Efficient Material Characterization by Means of the Doppler Effect in Microwaves

Dissertation

zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät
der Universität des Saarlandes

Roman Pinchuk

pinchuk@cs.uni-sb.de
roman.pinchuk@izfp.fraunhofer.de

Saarbrücken, Juli 2007

Tag des Kolloquiums:	
Dekan:	Prof. Dr.-Ing. Thorsten Herfet
Vorsitzender des Prüfungsausschusses:	Prof. Dr.-Ing. Philipp Slusallek
1. Berichterstatter:	Prof. Dr. Wolfgang J. Paul
2. Berichterstatter:	Prof. Dr. hc. mult. Michael Kröning
akademischer Mitarbeiter:	Dr. Jörg Baus

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, im Mai 2007

Acknowledgements

I am most indebted to Prof. Michael Kröning for giving me an opportunity to start with my graduation. I am very grateful for his endless support and guidance for both, technical and non-technical issues.

I would like to gratefully acknowledge my supervisor Prof. Wolfgang J.Paul for his enthusiastic supervision, consistent support, advice, and managing to read the whole thing so thoroughly and legibly.

I acknowledge the help of Dr. Christoph Sklarczyk for his technical support, relevant discussion, and general advice.

My heartfelt thanks to my wife Elena. I appreciate for her continual encouragement, love, and harp editorial pencil.

Furthermore, I would like to thank my colleges and friends at Institute for Non-Destructive Testing (IZfP), Saarbrücken, for creating such a splendid atmosphere for carnying on my PhD.

Finally, I am always indebted to my parents for their blessing, encouraging, and inspiration.

Abstract

Subject of this thesis is the efficient material characterization and defects detection by means of the Doppler effect with microwaves.

The first main goal of the work is to develop a prototype of a microwave Doppler system for Non-Destructive Testing (NDT) purposes. Therefore it is necessary that the Doppler system satisfies the following requirements: non-expensive, easily integrated into industrial process, allows fast measurements. The Doppler system needs to include software for hardware control, measurements, and fast signal processing.

The second main goal of the thesis is to establish and experimentally confirm possible practical applications of the Doppler system.

The Doppler system consists of the following parts. The hardware part is designed in a way to ensure fast measurement and easy adjustment to the different radar types. The software part of the system contains tools for: hardware control, data acquisition, signal processing and representing data to the user.

In this work firstly a new type of 2D Doppler amplitude imaging was developed and formalized. Such a technique is used to derive information about the measured object from several angles of view.

In the thesis special attention was paid to the frequency analysis of the measured signals as a means to improve spatial resolution of the radar. In the context of frequency analysis we present 2D Doppler frequency imaging and compare it with amplitude imaging.

In the thesis the spatial resolution ability of CW radars is examined and improved. We show that the joint frequency and the amplitude signal processing allows to significantly increase the spatial resolution of the radar.

Kurzzusammenfassung

Das Thema dieser Dissertation ist die effiziente Materialcharakterisierung und Fehlerdetektion durch Nutzung des Dopplereffektes mittels Mikrowellen.

Das erste Hauptziel der Arbeit ist die Entwicklung eines Prototyps eines Mikrowellen-Doppler-Systems im Bereich der zerstörungsfreien Prüfung. Das Doppler-System muss folgenden Voraussetzungen erfüllen: es sollte preisgünstig sein, leicht in industrielle Prozesse integrierbar sein und schnelle Messungen erlauben. Das Doppler-System muss die Software für die Hardware-Kontrolle, den Messablauf und die schnelle Signalverarbeitung beinhalten.

Das zweite Hauptziel der Dissertation ist es, mögliche praktische Anwendungsfelder des Doppler-Systems zu identifizieren und experimentell zu bearbeiten.

Das Doppler-System besteht aus zwei Teilen. Der Hardware-Teil ist so konstruiert, dass er schnelle Messungen und leichte Anpassungen an verschiedene Sensor- und Radartypen zulässt. Der Software-Teil des Systems beinhaltet Werkzeuge für: Hardware-Kontrolle, Datenerfassung, Signalverarbeitung und Programme, um die Daten für den Benutzer zu präsentieren.

In dieser Arbeit wurde zuerst ein neuer Typ der 2D-Doppler-Amplitudenbildgebung entwickelt und formalisiert. Dieser Technik wird dafür benutzt, Informationen über die gemessenen Objekte von verschiedenen Blickpunkten aus zu erhalten.

In dieser Doktorarbeit wird der Frequenzanalyse der gemessenen Signale besondere Aufmerksamkeit geschenkt, um die Ortsauflösung des Radars zu verbessern. Im Kontext der Frequenzanalyse wird die 2D-Doppler-Frequenzbildgebung präsentiert und mit der Amplitudenbildgebung vergleichen.

In dieser Dissertation werden die räumliche Auflösungsmöglichkeiten von CW-Radaren untersucht und verbessert. Es wird gezeigt, dass es die Frequenz- und Amplitudensignalverarbeitung erlaubt, die Ortsauflösung des Radars erheblich zu erhöhen.

Extended Abstract

Subject of this thesis is the efficient material characterization and defects detection by means of the Doppler effect with microwaves.

The first main goal of the work is to develop a prototype of a microwave Doppler system for Non-Destructive Testing (NDT) purposes. Therefore it is necessary that the Doppler system satisfies the following requirements: non-expensive, easily integrated into industrial process, allows fast measurements. The Doppler system needs to include software for hardware control, measurements, and fast signal processing. The low cost of the Doppler system is reached using continuous wave (CW) Doppler radars based on Gunn-transceivers. The CW radars produce signals of low frequency. This allows the usage of non-expensive measurement equipment available on the market. Since the microwaves operate in the air, the integration of the Doppler system into industrial processes is very simple. Thus, there is no need to couple the Doppler radar to the analyzed specimen. The Doppler-effect only appears if there is a relative movement between specimen and radar. Therefore it is well suited for measurements with quick radars or quick specimens. In order to reach the high spatial resolution often needed in the domain of non-destructive testing by using low-cost Doppler radars it is necessary to apply highly developed signal processing algorithms with high complexity. These have to be speeded up to ensure that the data evaluation is performed in reasonable time.

The second main goal of the thesis is to establish and experimentally confirm possible practical applications of the Doppler system.

The Doppler system consists of the following parts. The hardware part is designed in a way to ensure fast measurement and easy adjustment to the different radar types. The software part of the system contains tools for: hardware control, data acquisition, signal processing and representing data to the user. The signal processing tool includes algorithms which were developed to deal with Doppler measured data. The optimized versions of the algorithms have been developed, formalized and experimentally confirmed.

In this work firstly a new type of 2D Doppler amplitude imaging was developed and formalized. Such a technique is used to derive information about the measured object from several angles of view. This imaging allows the user to discover most of the defects of the analyzed specimen. Hardware realization and software implementation for Doppler imaging are presented.

In the thesis special attention was paid to the frequency analysis of the measured signals as a means to improve spatial resolution of the radar. In the context of frequency analysis we present 2D Doppler frequency imaging and compare it with amplitude imaging. We also present detailed comparison study of different algorithms which includes implementation features, testing on modelled and measured data and complexity analysis. Among others, we examine such algorithms as: Adaptive IF estimation, Linear Least Squares problems, Polynomial-Phase Difference techniques, and most famous, Time-Frequency Distributions.

In the thesis the spatial resolution ability of CW radars is examined and improved. We show that the joint frequency and the amplitude signal processing allows to significantly increase the spatial resolution of the radar. In that con-

text, we introduce the Maximum Entropy Deconvolution (MED) algorithm. Its optimized version is developed, formalized, and experimentally confirmed. The complexity of MED is reduced, from $O(n^3)$ to $O(m\,n^2)$ with $m \ll n$, applying the iterative GMRES (Generalized Residual) algorithm. Here the optimal preconditioning technique is developed, proved and experimentally confirmed. We also propose possible hardware implementation of MED algorithm which is based on the optimized Gauss-Elimination algorithm. Here we show that some properties of measured Doppler signal can be utilized to speed-up the optimized Gauss Elimination algorithm.

