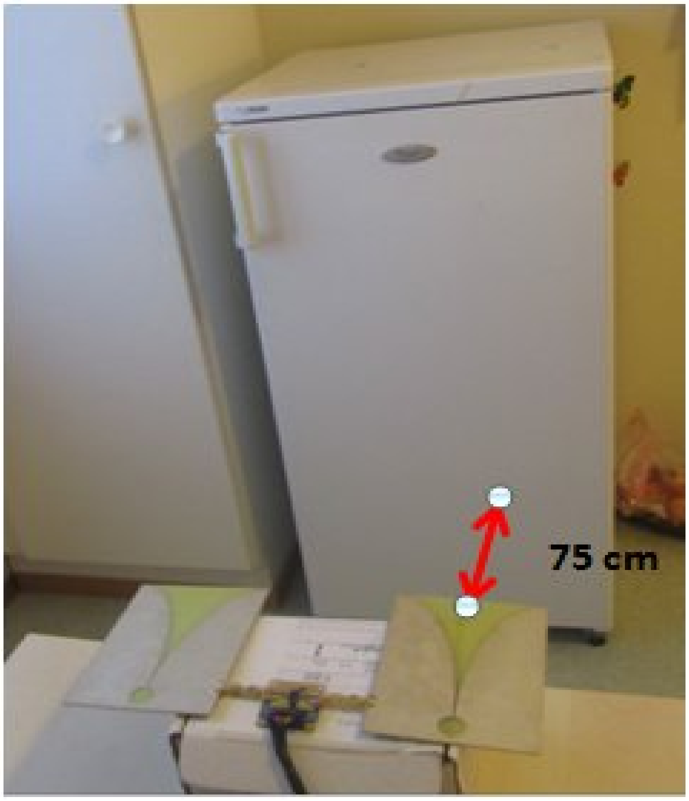


Figure 4.9: Static object in front of the radar.

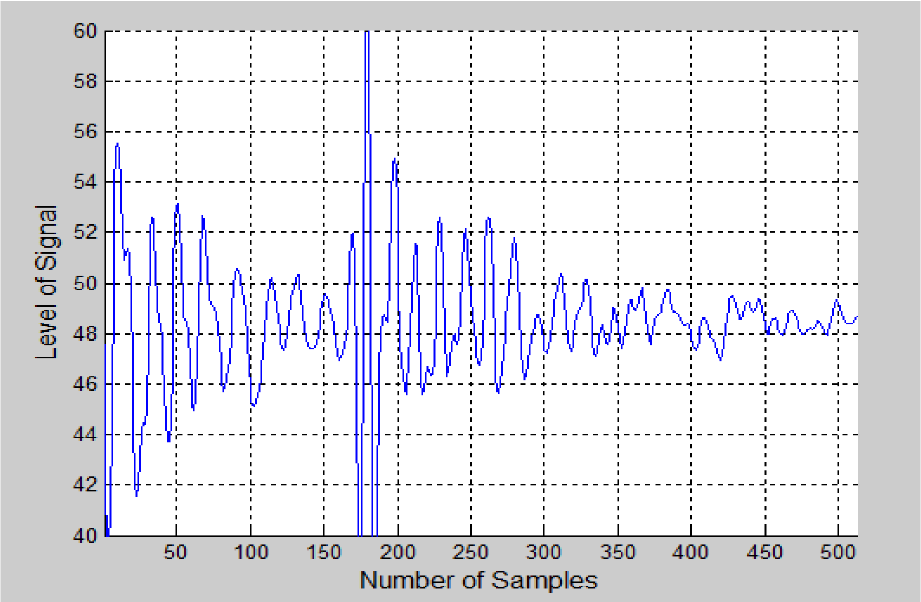


Figure 4.10: appearance of object around sample 180-200.

Radar signals are stored in measurement matrix A. Results from data processing on matrix A recorded in Target matrix. Interpretation of Target matrix helps to determine about the type of object (static and moving) and approximate distance of refrigerator from radar.

Figure *4.11* is whole overview of Target matrix. Target Matrix in Figure *4.11* indicates that radar received signal in each sample is less than threshold value except sample from 150 to 180. This result is repeated for every row of measurement matrix A. As results, it is understood there static object in front of radar in sample number of 180-200. Distance calculation of the refrigerator from the radar is written as below:

$$Distance = \frac{(180 * 2.1m)}{512} = 0.79\ m$$

Comparison shows there is small difference between measurement by meter stick and measuring distance by radar.

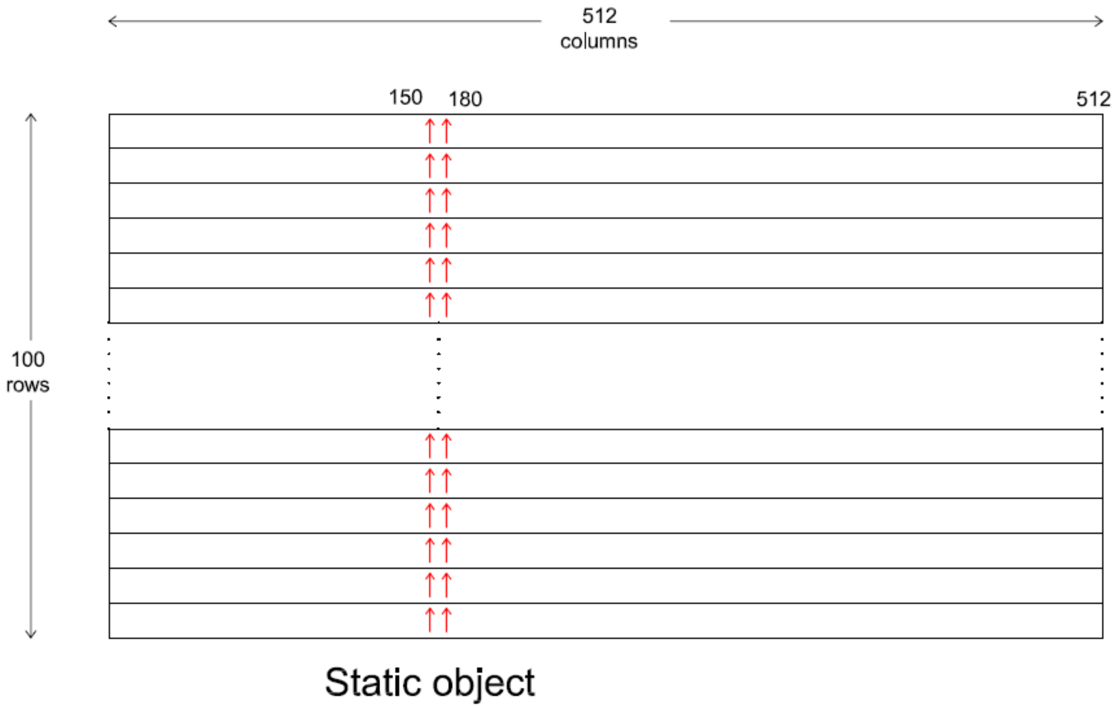


Figure 4.11: Static object detection.