In conclusion we present possible practical applications of the Doppler system.

Erweiterte Zusammenfassung

Das Thema dieser Dissertation ist die effiziente Materialcharakterisierung und Fehlerdetektion durch Nutzung des Dopplereffektes mittels Mikrowellen.

Das erste Hauptziel der Arbeit ist die Entwicklung eines Prototyps eines Mikrowellen-Doppler-Systems im Bereich der zerstörungsfreien Prüfung. Das Doppler-System muss folgenden Voraussetzungen erfüllen: es sollte preisgünstig sein, leicht in industrielle Prozesse integrierbar sein und schnelle Messungen erlauben. Das Doppler-System muss die Software für die Hardware-Kontrolle, den Messablauf und die schnelle Signalverarbeitung beinhalten. Die niedrigen Kosten des Doppler-Systems wurden durch die Verwendung eines Dauerstrich-Doppler-Radars (Continuous Wave (CW)-Radar) auf der Basis eines Gunn-Transceivers (Sende-Empfangs-Einheit) erreicht. CW-Radare erzeugen Messsignaldaten mit niedriger Frequenz. Dies erlaubt den Gebrauch von preisgünstigen auf dem Markt verfügbaren Messgeräten. Da die Mikrowellen sich in den Luft ausbreiten, ist die Integration des Doppler-Systems in den industriellen Prozess sehr einfach. Eine Ankopplung des Doppler-Radars an das untersuchte Messobjekt ist nicht notwendig. Der Doppler-Effekt tritt nur auf, wenn sich Objekt und Sensor relativ zueinander bewegen. Er eignet sich daher gut für Messungen mit schnell bewegten Sensoren und an schnellen Objekten. Um mit den preisgünstigen Doppler-Radaren die in der zerstörungsfreien Prüfung oft geforderte hohe Ortsauflösung zu erreichen, ist der Einsatz hochentwickelter komplexer Signalverarbeitungsalgorithmen erforderlich. Diese müssen beschleunigt werden, um eine Datenauswertung in einer praktikablen Zeit zu gewährleisten.

Das zweite Hauptziel der Dissertation ist es, mögliche praktische Anwendungsfelder des Doppler-Systems zu identifizieren und experimentell zu bearbeiten.

Das Doppler-System besteht aus zwei Teilen. Der Hardware-Teil ist so konstruiert, dass er schnelle Messungen und leichte Anpassungen an verschiedene Sensor- und Radartypen zulässt. Der Software-Teil des Systems beinhaltet Werkzeuge für: Hardware-Kontrolle, Datenerfassung, Signalverarbeitung und Programme, um die Daten für den Benutzer zu präsentieren. Das Signalverarbeitungswerkzeug beinhaltet Algorithmen, die dafür entwickelt wurden, um die Daten, welche mit dem Doppler-System gemessen wurden, zu bearbeiten. Die im Verlauf der Arbeit entwickelten optimierten Versionen der Algorithmen wurden formalisiert und experimentell bestätigt.

In dieser Arbeit wurde zuerst ein neuer Typ der 2D-Doppler-Amplitudenbildgebung entwickelt und formalisiert. Dieser Technik wird dafür benutzt, Informationen über die gemessenen Objekte von verschiedenen Blickpunkten aus zu erhalten. Diese Bildgebung erlaubt es dem Benutzer, fast alle Fehler, welche das analysierte Objekt hat, zu entdecken. Hardware -und Softwareausführungen für die Doppler-Bildgebung werden präsentiert.

In dieser Doktorarbeit wird der Frequenzanalyse der gemessenen Signale besondere Aufmerksamkeit geschenkt, um die Ortsauflösung des Radars zu verbessern. Im Kontext der Frequenzanalyse wird die 2D-Doppler-Frequenzbildgebung präsentiert und mit der Amplitudenbildgebung vergleichen. Auerdem wird eine detaillierte Vergleichstudie von verschiedenen Algorithmen, welche Ausführungseinzelheiten, Testergebnisse von simulierten und realen Daten und eine Komplexitäts-

analyse beinhalten, präsentiert. Unter anderen wurden folgenden Algorithmen untersucht: "Adaptive IF estimation", "Polynomial-Phase Difference techniques" und am meisten bekannt: "Time-Frequency Distributions".

In dieser Dissertation werden die räumliche Auflösungsmöglichkeiten von CW-Radaren untersucht und verbessert. Es wird gezeigt, dass es die Frequenz- und Amplitudensignalverarbeitung erlaubt, die Ortsauflösung des Radars erheblich zu erhöhen. In diesem Zusammenhang wird der "Maximum Entropy Deconvolution"-Algorithmus (MED) eingeführt. Seine optimierte Version wurde im Verlauf der Arbeit entwickelt, formalisiert und experimentell bestätigt. Die Komplexität des MED wurde von $O(n^3)$ auf $O(m\,n^2)$ mit $m \ll n$ reduziert, wobei der iterative GMRES-Algorithmus verwendet wurde. Im Verlauf der Dissertation wurde die Optimale Präkonditionstechnologie entwickelt, geprüft und experimentell bestätigt. Auerdem wurde eine mögliche Hardwareausführung des MED-Algorithmus, welche auf dem optimiertem "Gauss-Elimination"-Algorithmus basiert, vorgeschlagen. Dabei wird gezeigt, dass einige Eigenschaften des gemessenen Doppler-Signals genutzt werden können, um den optimalen Gauss-Elimination-Algorithmus zu beschleunigen.

Im Abschluss werden mögliche praktische Anwendungen des Doppler-Systems aufgezeigt.

Contents

List of Figures

Chapter 1

Introduction

Nowadays producing and exploitation of wares is inseparably linked with their quality control. For the last ten years in all fields of our live the quality demands have become very high. The competitive character of the modern industry forces manufacturers to check not every hundredth, as it was in the past, but all of the produced items. In order not to stop the production line it is required to integrate the high-speed quality control in the production process. For several items it is important that their quality control is performed during the whole period of the exploitation (e.g. turbines, planes, fuel injectors etc.).

The problem of quality control in all stages of production and exploitation belongs to the field of non-destructive testing (NDT). NDT itself can be split into branches where every branch deals with specific physical phenomena and materials of specific properties. A physical phenomenon is used as an implicit mean to gather information about the object of control. In general, in non-destructive testing, two tasks are under consideration. The first task is developing of sensors to measure characteristics of physical phenomena. The second one is processing of measured data.

Nowadays a great number of sensors for different branches of NDT became available on the market. Unfortunately, with rare exception, the sensors of high quality are very expensive. Moreover in some cases the sensors are immobile what makes them impossible to be integrated into an industrial process. On the other hand, the use of cheap sensors may significantly reduce an amount of acquired information about object or even make its quality control impossible. In some cases the lack of the information may be compensated through the excessive data acquisition. Then, the measured data are processed by some signal processing algorithms. Usually, algorithms which deal with a big amount of data about the process are computationally complex. This leads to high computational demands and increases time of computations so that real-time quality control becomes infeasible.

In the work described in this thesis we develop a prototype system for efficient non-destructive testing based on using of continuous wave (CW) radars. The physical phenomenon we exploit to detect defects in materials is the Doppler effect.

The sensors we suggest to use are cheap and available on the market. Since these sensors do not provide any range information we acquire it by mechanical tracking of the actual sensor position. We also suggest an approach for fast measurements which allows collection the information about the tested object.

We suggest an approach to represent a scope of measured data to the user in easly understandable form. We describe (if they are already known) and develop the signal and image processing techniques necessary for this approach.

There is a number of applications where the spatial resolution of the used sensors is not needed to be high to satisfy the requirements. In this thesis we represent detailed analysis of different signal processing algorithms which are used to increase resolution. We reuse and test some of them for the use in microwave NDT. Since the complexity of algorithms is very high we develop their fast versions exploiting some properties of the measured data.

We also outline some practical problems where the developed system can be applied and demonstrate the results of the performed experiments.

1.1 Definitions and Notations

We denote the set of *natural numbers* including zero as $\mathbb{N}$ and use $\mathbb{N}^+$ for $\mathbb{N}\backslash\{0\}$. The set of *integer numbers* is given by $\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$. The sets of *real* and *complex* numbers are denoted as $\mathbb{R}$ and $\mathbb{C}$, respectively.