Figure *4.12* shows presence probability of static object for one rwo in Target matrix. It shows in the range of 180 – 200 probabilty of presence of static object is 4. However this probabiltiy for other samples are zero.

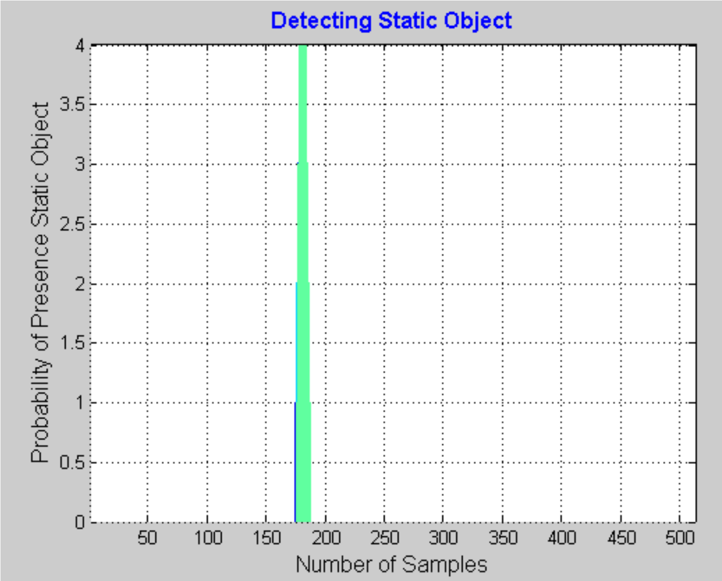


Figure *4.12: Presence probability of static object*

4.2.3Detecting Moving Object

In the presence of a moving object, analyzing becomes more complicated. In contrast to static object, measurement matrix A has different stored values in each row. So after data processing on measurement matrix A, each row in target matrix has different values too.

Figure *4.13* shows 100 consecutive measurements of a moving mirror in front of the radar. The mirror is moved from close to further distance of the radar slowly. As can be seen radar signal has the highest amplitude in the first samples and it is decreasing by taking distance from the radar.

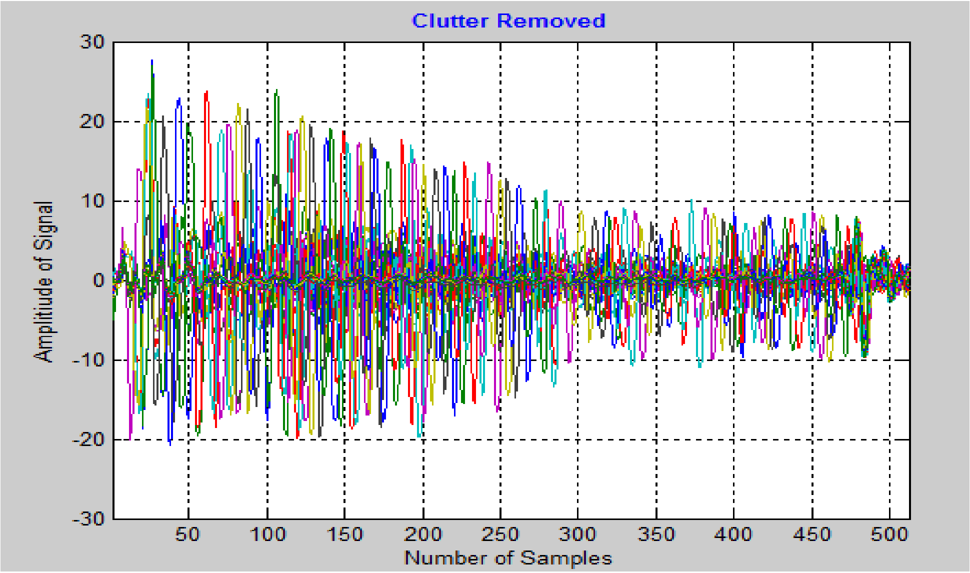


Figure 4.13: 100 consecutive measurements of moving metal object

Target matrix result is shown inFigure *4.14* . This matrix shows that every row has its own maximum value. These maximum values are shifting from sample number 1 to sample 512 during the measurement. These results indicate there is a moving object which moves from sample number 1 to 512. These sample number are recorded in MAX vector respectively.
MAX vector has 100 cells that each cell belongs to one sample number of each row in Target matrix. By taking all the maximum values of every row, we have an approximation of all the sample numbers that mirror has passed during the moving time.

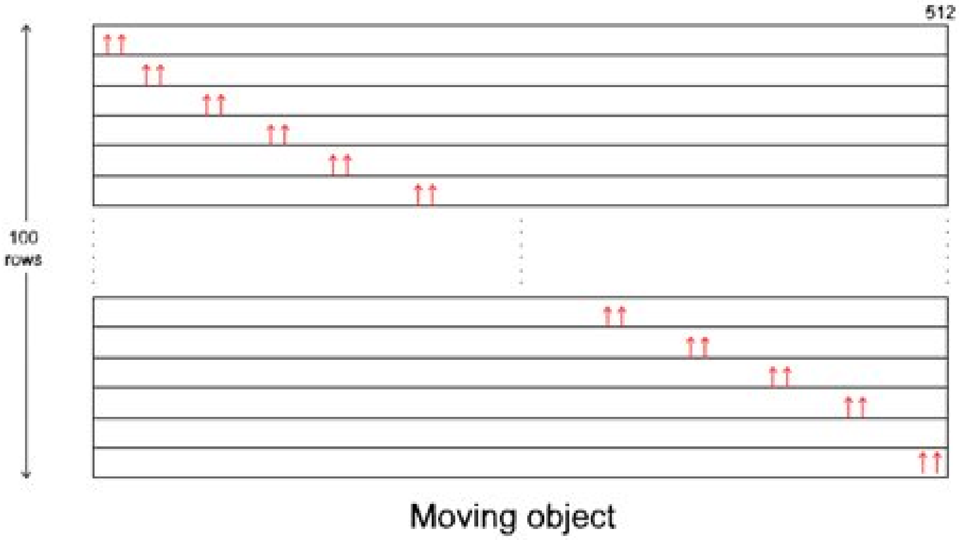


Figure *4.14: Detection of moving objec.*

In order to determine distance of the moving object from radar, we convert sample numbers in MAX vector to distance. Converting this sample numbers to distance has resulted Figure 4.15. Figure 4.15 shows that the moving object has started to move from very close to the radar and it continues until about 2.2 m. Also it shows that the object does not have movement in the last times of measurement, since the distance is constant.