Definition 1.1.1 *Let $m, n \in \mathbb{Z}$ be numbers. We define* ***integer intervals*** *as*

$$\begin{aligned}
[m:n] &:= m, \dots, n \\
[m:n[&:= m, \dots, n-1 \\
]m:n] &:= m-1, \dots, n
\end{aligned}$$

In this work we intensively operate with *analog* and *discrete* signals. Their definitions are given below.

Definition 1.1.2 *A* ***real-valued analog signal*** *is a function $x : \mathbb{R} \to \mathbb{R}$, where $x(t)$ is the signal value at time t. A* ***complex-valued analog signal*** *is a function $x : \mathbb{R} \to \mathbb{C}$, such that $x(t) = x^r(t) + jx^i(t)$. Value $x^r(t)$ is the real part of x; x^i is the imaginary part of $x(t)$; $j^2 = -1$ is an imaginary unit. Both x^r and x^i are real-valued analog signals.*

Definition 1.1.3 *Let $\mathbb{T}$ be an arbitrary type. Thus, it can be a set of real numbers $\mathbb{R}$, complex numbers $\mathbb{C}$, or arbitrary type, etc. We define $\mathbb{T}^n$ to be the set of* ***vectors*** *of size $n \in \mathbb{N}^+$. We refer to $\mathbb{T}^{1\times n}$ and $\mathbb{T}^{n\times 1}$ as the sets of* ***row*** *and* ***column-vectors*** *such that $\mathbb{T}^{1\times n}, \mathbb{T}^{n\times 1} \subset \mathbb{T}^n$. An i-th entry of vector $\mathbf{v} \in \mathbb{T}^n$ is addressed as v_i, where $i \in [0 : n-1]$.*

We denote a vector by a lower case letter of a bold style. For addressing the particular vector entry we use the same letter of a non-bold style.

Definition 1.1.4 *Any column-vector can be transformed into the corresponding row-vector and vice versa by means of the* ***transpose*** *operation. Let $\mathbf{v} \in \mathbb{T}^{1\times n}$ be* *a row-vector, then*

$$\mathbf{v}^T = (\mathrm{v}_0, \mathrm{v}_1, \ldots, \mathrm{v}_{n-1})^T = \begin{pmatrix} \mathrm{v}_0 \\ \mathrm{v}_1 \\ \vdots \\ \mathrm{v}_{n-1} \end{pmatrix}$$

Definition 1.1.5 *Let $\mathbf{v} \in \mathbb{T}^n$ be a vector, then its **length** is defined as*

$$|\,\mathbf{v}\,| = n$$

Definition 1.1.6 *Let $\mathbb{T}$ be either $\mathbb{R}$ or $\mathbb{C}$. We define the function*

$$\mathrm{abs} : \mathbb{T}^n \to \mathbb{R}^n$$

such that for $\mathbf{v}' \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{T}^n$, and $\mathbf{v}' = \mathrm{abs}(\mathbf{v})$ we have

$$\text{for all } i \in [0 : n-1] \quad \mathrm{v}'_i = \sqrt{\mathrm{Re}(\mathrm{v}_i)^2 + \mathrm{Im}(\mathrm{v}_i)^2}$$

In the latter definition functions **Re** and **Im** stand for extraction of the real and imaginary parts of a complex number, respectively.

Definition 1.1.7 *We define the function*

$$\arg : \mathbb{C}^n \to \mathbb{R}^n$$

such that for $\mathbf{v}' \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{C}^n$, and $\mathbf{v}' = \arg(\mathbf{v})$ we have

$$\text{for all } i \in [0 : n-1] \quad \mathrm{v}'_i = \arctan\left(\frac{\mathrm{Im}(\mathrm{v}_i)}{\mathrm{Re}(\mathrm{v}_i)}\right)$$

Definition 1.1.8 *Let $\mathbf{v} \in \mathbb{R}^n$ be a vector. We define its **second norm** as*

$$\|\mathbf{v}\|_2 = \sqrt{\sum_{i=0}^{n-1} \mathrm{abs}(\mathrm{v}_i)^2}$$

Definition 1.1.9 *A **real-valued discrete signal** is a function $\hat{x} : \mathbb{Z} \to \mathbb{R}$, where $\hat{x}(n)$ is the signal value (sample) at time instant n. A **complex-valued discrete signal** is a function $\hat{x} : \mathbb{Z} \to \mathbb{C}$.*

We refer to a discrete signal defined on domain $\mathbb{N}$ as a vector.

Definition 1.1.10 *In order to convert an analog signal into its digital representation we use the **sampling** procedure. Let $x : \mathbb{R} \to \mathbb{R}$ be an analog signal. Then the corresponding discrete signal $\mathbf{x} \in \mathbb{R}^n$ is given for all $k \in [0 : n-1]$ as*

$$\mathrm{x}_k = x(k\,\Delta t),$$

where $n \in \mathbb{N}^+$ is a number of samples *and $\Delta t \in \mathbb{R}$ is a **sampling interval**.*

1.2 Microwave Non-Destructive Testing

The term *microwaves* is used to define *electromagnetic waves* or *electromagnetic radiation* of frequency f with range from 300 MHz to 300 GHz. Microwaves have many scientific and industrial applications. These include wireless communications, telemetry, biomedical engineering, food science, medicine, material processing, and process control in the industry [1].

Microwave non-destructive testing (microwave NDT) is defined as inspection of materials and structures using microwaves without damaging or preventing the future use of the tested object [2], [3]. These are some among others areas that may benefit from using of microwave NDT:

- Composite inspection (accurate thickness measurement, detection of material impact damages, and corrosion under the paint etc.)
- Material surface inspection (stress and cracks detection, surface roughness evaluation etc.)
- Microwave imaging (imaging of surface interior)
- Medical and industrial applications (detection of buried objects, humidity detection, detection of unhealthy skin patches)

More detailed information about microwave NDT and its applications can be found in [4].

There is a number of physical microwaves properties which make them particulary attractive for NDT. Microwaves are able to *penetrate* into *dielectric* materials. The *spatial resolution* of microwaves varies with frequency f and has a range from 1 meter to 1 millimeter. It indicates the ability of microwaves to discern closely spaced discontinuities in the material. Another important advantage of microwaves is its easy coupling with the *medium.* The coupling can be easily done, for example, through air by using an antenna.

One of serious drawbacks of microwave NDT is high equipment cost. In many applications microwave NDT is rather a laboratory method which is difficult to install on the production line. Often, applying advanced signal processing techniques slows down the data evaluation. It makes microwave NDT impossible to operate in real time. This thesis concerns the problem of fast computation in microwave NDT.

1.2.1 Microwaves Propagation

Microwaves i.e. electromagnetic radiation consists of two components. These are time-changing *magnetic field* and associated with it time-changing *electric field*, [5, 6]. Quantitative expression of magnetic and electric fields is given by its *intensities.* We define the electric field intensity or simply electric field $\vec{E}(\vec{r}, t)$ as a function of *spatial location* $\vec{r} = (x, y, z)$ and time t. The spatial location is determined by cartesian coordinates x, y, and z. Similarly we denote magnetic field by $\vec{H}(\vec{r}, t)$.

The propagation of microwaves depends on the interrelation between electric field, magnetic field, and a medium. We understand the medium to be any environment where microwaves propagate in. It can be air, dielectric, or any other material. Every medium is characterized by *relative magnetic permeability* $\mu_r \geq 1$ and *relative dielectric permittivity* $\epsilon_r \geq 1$. Parameters μ_r and ϵ_r express influence of the medium on magnetic and electrical fields, respectively. We assume the medium to be *isotropic*, i.e. properties of the medium do not change with distance or direction. In this case μ_r and ϵ_r are plain constants.

In microwave NDT it is preferable to work with media or materials which are non-magnetic i.e. have $\mu_r = 1$. Under that assumption the propagation of microwaves only depends on electrical properties of the material. Further, we shortly present the theory of microwave only regarding the electric field $\vec{E}$.

The electromagnetic wave equation in terms of electric field $\vec{E}$ is given below

$$\frac{\partial^2 \vec{E}}{\partial t^2} = \frac{1}{\mu \epsilon} \frac{\partial^2 \vec{E}}{\partial z^2} \tag{1.1}$$

We assume that electric field $\vec{E}(\vec{r}, t)$ propagates in direction z, i.e. $\vec{r} = (0, 0, z)$. A *total permeability* μ and *total permittivity* ϵ in (1.1) are given by i) medium constants μ_r and ϵ_r; and ii) *vacuum* (i.e. free space) constants μ_0 and ϵ_0

$$\mu = \mu_r \, \mu_0$$
$$\epsilon = \epsilon_r \, \epsilon_0,$$

where vacuum permeability and permittivity are defined as $\mu_0 = 4\pi \times 10^{-7}$ and $\epsilon_0 = 8.854 \times 10^{-12}$. Microwaves propagate in vacuum at the *speed of light* c_0 defined as

$$c_0 = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \tag{1.2}$$

A solution of (1.1) is a cosine propagated in direction z:

$$\vec{E}(z, t) = \vec{E_0} \cdot \cos(\phi_0 + 2\pi f t - \beta z), \tag{1.3}$$

where $\vec{E_0} = (0, E_y, 0)$ represents the electric field *amplitude*. The *initial phase* ϕ_0 is an angle of the cosine function at time $t = 0$. The frequency of microwaves is given by some constant f. A *propagation factor* β introduces an influence of the medium on *propagation velocity* of $\vec{E}$:

$$\beta = \frac{2\pi f}{c_0} \sqrt{\mu_r \epsilon_r} = \frac{2\pi f}{c}, \tag{1.4}$$

where c is propagation velocity of microwaves in the medium with μ_r and ϵ_r.
If the medium is a free space (i.e. $\mu_r = 1$, $\epsilon_r = 1$), then the velocity in the medium is equal to the velocity of light, i.e. $c = c_0$, see equation (1.4). In that case the difference of terms $2\pi f t$ and βz in (1.3) is always zero since $z = t \, c_0$. This implies propagation of $\vec{E}(z, t)$ at time t and space location z with the initial phase ϕ_0. If the medium is not free space, a propagation of $\vec{E}(z, t)$ is delayed by a *media-dependent* factor δ_ϕ which can be derived from equations (1.3) and (1.4) as

$$\delta_\phi = 2\pi f t (1 - \sqrt{\mu_r \epsilon_r})$$

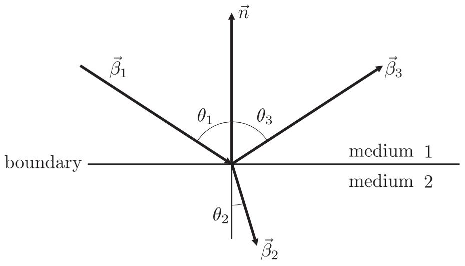


Figure 1.1: Refraction and reflection of microwaves at media boundary

Equation (1.3) can be written in general form for any direction of propagation determined by *direction vector* $\vec{\beta} = \beta\,(e_x, e_y, e_z)$, where e_x, e_y, and e_z are normalized projections of $\vec{\beta}$ on to axis x, y, and z, correspondingly [7]:

$$\vec{E}(\vec{r}, t) = \vec{E}_0 \cos(\phi_0 + 2\pi f t - \vec{\beta} \cdot \vec{r}\,) \tag{1.5}$$

In the latter expression a dot product $\vec{\beta} \cdot \vec{r}$ gives a projection of $\vec{r}$ onto $\vec{\beta}$, i.e. a delay of propagation of microwaves along the direction given by $\vec{\beta}$.

1.2.2 Microwaves Refraction and Reflection

In the previous section we have introduced the mechanism of propagation of microwaves in the medium. It was shown that the velocity of propagation depends on its permittivity and permeability. In this section we will discuss two other mechanisms which describe the behavior of microwaves at the boundary of two media.

Let there be two media which have different permittivities ϵ_{r1} and ϵ_{r1}. Medium 1 is assumed to be more transparent then medium 2, i.e. $\epsilon_{r1} < \epsilon_{r2}$ (see Figure 1.1). Microwaves propagate in the medium 1 in direction $\vec{\beta}_1$ towards the boundary. An *incident angle* between direction $\vec{\beta}_1$ and normal $\vec{n}$ to the boundary is θ_1.

When microwaves arrive at the boundary, they split into two parts, namely reflected and refracted ones. The first part reflects back (*reflection phenomenon*) into medium 1 at direction $\vec{\beta}_3$. Analogously with *Fresnel law* [8, p.18] the *reflection angle* θ_3 is equal to the incident angle θ_1.

Since properties of the media are different, the second part diffracts at the boundary. It propagates in medium 2 along direction $\vec{\beta}_2$. That phenomenon is known as *refraction* . A *refraction angle* θ_2 is related to the incident angle θ_1 by *Snell's* law [9, p.435]:

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{\sqrt{\epsilon_{r2}}}{\sqrt{\epsilon_{r1}}} \tag{1.6}$$

Both parts (reflected and refracted) have particular electric field intensities after the interaction at the boundary. We do not give the mathematical definition of the intensities. This information can be found in the microwave literature, see for reference [4–6, 10–13]. We only keep in mind that the field intensity depends on transparency of the medium. The more transparent the medium, the higher is the intensity of the component.

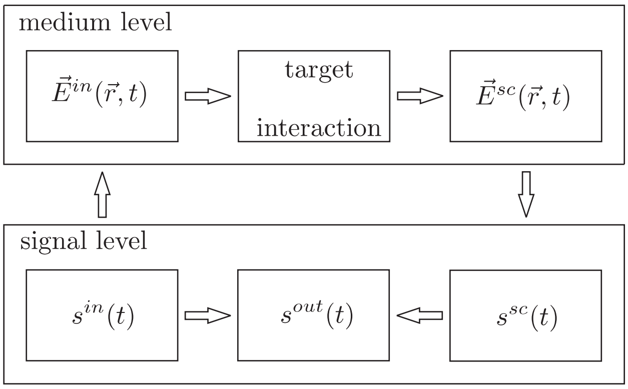


Figure 1.2: Two-level radar operational system

1.3 Continuous Wave (CW) Radar

Generally, *radars* are understood as electrical devices which are used to produce microwaves, for reference see [14]. Radars function by radiating microwaves and detecting the returned echo from reflecting objects. Radars differ by the form of radiated microwaves. The choice of form depends on desired *information* about the target (i.e. target properties to be evaluated) such as size, spatial location, material properties, etc. Detailed description of different types of radars such as *pulse* radars, MTI radars, etc. is given in [15]. In this thesis we investigate an application of *continuous wave* (CW) radars for non-destructive testing.

1.3.1 CW Radar Principle

Let us describe the principle of CW radar. A radar *operational system* is represented in Figure 1.2. A time-varying signal[1] s^{in} (we use term *"incident"* signal) is applied to the input of the radar. This causes activation of the electromagnetic field E^{in} which is defined at spatial position $\vec{r} \in \mathbb{R}^3$ and time t, see equation (1.5). The field E^{in} propagates towards the reflecting object (*target*) and interacts with it. This interaction establishes the reflected (*scattered*) electromagnetic field E^{sc} which propagates back to the radar. While radar is receiving scattered field E^{sc} it excites the time-varying signal s^{sc}. According to physics CW radar produces an output signal s^{out} by *mixing* both incident s^{in} and scattered s^{sc} signals as given below

$$s^{out}(t) = (s^{in}(t) + s^{sc}(t))^2 \tag{1.7}$$

A radar operational system in Figure 1.2 is split into a *medium level* and a *signal level* . On the medium level the information about targets is encoded in the incident $\vec{E}^{in}$ and scattered $\vec{E}^{sc}$ electromagnetic fields. Inside the radar both of the fields are transformed into electrical signals (or voltage). By measuring and further processing of these signals we retrieve specific information about the target. For the CW radar we define the incident signal as

$$s^{in}(t) = A^{in} \cos(2\pi f^{in} t + \varphi^{in}) \tag{1.8}$$

[1] under a term time-varying signal we understand an electrical signal of varying voltage

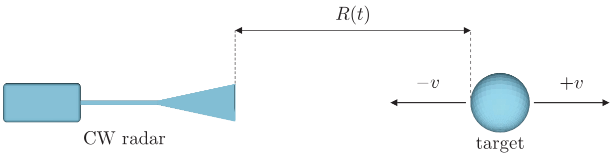


Figure 1.3: Scheme of the Doppler experiment

In equation (1.8) an amplitude A^{in} is proportional to the intensity of the incident electric field E^{in}; f^{in} and φ^{in} are the transmission frequency and the initial phase, respectively. In this work we use the CW radar with $f^{in} = 24\,\text{GHz}$.