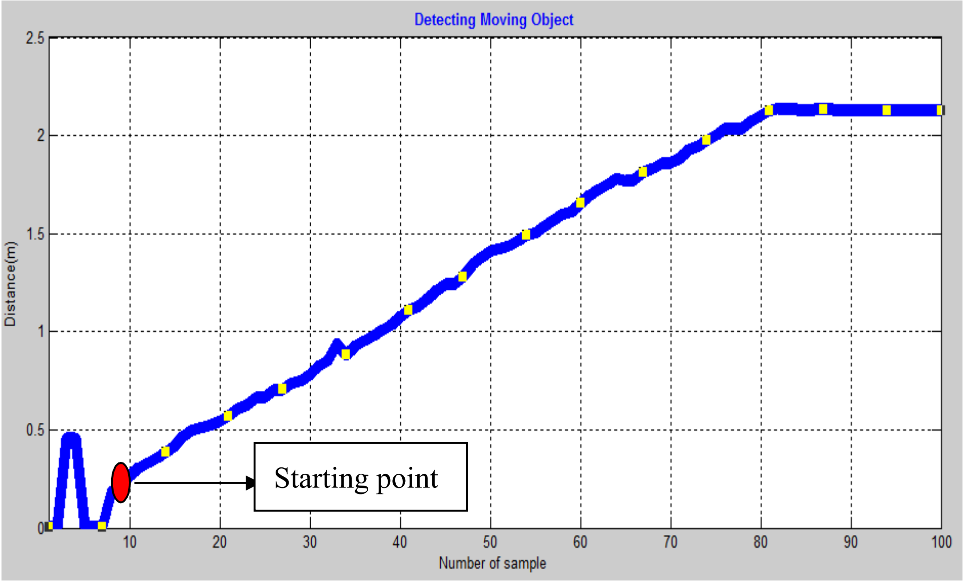


Figure 4.15: Distances of the moving object from radar

4.2.4Detecting Static and Moving Object

Figure 4.16 shows 100 consecutive measurements of one moving mirror and one static mirror. This figure illustrates that there are some movements from first samples of radar until about 200 and also there is high signal amplitude in around sample number 310 to 330.

Result from Target matrix illustrated inFigure *4.17* **.** Results from this figure shows there is a static object around sample 310-340 and a moving abject 1 and 150.

If we follow the same strategy in for moving chapter in the last section, we will see that there is a fixed maximum value in sample 315 that is repeated in each row. So this fixed maximum values shows there is a static object in this area. In addition, we see the maximum value in each row for the first 150 samples has been changed. Therefore it is clear that there is a moving object in this area. Converting the these sample number to distance results that the static object is located in 1.4 m and the moving object moves from in the area before about before the static object. Distance of moving and static object from radar is shown Figure 4.18

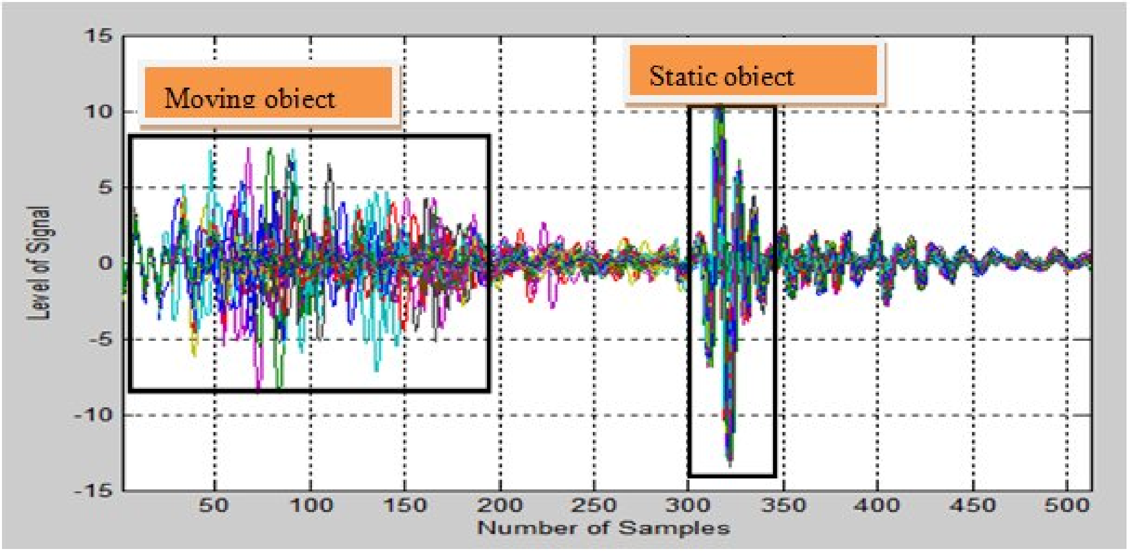


Figure 4.16:100 consecutive measurements of one moving mirror and one static object.