The scattered signal s^{sc} depends on the properties of the target. In general case s^{sc} is referred to be some real-valued analog signal, i.e. $s^{sc} : \mathbb{R} \to \mathbb{R}$.

In case of CW radar the output signal $s^{out} : \mathbb{R} \to \mathbb{R}$ is the only one provided for analysis. In the following sections we discuss which information about the target can be extracted from s^{out}. We will also consider possible applications of CW radar for non-destructive testing needs.

1.3.2 Doppler Effect

The *Doppler effect* was first explained in 1842 by Christian Doppler. The Doppler effect is the shift in frequency of a sound wave that is perceived by an observer. The frequency shift happens with the moving of either the sound source, or the observer, or both.

The Doppler effect takes place for all types of radiation such as light, sound, microwaves etc. In order to measure the Doppler effect in microwaves CW radars were invented. In this work we introduce the Doppler effect on the signal level, i.e. in terms of electrical signals, see Section 1.3.1. We often use expression "signal was sent" or "signal was received" keeping in mind that the sending and receiving of an electromagnetic field is meant.

The basic idea of the Doppler effect in microwaves is represented in Figure 1.3. The CW radar does not change its position whereas the target moves. We define the speed of the target v to be a constant. This ensures linear increasing or decreasing of the distance R between the radar and the target in time. We also refer to it as to *radar-target distance* :

$$R(t) = R_0 \pm v(t - t_0), \tag{1.9}$$

where time t_0 is an *initial time*, i.e. the time when the target begins its movement; R_0 is the initial radar-target distance. In equation (1.9) approaching of the target is given by a "minus" sign before v. Analogously, "plus" sign denotes moving of the target away from the radar.

In the following equation we define the *propagation time* t^{pr} that microwaves need to travel towards the target and back at every distance R

$$t^{pr}(t) = \frac{2\,R(t)}{c_0} \tag{1.10}$$

Since we carry on the experiment in the air, the microwaves propagation speed is equal to the *speed of light* in the vacuum, i.e. c_0.

According to the scheme of the experiment, the scattered signal s^{sc} is a time-delayed version of the incident signal s^{in}. The time delay is equal to the propagation time t^{pr}. The amplitude A^{sc} of signal s^{sc} differs from the amplitude of signal s^{in} due to propagation and interactions losses (see Sections 1.2.1 and 1.2.2). The amplitude A^{sc} also depends on radar-target distance R, geometrical shape and material properties of the target. For simplicity we introduce A^{sc} by means of some function f as

$$A^{sc}(t) = f(A^{in},\, R(t),\, \epsilon_t), \tag{1.11}$$

where ϵ_t is a permittivity of a target material. The complete definition of amplitude A^{sc} can be found in literature given in Section 1.2.2. By using definitions (1.8), (1.10) and (1.11) we define the scattered signal as

$$s^{sc}(t) = A^{sc}(t)\cos(2\pi f^{in}\,(t - t^{pr}(t)) + \varphi^{in}). \tag{1.12}$$

Let us assume that in (1.10) the initial time $t_0 = 0$ then

$$\begin{aligned} s^{sc}(t) &= A^{sc}(t)\cos\left(2\pi f^{in}\left(t - \frac{2(R_0 \pm vt)}{c_0}\right) + \varphi^{in}\right) \\ &= A^{sc}(t)\cos\left(2\pi\left(f^{in} \mp f^{in}\frac{2v}{c_0}\right)t + \varphi^{in} - \frac{2\pi f^{in}}{c_0}2R_0\right) \\ &= A^{sc}(t)\cos(2\pi f^{sc}t + \varphi^{sc}), \end{aligned} \tag{1.13}$$

where frequency f^{sc} and initial phase φ^{sc} of the scattered signal are given as:

$$\varphi^{sc} = \varphi^{in} - \frac{2\pi f^{in}}{c_0}\cdot 2R_0 \tag{1.14}$$

The frequency f^{sc} is given as:

$$f^{sc} = f^{in} \mp f^{d} = f^{in} \mp f^{in}\left(\frac{2v}{c_0}\right) \tag{1.15}$$

Equation (1.15) introduces the nature of the Doppler effect. The target approaching or moving away from the radar causes increasing or decreasing in the received signal frequency f^{sc}. The frequency f^{d} is called *Doppler shift* . It depends on the transmitted frequency f^{in} and the radar speed v.

In the experiment represented in Figure 1.3 *Doppler frequency* (Doppler shift) f^{d} stays constant because speed v is constant too. Generally, both the radar and the target change their spatial locations. They also can move in different directions having different non-constant speeds. In that case the distance between the radar and the target is a non-linear function of time. In order to define the Doppler frequency the time derivative of R is used. A more general definition of the Doppler frequency f^{d} is

$$f^{d}(t) = \mp f^{in}\frac{2}{c_0}\frac{\partial R}{\partial t} = \mp\frac{2}{\lambda^{in}}\frac{\partial R}{\partial t}, \tag{1.16}$$

where λ^{in} is called *wavelength* . In equation (1.16) positive Doppler frequency means approaching of the radar and the target, i.e. R decreases. If R increases, then the Doppler frequency f^{d} is negative.

Thus, in the general case a Doppler frequency is the time-varying function defined by equation (1.16).

From the proof given in [16, p. 522] follows that a time-varying signal s of some time-varying frequency f can not be defined as

$$s(t) = \cos(2\pi f(t)\, t) \tag{1.17}$$

because this leads to the physical inconsistency. The equation (1.17) holds only if $f(t)$ has the same value for any t. In order to overcome the physical inconsistency an integral of f is required, i.e.

$$s(t) = \cos\left(2\pi \int\limits_0^t f(t)\; dt\right). \tag{1.18}$$

Since the Doppler frequency f^d is also a time-varying function let us modify the definition of the scattered signal s^{sc}. We substitute equation (1.15) into (1.13) and take the integral of time-varying f^d as it was shown in (1.18) so that we have

$$s^{sc}(t) = A^{sc}(t)\, \cos\left(2\pi f^{in} t \,\pm\, 2\pi \int\limits_0^t f^d(t)\; dt + \varphi^{sc}\right) \tag{1.19}$$

Comparing equations (1.13) and (1.19) we note that the Doppler frequency is not a constant rather the integral over the time. This integral introduces the *instantaneous frequency* phenomenon which we discuss in details in Chapter 4.

1.3.3 Measurement of the Doppler Effect

In the previous sections we gave definitions of incident (or sent) s^{in} and scattered (or received) s^{out} signals that CW radar is operating with. Practically it is difficult to measure both s^{in} and s^{sc}. Usually, in hardware implementation of CW radars the only signal which is available to be acquired is the radar output s^{out}, see Section 1.3.1. In order to derive s^{out} we substitute equations (1.8) and (1.19) into (1.7) and apply some trigonometric transformations:

$$s^{out}(t) = A(t)\, \cos\left(\pm 2\pi \int\limits_0^t f^d(t)\; dt + \varphi\right) + \Lambda \tag{1.20}$$

Derivation of equation (1.20) is represented in Appendix A.0.2. In (1.20) the first term is also called Doppler term. It represents harmonic oscillations of the time-varying Doppler frequency f^d, some constant phase φ and time-varying amplitude A. Currently we are not interested in the behavior of A. Instead, we assume amplitude to be a real function such that $A : \mathbb{R} \to \mathbb{R}$. The second term Λ represents the multi-component sum of cosines. We discuss this term in the in the following.