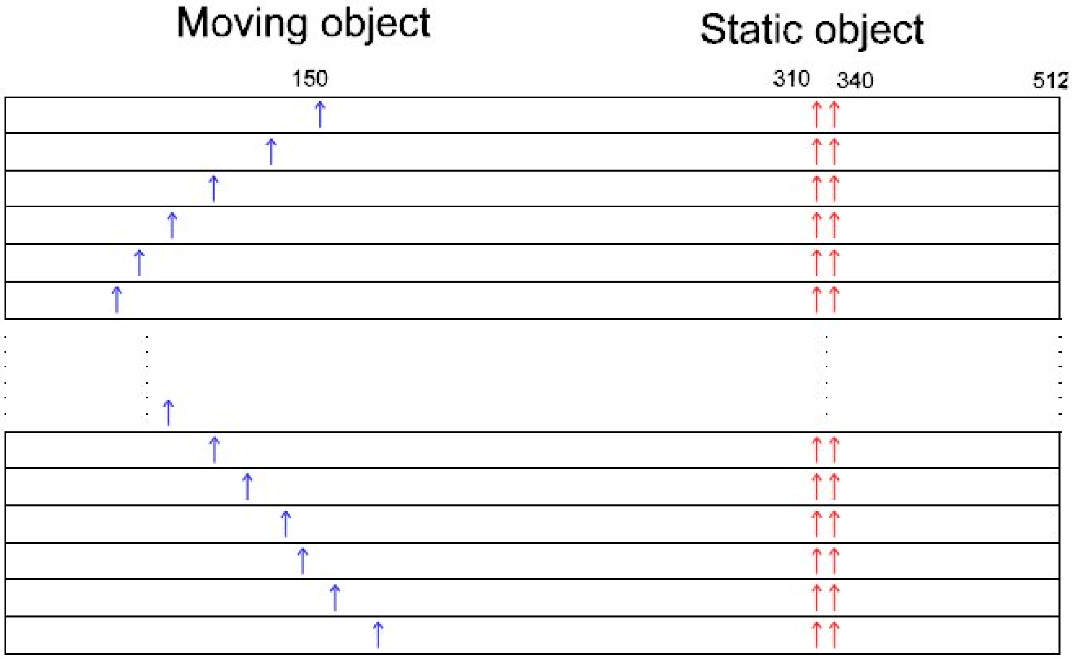


Figure *4.17: Detection the locations of moving object.*

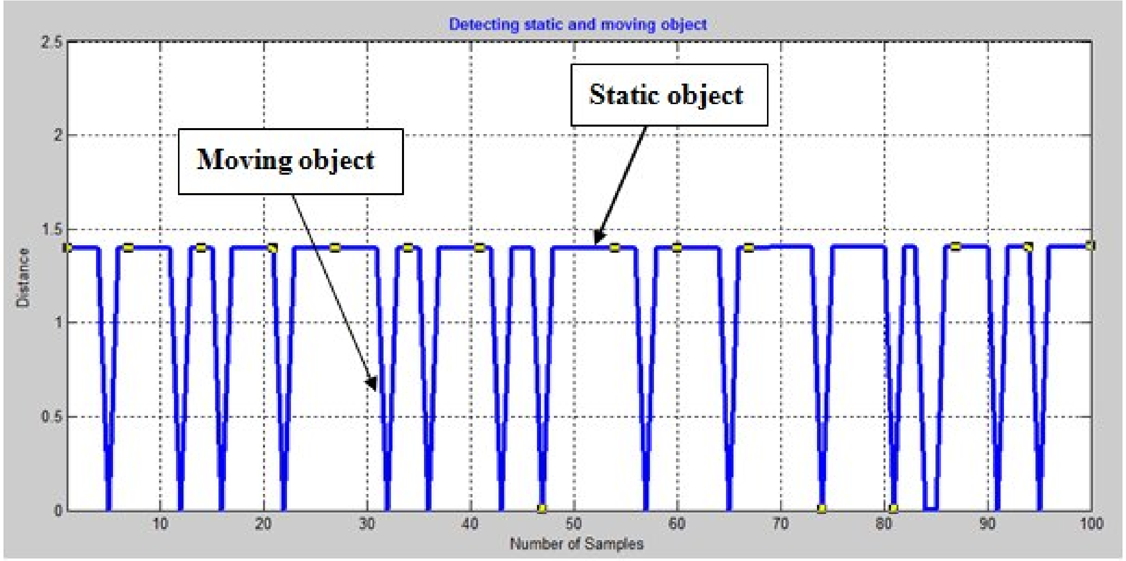


Figure 4.18: distance of static and moving object from radar.

Some basic MATLAB code related to static and moving object detection is presented in Appendix 8.3.

5 Radar Tracking Using Kalman Filter

Measuring and predicting the correct position and velocity of a moving object is demanding in radar application. The approach in this chapter is one- dimensional radar tracking by Kalman filter. It is according that using an initial estimated position and velocity of the object and then using the Kalman filter. Kalman filter equations are applied to reduce the error in position and velocity and predicting the correct position and velocity [22]. In this chapter, first we will discuss Kalman Filter equations and algorithms and then the algorithm will be used for tracking the object in one dimension.

5.1 Kalman Filter for One Dimensional Tracking

In one dimensional tracking, the distance of the object to radar and velocity of the object is estimated by using Kalman filter.

5.2 Initialization

The position and velocity of the moving object is described by the linear state space of (5.1).

$$X_k = \begin{pmatrix} X_k \\ \dot{X}_k \end{pmatrix} \tag{5.1}$$

$\dot{\mathrm{X}}_{\mathrm{k}}$ is the velocity of the moving object which is the derivative of position of the object with respect to time. The state equation (4.2) describes the relation between the state and the next state of the motion object that each X_{k+1} is a linear combination of its previous value plus a process noise and the radar measurement equation is presented in (4.3) is a linear combination of the signal value and the measurement noise.

$$X_{k+1} = A\, x_k + W_{k+1} \quad W_k \sim W_k = N(0, Q) \tag{5.2}$$

$$Z_K = H\, x_k + v_k \quad V_K \sim V_K = N(0, R) \tag{5.3}$$

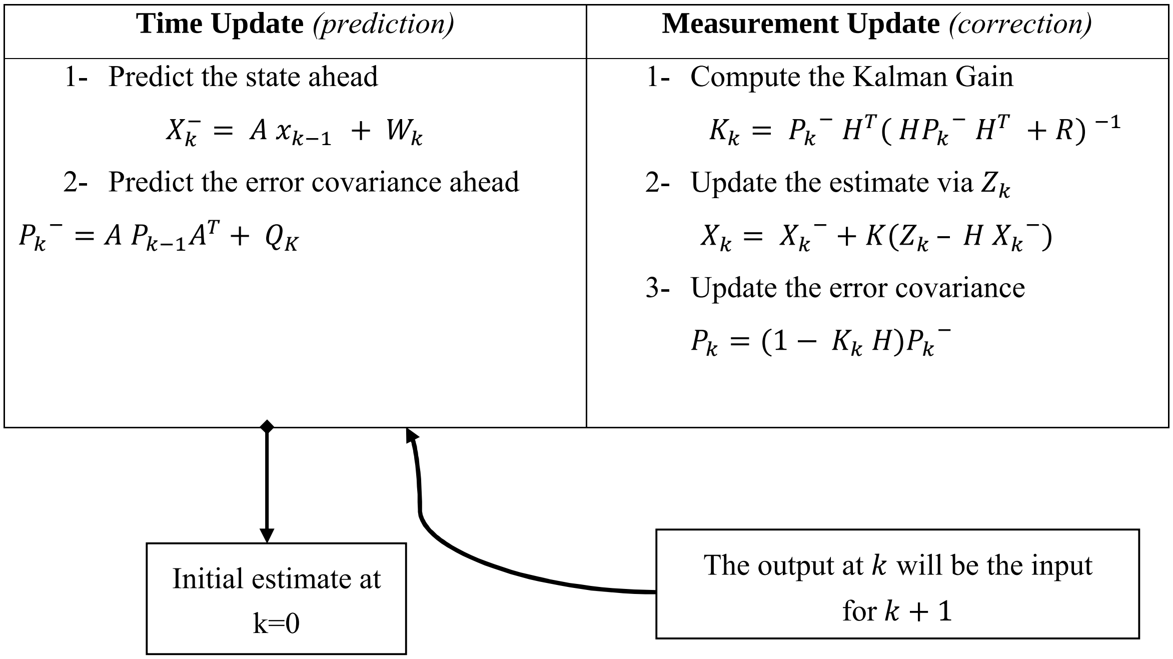
Where $W_t = N(0, Q)$ is defined Process noise and assumed to be zero mean Gaussian white noise with covariance Q.

And $V_K = N(0, R)$ is defined Observation noise and assumed to be zero mean Gaussian white noise with covariance R.

Where $A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$ is the Transfer function with step size Δt and H is the measurement matrix which represents the relation between the true state xk and measurement Y_K and C is initialized as [1 0].

Kalman Filter Equations

The Kalman filter can be described in two distinct set of phases: **Time Update** (prediction) and **Measurement Update** (correction). Both equation sets are applied at each $\boldsymbol{k^{th}}$ state in Figure 5.1.



Time Update *(prediction)*	**Measurement Update** *(correction)*
1- Predict the state ahead $X_k^- = A\,x_{k-1} + W_k$ 2- Predict the error covariance ahead $P_k{}^- = A\,P_{k-1}A^T + Q_K$	1- Compute the Kalman Gain $K_k = P_k{}^-\,H^T(\,HP_k{}^-\,H^T + R)^{-1}$ 2- Update the estimate via Z_k $X_k = X_k{}^- + K(Z_k - H\,X_k{}^-)$ 3- Update the error covariance $P_k = (1 - K_k\,H)P_k{}^-$

Figure 5.1: Kalman filter's equations.