In practical applications the speed of both the radar and the target is much lower than the speed of light c_0. This ensures the Doppler frequency f^d to be much lower than the transmitted frequency f^{in} (see equation (1.16)), i.e. for all t holds

$$f^d(t) \ll f^{in} \pm f^d(t) \tag{1.21}$$

We remind that the transmitted frequency f^{in} is in GHz range whereas the doppler frequency f^d is only a few tens of hertz.

As we can see, equation (1.20) is a mixture of harmonic oscillations of very high (Λ-term) and very low (Doppler term) frequencies. When s^{out} is acquired, its high frequency part Λ will be cut off. We discuss the acquisition (*measuring*) of the Doppler signal s^{out} in detail in Chapter 2. Using the consideration presented above we derive the output of the CW radar as the following

$$s(t) = A(t)\cos\left(\pm 2\pi \int\limits_0^t f^d(t)\,dt + \varphi\right) \tag{1.22}$$

In the latter expression in some cases we can neglect φ because it remains constant since only depends on send frequency f^{in} and initial distance R_0 (for reference see equations (1.14) and (A.5)). The ± sign can be also omitted since the cosine is an even function.

Equation (1.22) introduces the main idea of CW radars which we will discuss in details in Chapter 2. For now we note that CW radars are incapable to measure any range information about the target. The only information which can be derived from the measured signal $s : \mathbb{R} \to \mathbb{R}$ is the Doppler frequency $f^d : \mathbb{R} \to \mathbb{R}$ and amplitude A. The alternations of the frequency and the amplitude appear when the distance between the target and the radar is not constant.

A simple operational principle of CW radars allows their production using non-expensive components. Because of that reason CW radars become cheap to be manufactured what makes them attractive for NDT. Otherwise, in many applications only evaluation of Doppler frequency is insufficient for solving a given problem in a proper way. This, certainly, turns CW radars to be not widely used in NDT.

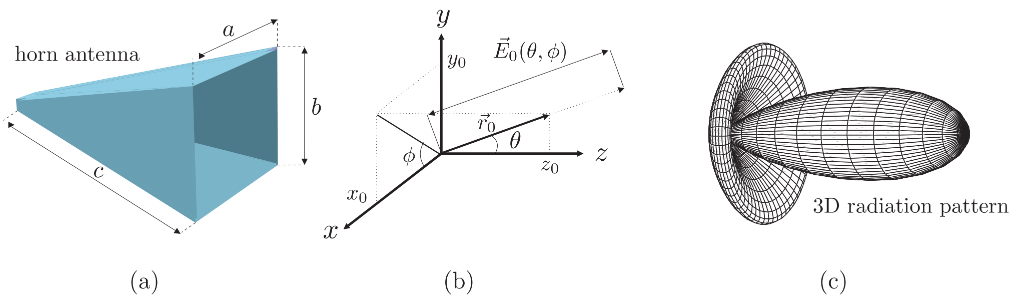
The aim of this work is to test whether CW radars are applicable for microwave non-destructive testing in order to characterize test objects.

In the next section we shortly present radar equipment such as antenna and waveguide.

1.4 Radar Antenna

The role of an *antenna* is to provide coupling between microwaves in a free space and *transmitted* or *received* microwaves radiated by the radar. In many applications the antenna serves both as a *transmitter* and as a *receiver*. Antennas are also used to concentrate microwaves in a particular direction. Antenna theory offers various antenna designs. Choice of an antenna depends on its area of application and price [17].

In microwave NDT *horn antennas* are widely used, see Figure 1.4(a). *Physical parameters* of a horn antenna are width a, height b, and length c. Parameters a and b determine *antenna aperture* A which is a surface of size $a \times b$ where microwaves



(a) (b) (c)

Figure 1.4: Radar antenna

pass through. The aperture lays in the plane xy. We establish z to be a direction of propagation of microwaves, see orientation of the coordinate system in 1.4(a).

Microwaves do not propagate through the aperture in *omnidirectional* manner. The field intensity changes over the aperture surface. The antenna *radiation pattern* determines variation of the electric field intensity. Let us assume the electric field $\vec{E}(\vec{r}, t)$ is defined by the equation (1.5). Its amplitude $\vec{E}_0$ is not a constant but a function of direction given by the unit vector $\vec{r}_0 = \vec{r} \backslash \|r\|_2 = \{x_0, y_0, z_0\}$. For convenience the dependency $\vec{E}_0(\vec{r}_0)$ is expressed in *spherical coordinates* [18, pp.102-111] by means of triple $\{r_0, \theta, \phi\}$, where the radius r_0, *zenith* θ, and *azimuth* ϕ are given as

$$
\begin{aligned}
r_0 &= \|\vec{r}_0\|_2, \\
\phi &= \arctan\left(\frac{y}{x}\right), \\
\theta &= \arccos\left(\frac{z}{r_0}\right).
\end{aligned}
$$

Since $\vec{r}_0$ is a unit vector we define the electric field amplitude as a function of zenith and azimuth i.e. $\vec{E}_0(\theta, \phi)$, where $0 \leq \theta \leq \pi/2$ and $0 \leq \phi \leq 2\pi$, see Figure 1.4(b). An example of a 3D radiation pattern is shown in Figure 1.4(c). From a radiation pattern the *spatial resolution* or *lateral resolution* of the radar can be derived. The radiation pattern of an antenna is delivered by its manufacturer.

Let us introduce the most important parameters of a radiation pattern. We build 2D radiation pattern $\vec{E}(\theta, \phi)$ (see Figure 1.5), such that $0 \leq \theta \leq \pi/2$ and $\phi = 0$. For convenience, the radiation pattern is *normalized* such that its highest intensity is 1. Various parts of a radiation pattern are referred to as *lobes*. A lobe is defined as a part of a radiation pattern bounded by weak intensity. Lobes having highest intensity and oriented in the direction of propagation are called *major lobes*. *Minor lobes* are lobes of low intensity, see Figure 1.5. They deviate from the main direction of propagation.

Narrowness of the main lobe is defined by angle θ_{hw} that is determined from the half-value of the field intensity (Figure 1.5). Obviously, the narrower the major lobe is the higher is spatial resolution. In antenna design developers try to get an antenna with a narrow major lobe and low intensity of minor lobes.

The second most important parameter of an antenna is the *antenna gain* G. It determines the ability of an antenna to concentrate microwaves in a particular

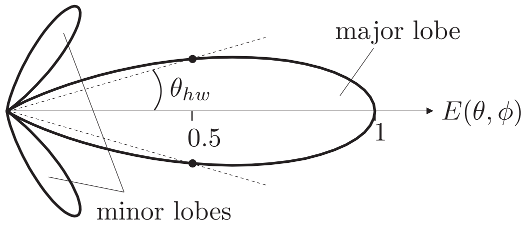


Figure 1.5: 2D radiation pattern

direction. Gain G also introduces sensitivity of an antenna to microwaves incident from a specific direction. Gain G depends on the antenna aperture A and microwave wavelength λ

$$G = \frac{4\pi A}{\lambda^2} \tag{1.23}$$

1.4.1 Radar Cross Section

Definition 1.4.1 *The target* ***radar cross section*** (RCS) *is the effective echoing area of a target which isotropically radiates towards a radar all of its incident power at the same radiation intensity* [19].

We denote RCS as σ. Its mathematical formulation in terms of *power densities* is defined in [19]:

$$\sigma = 4\pi R^2 \frac{w_e}{w_i}, \tag{1.24}$$

where w_e is the echo power density seen at the radar; and w_i is the echo power density at the target. Definitions of w_e and w_i are given in the following section. Equation (1.24) can be rewritten using the corresponding radiation field intensities E_e and E_i

$$\sigma = 4\pi R^2 \frac{|E_e|^2}{|E_i|^2} \tag{1.25}$$

1.4.2 Radar Equation

The power density w_i at the target is defined as the radar initial power P_t (i.e. transmitted echo power) which is emitted into space through antenna gain G. The target is located in front of the antenna at distance R:

$$w_i = \frac{P_t G}{4\pi R^2 L}, \tag{1.26}$$

where L introduces total system loss[2].

The received echo power P_r is defined from power density of the radar w_e multiplied by antenna aperture A. By using (1.23) and substituting equation (1.26) into (1.24) we derive P_r [20]:

$$P_r = w_e A = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L}. \tag{1.27}$$

[2] system loss is a term that normally includes transmission line loss, propagation loss, receiving-system loss etc., for more information see [19, page 34]

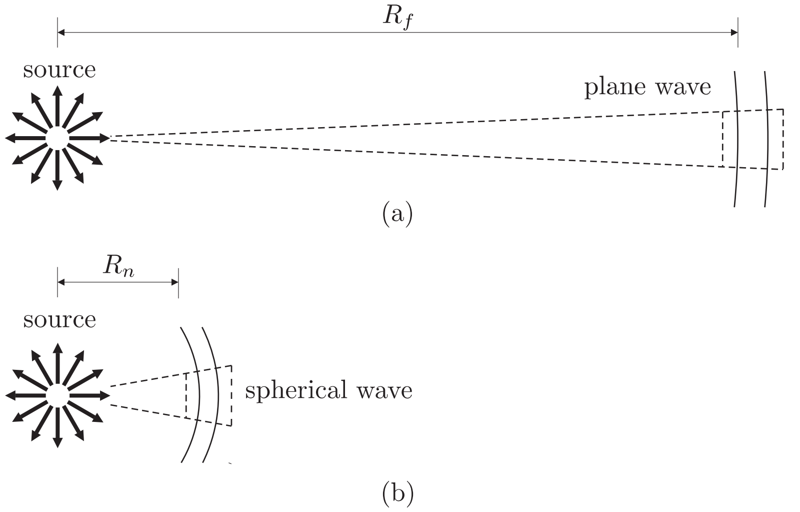


Figure 1.6: Far field and near field regions

Equation (1.27) is valid for the *far-field* region. In radar literature the far field is defined as a region of the electromagnetic field of an antenna where the angular field distribution is essentially independent on the distance from the antenna [17]. In many practical applications the far field is preferable since the antenna radiation pattern is well formed and usually has one major lobe, see Section 1.4. This increases the scattered field intensity and improves the signal to noise ratio in the measured signal. In the far-field region microwaves can be approximated by *plane waves*. A flow chart which introduces the far field region is represented in Figure 1.6(a). In literature the distance R_f where the far-field region starts is given as

$$R_f = \frac{2D^2}{\lambda},$$

where $D = \max(a, b, c)$ is a maximal dimension of the antenna.

There is also a number of applications where the *near-field* region is used. In that region angular field distribution depends upon the distance from the antenna, see Figure 1.6(b). In the near field the radiation pattern of the antenna is smooth and does not form lobes. Advantage of the near field is high *spatial resolution* since the antenna is close to the specimen. The range of the near-field region is defined as

$$R_n = 0.62\sqrt{\frac{D^3}{\lambda}} \tag{1.28}$$

In the near-field region microwaves can be approximated by *spherical waves*.

Often in practice it is required to find distance $R \in [R_n : R_f]$ such that R is close as possible to R_n. This ensures high spatial resolution (since microwaves resemble properties of microwave in near field region) and also allows the use of many attractive properties of plane waves (since microwaves can be also approximated by the plane waves). Some of these properties we will introduce in the following chapters.

A common approach to find distance R is to measure the field intensity in different points, step by step at the increasing distance between the radar and the specimen. The desired distance R is found when the measured intensity starts to exhibit an R^{-4} - law. Generally, this is introduced in equation (1.27) and can be represented as the following relation

$$|E(R)|^2 \propto \frac{1}{R^4}, \tag{1.29}$$

where $E(R)$ is the measured intensity at the distance R.

Chapter 2

Doppler System

2.1 CW Radars in Microwave NDT

Nowadays CW radars are widely used in different applications such as:

- intrusion alarms,
- automatic door openers,
- speed and motion detection, traffic control etc.

All these areas of application have the same in common, namely presence of motion. For example, intrusion alarm systems observe the interior of a room. When a burglar moves, the Doppler effect appears and alarm turns on. Automatic door openers operate in a similar way. The police uses CW radars to measure movement speed of a vehicle while it is driving. In that case the Doppler frequency shift is proportional to the speed.

Many attractive features make CW radars popular. Some of them are low cost, small size, high sensitivity etc.

CW radars do not give any range information about the target. This reduces areas of application of CW radars in microwave NDT. Generally, any application that includes detection of the distance to the target is out of consideration. For example, such problems as plastic thickness determination on tubes [21] or level measurements [22, 23] seem unlikely to be solved.

Let us discuss two main mechanisms that make possible application of CW radars in microwave NDT. The first mechanism is causing the Doppler effect by geometrical irregularities on the object surface. The second mechanism is alternation of the object electrical permeability ϵ_r. Both mechanisms cause variation of the electric field intensity. Among others possible areas of application of CW radars in NDT are detection of

- metal surface cracks,
- hidden interstices in non-conductive or semi-conductive materials,
- impact damages of the materials (e.g. **G**lass **F**iber **R**einforced **P**olymer),
- moisture and knots in wood

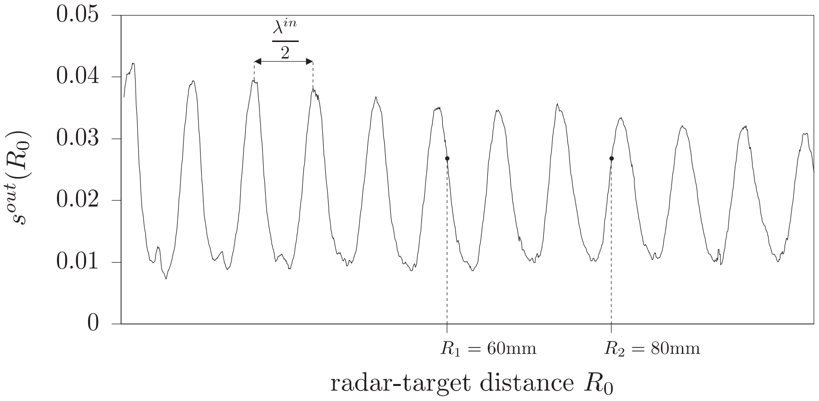


Figure 2.1: Raster measurement approach (distance variation measurements)

2.2 Raster Measurements

The most popular technique which is used to perform measurements is *raster measurements*. The idea of that technique is simple. A radar moves stepwise in a particular direction above a specimen. The size of a step δ_x is determined by the user. At every position the radar stops and the output signal of the radar is acquired. When the acquisition finishes, the radar moves to the next point. The procedure ends as soon as all points are measured. Raster measurements do not depend on a motion speed v of the radar. This implies for CW radars the Doppler frequency f^d to be zero, see equation (1.16). From equation (1.20) we derive the output of the CW radar in case of the raster measurement as follows:

$$s^{out}(t) = A(t)\cos(\varphi) + \Lambda. \tag{2.1}$$

In Appendix A.0.3 we show that under condition of zero Doppler frequency the signal s^{out} depends on radar-target distance R_0 but not on time. Such a modification of equation (2.1) is given below:

$$s^{out}(R_0) = A(R_0)\cos\left(2\pi\frac{2f^{in}}{c_0}R_0\right) + \mathrm{C}(R_0), \tag{2.2}$$

where f^{in}, c_0, R_0 are the radar transmitted frequency, the speed of light and the radar-target distance, respectively. In equation (2.2) the function C represents the offset of the signal s^{out} which also depends on R_0.