In the above equation $\mathrm{X_k^-}$ is the "prior estimate" which means the rough estimate before the measurement update correction and also $\mathrm{P_k}{}^-$ is the "prior error covariance". We use these "prior" values in our "Measurement Update" equations.

In Measurement Update equations, we find $\mathrm{X_k}$ which is the estimate of X at time k and also we find $\mathrm{P_k}$ which is necessary for the $\mathrm{k+1}$ (future) estimate, together with $\mathrm{X_k}$. The Kalman Gain $\mathrm{K_k}$ we evaluate in this phase is not needed for the next iteration step however it is the most important part of set of equations in Measurement Update equations. The values we evaluate at Measurement Update stage are also called "posterior" values.

These equations show that Kalman filter gain only depends on model parameters and the initial covariance P^{-}. Therefore, Kalman filter gain is independent from real time measurements and it is a time-variant linear system. Besides, these equations show that Kalman filter acts like a feedback control system. In our case Impulse Radar is the senor and measurement of the Impulse Radar is a reference signal to Kalman filter as feedback system.

As previously described in chapter 4 because of imperfect Impulse Radar, temperature variation etc, our measurements are noisy. Therefore, the instantaneous position and velocity, derived from impulse radar are inaccurate As a result, we use Kalman filter to estimate accurate position and velocity of the in the future.

5.3 Evaluation and Results

Figure *4.12* and Figure *4.13* in chapter 4 show movements and position of the mirror from close to further distance of the radar. The mirror is initially located at sample point $\mathrm{x} = 1$ and moves along the x axis with velocity varying around a constant speed 50 mm/sec. The motion equation of the mirror is:

$$x = x_0 + v * t \qquad (5.1)$$

$x(1) = 0$ and $x^{\cdot}(\mathit{1}) = 50$ are initial position and velocity respectively. So from (5.1) it is easy to see the position of the mirror will be 2.1 m after 44s and then the position is constant.

Estimation of the mirror's position and velocity in Figure 5.2 and Figure 5.3 is performed by the Kalman Filter. Kalman filter is applied to estimate position and velocity from the noisy measurements.

Figure 5.2 and Figure 5.3 show the estimated positions and velocities by using the tracking algorithm. It can be seen that the Kalman filter algorithm gives a very reasonable result. Estimated position and velocity of the moving mirror has slightly different from the true measurements. Figure 5.3 shows that true velocity has some variation about. However Kalman filter estimation predicts roughly 50 mm/s.Kalman Filter MATLAB code can be seen in Appendix 8.4.

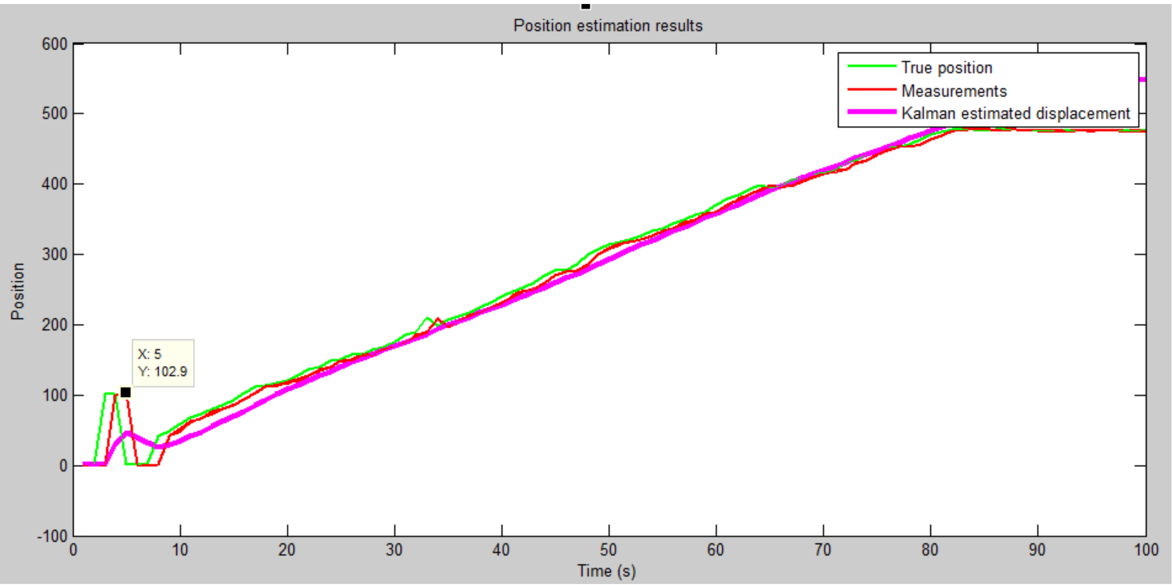


Figure 5.2: Estimated position by Kalman filter algorithm.

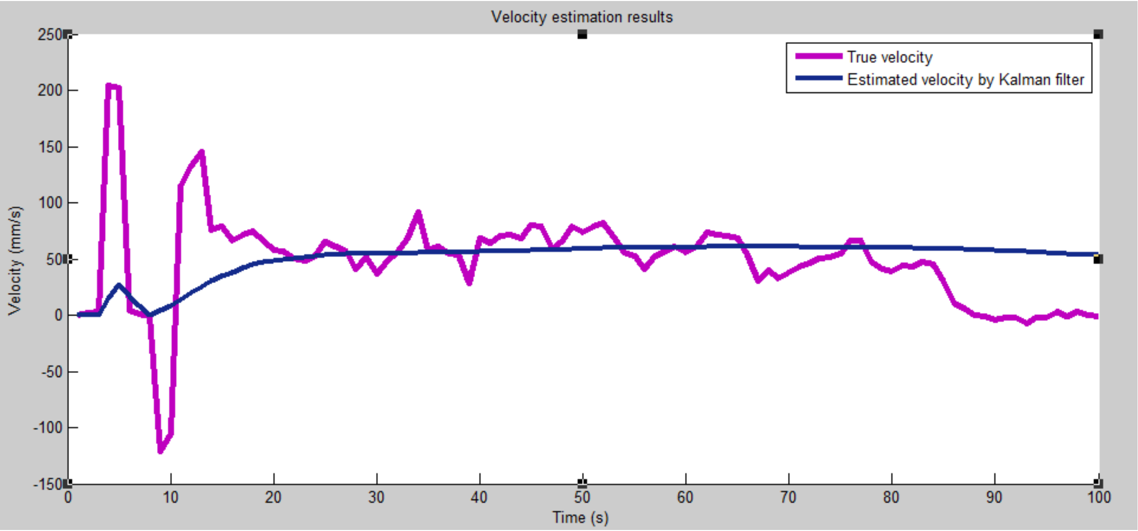


Figure 5.3: Estimated velocity by Kalman filter algorithm.

6 Conclusion and future work

6.1 Conclusion

The goal of the work presented in this thesis was developement of signal processing algorithms for UWB radar. In this project we got the opportunity to work with Novelda radar technology and basic signal processing in MATLAB.

The first step was understanding the theory behind Novelda UWB Impulse Radar. Therefore, a comprehensive study of UWB radar system consisting of two directional vivaldi antennas and UWB CMOS chip presented. In summary, UWB radar transmits an UltraWideband electromagnetic pulse through an antenna. Then the pulse propagates through the media and got reflected by different objects in the media. Next Pulses back to the receiver antenna and the radar samples these pulses.

The received pulse consists of three components including clutter, the reflections from the object, and noise. We needed to eliminate the clutter and noise to retrieve the pulse of interest. Singular Value Decomposition and DC removal methods were two filters that we applied to reduce the presence of clutter and noise.

The main part of signal processing in this project was developing an algorithm to check static object by Impulse Radar. Therefore, we needed to develop an algorithm to find a relationship between radar received signal amplitude and distance. For our project we took some practical measurements and calculated an Empirical formula with Curve fitting method. Moreover, we extended this algorithm to detect both static and moving object at the same time, and also measuring distance of objects.

Measuring and predicting the correct position and velocity of a moving object is demanding in radar application. Therefore we used Kalman filter equations and algorithms to track the moving object in one dimension.

Suggested algorithms that we used in this project gave acceptable results for limited applications. All the measurement in this project was carried out with a mirror with good reflection. Moreover, all the measurement was done in the radar line of sight.

6.2 Future work

6.2.1 Two Dimensional Tracking

In this thesis we just work with one Impulse Radar. Therefore, we had to focus on one dimension measurement. For this reason we had to use mirror with high reflection along the radar line of sight. Using two Radars helps to measure the presence and movement of object in two dimensions. Moreover, it is not necessary to use an object with high reflection like mirror or we do not have to move the object only in the radar line of sight.

6.2.2 Radar Tracking Using Alpha Beta Filter

Two types of tracking filter are used as radar tracking filters:

1. The Kalman filter.
2. The Alpha-Beta filter.

Both Kalman filter and Alpha-Beta filter are used for tracking the position and velocity.
It is suggested to use both filters. Tracking performance with both filters can be compared and evaluated.

7 Reference[i]

[1] Novelda, http://www.novelda.no/.

[2] M. Z. Win, D. Dardari, A. F. Molisch, W. Wiesbeck, and J. Zhang, “History and applications of UWB,” Proceedings of the IEEE, vol. 97, no. 2, pp. 198–204, 2009.

[3] P. J. Hall, “The square kilometre array: an international engineering perspective,” Experimental Astronomy, vol. 17, no. 1–3, pp. 5–16, 2004.

[4] Q. Ren, “Energy detection performance analysis for UWB radar sensor networks,” Eurasip Journal on Wireless Communications and Networking, vol. 2010, Article ID 709723, 2010.

[5] H. A. Hjortland. UWB impulse radar in 90nm CMOS. Master’s thesis, Department of Informatics, University of Oslo, 2006.

[6] Yinan Yu, Jian Yang, Tomas McKelvey, Borys Stoew, "A compact UWB indoor and through-wall radar with precise ranging and tracking" International Journal of Antennas and Propagation, Volume 2012, Article ID 678590

[7] R. Brown and P. Hwang, Introduction to Random Signals and Applied Kalman Filtering, vol. 1, JohnWiley & Sons, New York, NY, USA, 1992.

[8] M. Z. Win, D. Dardari, A. F. Molisch, W. Wiesbeck, and J.Zhang, “History and applications of UWB,” Proceedings of the IEEE, vol. 97, no. 2, pp. 198–204, 2009.

[9] J. Liang. Antenna Study and Design for Ultra Wide Band Communication Applications. Department of electronic engineering Queen Mary, University of London, United Kingdom, July 2006

[10] Q. Ren, “Energy detection performance analysis for UWB radar sensor networks,” Eurasip Journal on Wireless Communications and Networking, vol. 2010, Article ID 709723, 2010.

[11] J. Taylor, Ultra-Wideband Radar Technology, CRC, 2001.

[12] R. J. Fontana, "Recent system applications of short-pulse ultra-wideband (UWB) technology," IEEE Transactions on Microwave Theory and Techniques, vol. 52, no. 9, pp. 2087–2104, 2004.

[13] M. Skolnik, Introduction to Radar Systems, ch. 1: An Introduction to Radar, pp. 1–29.McGraw-Hill, 3rd ed., 2002. 1, 3, 18, 19.

[14] R. Abrahamsson, J. Einarsson, J. Goop, E. Larsson, Antenna design and application development for ultra-wideband-radar BS. Swedish. thesis, Chalmers University of Technology, Gothenburg, Sweden, Feb. 2009.

[15] A. Razavi and J. Yang, "Investigation of penetration ability of UWB antennas in nearfield sensing applications," in Proceedings of the 6th European Conference on Antennas and Propagation (Eu-CAP '12), IEEE, Prague, Czech Republic, March 2012.

[16] Wisland DT, Stoa S, Andersen N, Granhaug K, Lande TS, Hjortland HA (2012) CMOS Nanoscale Impulse Radar Utilized in 2-Dimensional ISAR Imaging System IEEE RAD CONF

[17] wikipedia.org/wiki/Clutter_(radar)

[18] Alan J. Laub, Matrix Analysis for Scientists and Engineers, Society for Industrial and Applied Mathematics; December 1, 2004

[19] G. Benmouyal, "Removal of dc offset in current waveforms using digital mimic filtering," IEEE Trans. Power Delivery, vol. 10, no. 2, pp. 621–630, Apr. 1995

[20] Harald Friis, "A Note on a Simple Transmission Formula," Proc. IRE, 34, 1946, pp. 254-256.

[21] Hans Schantz, Near Field Channel Model, IEEE P802.15-04/0417r2, October 27, 2004

[22] Brown, R. G. and P. Y. C. Hwang: Introduction to Random Signals and Applied Kalman Filtering, Second Edition, John Wiley Sons, Inc., 1992.

8 Appendix

In this section the MATLAB code is listed in all its detail.

8.1 Remote Monitoring Tool using Novelda Impulse Radar

```
function RMT
%{
* Remote Monitoring Tool using Novelda Impulse Radar
* FMH606 Master's Thesis: Remote Monitoring by an Impulse
Radar
* TUC Supervisor: Ola Marius Lysaker
* Student: Ensieh Taghi Mohammadi
* Version: 0.1, 29 March 2013
%}
  AppName = 'Remote Monitoring Tool (RMT)- Version 0.1, 29
March 2013';
  MainApp =
figure('MenuBar','none','Name',AppName,'NumberTitle','off');

  fprintf(AppName);
  fprintf('By: Ensieh Taghi Mohammadi\n');
  fprintf('Starting RMT GUI...\n');

  DropMenu = uimenu('Label','&Fule');
  uimenu(DropMenu,'Label','Start
radar','Callback',@StartRadar);
  uimenu(DropMenu,'Label','Options...','Callback',@Dummy,...
      'Separator','on','Accelerator','O');
  uimenu(DropMenu,'Label','Exit','Callback','Close',...
      'Separator','on','Accelerator','X');

  DropMenu = uimenu('Label','&Help');
  uimenu(DropMenu,'Label','Help',...
    'Callback', @Dummy);
  uimenu(DropMenu,'Label','About...','Separator','on',...
    'Callback', @About);

function StartRadar(varargin)
  RMT_Main;

%**************************************************************
***********
%% Function: About
%**************************************************************
***********
function About(varargin)
```

```
  msgbox('Remote Monitoring Tool - Version 0.1 (29 March
2013), by Ensieh Taghi Mohammadi','About','help','modal')

%*************************************************************
***********
%% Function: Dummy
%*************************************************************
***********
function Dummy(varargin)
  msgbox('This operation is currently not
implemented!','Warning','warn','modal')

%%
%% EOF: RMT.m
%%
function RMT_Main
  radarlib='C:\Program Files\Novelda\Radarlib3 API\';
  asmInfo = NET.addAssembly([radarlib,'Radarlib3.NET.dll']);
  portname = 'FTx232H!Serial No: (FTV9JR75)!NVA-R641!NVA6100';

  % Variables
  % The variables that are going to be used are stored in a
map for
  % convenient access and automatic update of the radar
software and hardware
  Variables = containers.Map;
  Variables('AveragingFactor') = 5;
  Variables('Iterations') = 10;
  Variables('Gain') = 5;
  Variables('ZoomMin') = 30;
  Variables('ZoomMax') = 70;
  Variables('FrameOffset') = 80;

  try
    % Initialize connection
    radar = Novelda.Radarlib3Wrapper.RadarWrapper;
    radar.Open(portname);
    fprintf('Object counter %d\n', radar.ObjectCounter);
    fprintf('Connected to Novelda Radar on port %s\n',
portname);
  catch ME
    % Print error message
    fprintf('Fatal: Connection error\n');
    fprintf('\n%s\n',ME.message);
    % Try to get name of connected modules
    modules = radar.ConnectedModules;
    if modules(1).Length > 0
        fprintf('\nAvailable Radar Modules: \n');
        for module = 1:modules.Length
            moduleName = char(modules(module));
            fprintf('%s\n', moduleName);
        end
    else
        fprintf('\nNo Radar Modules found!\n');
    end
    return
end

% Initialize some local variables
try
    % Read the NumberOfCounters variable from library
    numCounters =
radar.Item('SamplersPerFrame').Value.ToInt();
    % Allocate space for the datavector
    maxlen = 100;
    datavector = zeros(maxlen,numCounters);
    frame = zeros(numCounters,1);
    frameptr = libpointer('doublePtr',frame);
    n = 0;
    l = 0;
    i = 0;
    time= 0;

    % Update parameters by browsing through the Variables map
    varlist = keys(Variables);
    for n=1:length(varlist)
        fprintf('Setting %s to %d\n', char(varlist(n)),
Variables(char(varlist(n))));

radar.TryUpdateChip(char(varlist(n)),Variables(char(varlist(n)
)));
    end

    % Define graphics
    % Figure 1 is used to plot the raw signal
    figure;
    axes('xlim', [1,numCounters], 'ylim', [40, 60]);
    h = line(1:numCounters, frame);
    shg;


    % Read and plot data from the radar

    % Continue reading frames as long as the plot window is
valid
    while(ishandle(h))
        tic;
        c = double(radar.GetFrameNormalizedDouble());
        set(h,'ydata',c);
        % Draw frames takes some time.
        % Uncomment "if statement" to get higher frame rate
        if (mod(i, 10) == 0)
            drawnow;
        end
        time = time + toc;
        i = i+1;
        % Store the data to datavector
        if(i<=maxlen)
            datavector(i,:) = c;
        end
    end

    fprintf('FPS: %f\n',i/time);

catch ME
    fprintf('An error occurred\n');
    fprintf('\n%s\n',ME.message);
    % Close the connection to the radar
    fprintf('Closing connection...\n');
    radar.Close();
    return
end
% Close the connection to the radar
fprintf('Closing connection...\n');
radar.Close();

end
%
% EOF: RMT_Main.m
```

8.2 Filtering

8.2.1SVD Filter

```
[U,S,V] = svd(datavector);
Clutter1 = U(:,1:2)*S(1:2,1:2)*V(:,1:2)';
removeclutter=datavector- clutter1;
abs(removeclutter);
plot(removeclutter')
grid on
xlim([1,512]);
xlabel('Number of Samples')
ylabel('Level of Signal')
```

8.2.2DC Removal Filter

```
dcremoval= datavector-meandatavector;
removeclutter=dcremoval- clutter1
abs(removeclutter);
plot(removeclutter')
grid on
```

```
xlim([1,512]);
xlabel('Number of Samples')
ylabel('Level of Signal')
```

8.3 Object Detection

8.3.1Treshold Equation

```
position=[51;65;126;160;194;233;288;331;383;495]
ampitude=[63.56;61.08;59;56.68;55.49;53.86;52.49;52.26;52.25;5
1.29]
B=[1 51; 1 65; 1 126; 1 160; 1 194; 1 233; 1 288; 1 331; 1
383; 1 495]
A=1./B
c=(inv(A'*A))*A'*(ampitude)
x=linspace(40,512)
y=c[1]+ (c[2])/x
plot(x,y)
hold on
plot(position,ampitude,'*')
```

8.3.2Threshold Algorithm

```
for counter=1:100  %next row

     for ii=40:512
       j=-sweep;
       %target(counter,ii)=0;
      while(j<sweep)
         if (ii+j>=1)&&(ii+j<512)
             if frame1(counter,ii+j)>threshold(1,ii+j)
                  target(counter,ii)=target(counter,ii)+1;
             end
         end
      j=j+1;
      end
      %finished 1 column
      end
      % finished 1 row
  end


  target
```

8.3.3Object Detection

```
max1=zeros(1,100)

for row=1:100
       for column=40:512
           [max1,num]=max(target(row,:)) ;
          nn(1,row)=num;
          yy(1,row)=max1;


       end
end

 distance1=nn.*2.29/512;
```

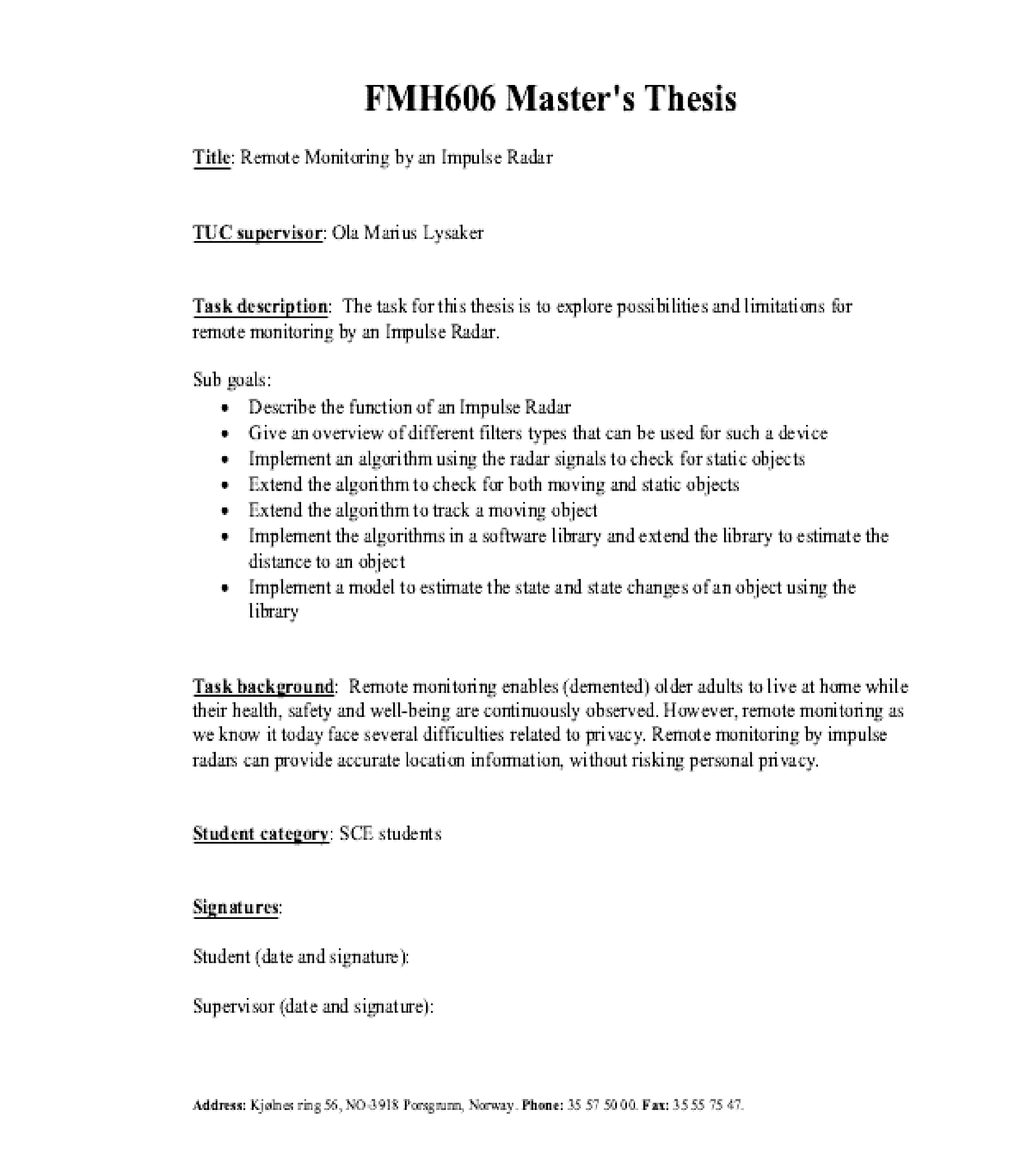
8.4 Kalman Filter

```
%Kalman Filter Definition
% Set true trajectory
Nsamples=counter;
dt = .1;
t=1:1:counter
Vtrue = 0.01;

% Xtrue is a vector of true positions of the train
Xinitial = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Motion equations %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Xk_prev = [1;
    .5*Vtrue];

% Current state estimate
Xk=[];

Phi = [1 dt;
       0  1];

%
sigma_model = 1;
% P = sigma^2*G*G';
P = [sigma_model^2             0;
                 0 sigma_model^2];
```

```
% Q is the process noise covariance. It represents the amount
of
% uncertainty in the model. In our case, we arbitrarily assume
that the model is perfect (no
Q = [0 0;
     0 0];

% M is the measurement matrix.
% We measure X, so M(1) = 1
% We do not measure V, so M(2)= 0
M = [1 0];

% R is the measurement noise covariance. Generally R and
sigma_meas can
% vary between samples.
sigma_meas = 1; % 1 m/sec
R = sigma_meas^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Kalman iteration %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Buffers for later display
Xk_buffer = zeros(2,Nsamples);
Xk_buffer(:,1) = Xk_prev;
%z_buffer = zeros(1,Nsamples);

for k=1:Nsamples-1

    % Z is the measurement vector. In our
    % case, Z = TrueData + RandomGaussianNoise
    %Z = Xtrue(k+1)+sigma_meas*randn;
    z=nn(k)+sigma_meas*randn
    z_buffer(k+1) = z;

    % Kalman iteration
    P1 = Phi*P*Phi' + Q;
    S = M*P1*M' + R;

    % K is Kalman gain. If K is large, more weight goes to the
measurement.
    % If K is low, more weight goes to the model prediction.
    K = P1*M'*inv(S);
    P = P1 - K*M*P1;

    Xk = Phi*Xk_prev + K*(z-M*Phi*Xk_prev);
    Xk_buffer(:,k+1) = Xk

    % For the next iteration
    Xk_prev = Xk;
end;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Plot resulting graphs %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Position analysis %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3) ;
plot(t,nn,'g');
hold on;
plot(t,z_buffer,'r');
plot(t,Xk_buffer(1,:),'m');
title('Position estimation results');
xlabel('Time (s)');
ylabel('Position');
legend('True position','Measurements','Kalman estimated
displacement');


  %#########################################
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Velocity analysis %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The instantaneous velocity as derived from 2 consecutive
position
% measurements
InstantaneousVelocity = [0 (z_buffer(2:Nsamples)-
z_buffer(1:Nsamples-1))/dt];

% The instantaneous velocity as derived from running average
with a window
% of 5 samples from instantaneous velocity
WindowSize = 5;
InstantaneousVelocityRunningAverage =
filter(ones(1,WindowSize)/WindowSize,1,InstantaneousVelocity);

figure (4) ;
%plot(t,ones(size(t))*Vtrue,'m');
hold on;
%plot(t,InstantaneousVelocity,'g');
plot(t,InstantaneousVelocityRunningAverage,'c');
plot(t,Xk_buffer(2,:),'k');
title('Velocity estimation results');
xlabel('Time (s)');
ylabel('Velocity (mm/s)');
legend('True velocity','Estimated velocity by Kalman filter');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

8.5 Thesis Definition



FMH606 Master's Thesis

Title: Remote Monitoring by an Impulse Radar

TUC supervisor: Ola Marius Lysaker

Task description: The task for this thesis is to explore possibilities and limitations for remote monitoring by an Impulse Radar.

Sub goals:

- Describe the function of an Impulse Radar
- Give an overview of different filters types that can be used for such a device
- Implement an algorithm using the radar signals to check for static objects
- Extend the algorithm to check for both moving and static objects
- Extend the algorithm to track a moving object
- Implement the algorithms in a software library and extend the library to estimate the distance to an object
- Implement a model to estimate the state and state changes of an object using the library

Task background: Remote monitoring enables (demented) older adults to live at home while their health, safety and well-being are continuously observed. However, remote monitoring as we know it today face several difficulties related to privacy. Remote monitoring by impulse radars can provide accurate location information, without risking personal privacy.

Student category: SCE students

Signatures:

Student (date and signature):

Supervisor (date and signature):

Address: Kjølnes ring 56, NO-3918 Porsgrunn, Norway. **Phone:** 35 57 50 00. **Fax:** 35 55 75 47.

Telemark University College
Faculty of Technology
Kjølnes
3914 Porsgrunn