The only dependence of s^{out} on R_0 may cause difficulties in defect detection. Let us consider it by an experiment. As a specimen we use a flat metal plate. The surface of the plate is free from any defects. We acquire the output signal of the radar for different distances between the radar and the specimen such that $R_0 \in [30\text{mm} : 100\text{mm}]$ with step $\delta_x = 0.05$mm. Figure 2.1 presents how s^{out} develops. It is a harmonic cosine oscillation with a period $\lambda^{in}/2$, see Appendix A.0.3. We observe that amplitude of the signal s^{out} decreases very slowly, whereas s^{out} oscillates fast. Thus, at different radar-target distances the amplitude may be the same. This causes ambiguity in distance detection in the following sense. For example, in Figure 2.1 value $s^{out}(R_1)$ is almost equal to $s^{out}(R_2)$ even if the distance difference is about 20mm. From the experiment given in Figure 2.1 we also conclude that C changes very little with radar-target distance R_0. It offsets the signal s^{out} at some constant value.

Insensibility of CW radars to respond to distance variation makes raster measurements not really efficient in microwave NDT. Small defects such as cracks or impact damages can be lost among other reflections from the surface. A possible reason of it can be, for example, a small radar cross section of the defect, see Section 1.4.

In some practical applications it is difficult to keep a constant distance between the radar and the specimen. This may cause false defect alarms.

In the next section we will consider another measurement technique called *continuous measurements*. That technique sufficiently decreases most of the drawbacks of CW radars outlined in this section.

2.3 Continuous Measurements

Two ways (or approaches) to perform continuous measurements are under consideration. The difference between them is that either CW radar is *perpendicular* or *oblique* by angle α to the analyzed surface, see Figure 2.2(a) and 2.2(b). In both cases the radar moves in a given direction with a constant speed v so that a track of movement is a straight line. We call it the *line of scan* .

We will compare both approaches by analyzing the radar response on the same kind of defect. The defect is assumed to be a *point scatterer* (or point reflector) located on the line of scan. Its main property that it reflects an incident signal back independently on the microwave incidence angle i.e. the point scatterer has a constant RCS. Motion of the radar in a particular direction is identical to motion of the defect in the opposite direction. For both the perpendicular and oblique approaches we determine sets of *visible* spatial positions, Ω' and Ω'', respectively, when a defect is visible to the radar.

$$\Omega' = \{x \mid x'_1 \leq x \leq x'_2\} \quad \text{and} \quad \Omega'' = \{x \mid x''_1 \leq x \leq x''_2\}, \tag{2.3}$$

where boundaries x'_1, x'_2 and x''_1, x''_2 are determined by the radar altitude h and narrowness of the radiation pattern θ_{hw}, see Figure 2.2(c) and 2.2(d). Choice of h will be discussed in Section 2.6. Currently we assume it to be some non-negative value.

In both cases (perpendicular and oblique) the distance between radar and defect does not change linearly. We introduce a function $L : \mathbb{R} \rightarrow \mathbb{R}$ which determines variation of the radar-defect distance as

$$L(x) = \sqrt{(x^r - x)^2 + h^2},$$

where $x \in \Omega'$ or $x \in \Omega''$ in the perpendicular or oblique case, respectively.

The *Doppler spatial phase* $\varphi^d : \mathbb{R} \rightarrow \mathbb{R}$ determines the number of periods of the transmitted frequency f^{in} on the way towards the defect and back to the radar

$$\varphi^d(x) = -\frac{2L(x)}{\lambda} \tag{2.4}$$

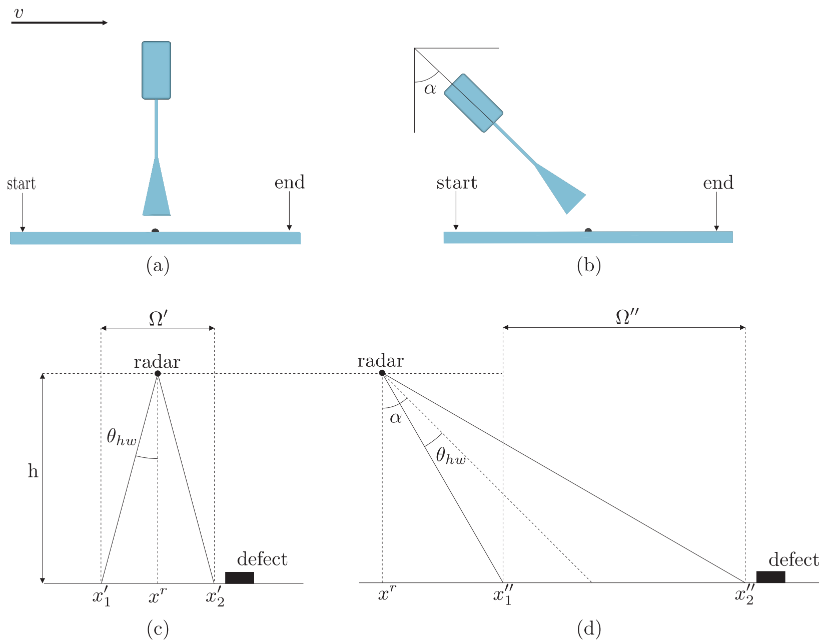


Figure 2.2: Continuous measurement approach: (a) scheme of perpendicular measurement; (b) scheme of oblique measurement; (c) perpendicular measurement flowchart; (d) oblique measurement flowchart

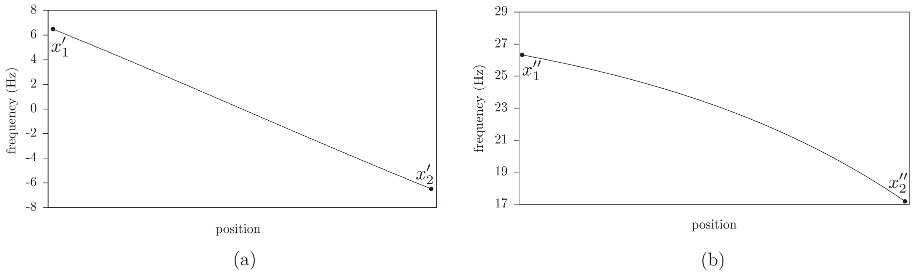


Figure 2.3: Simulated Doppler frequency: (a) perpendicular case; (b) oblique case

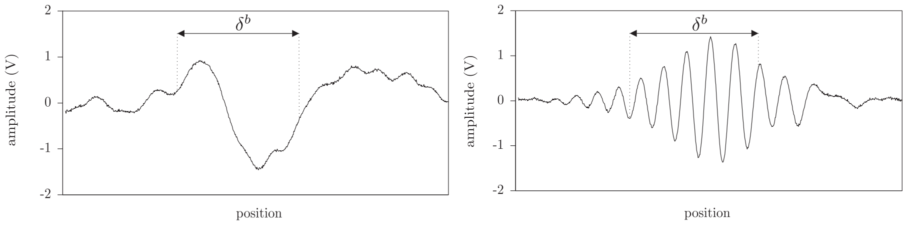
The minus sign in (2.4) is used to emphasize that distance L as well as spatial phase φ^d decrease when the defect approaches the radar. By applying the spatial derivative to (2.4) we derive the time-dependent Doppler frequency f^d as

$$f^d(t) = \frac{\partial \varphi^d}{\partial t} = \frac{\partial \varphi^d}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial \varphi^d}{\partial x}v \tag{2.5}$$

where speed v is represented as the time derivative of spatial coordinate.

Equation (2.5) introduces the correlation between physical radar adjustments (its altitude and radiation pattern), radar motion, and the defect position. Later we will use this equation to understand and explain advantages and disadvantages of perpendicular and oblique measurement approaches.

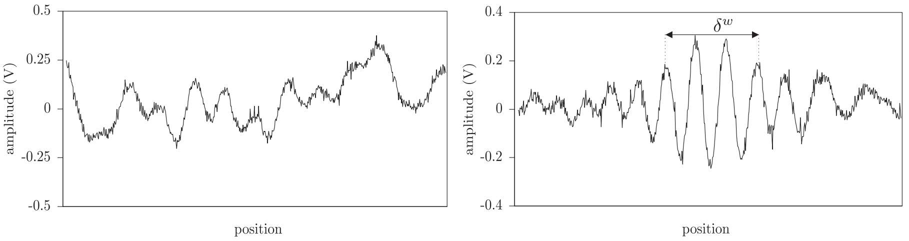
A simulated Doppler frequency computed by (2.5) in the perpendicular and oblique cases is shown in Figures 2.3(a) and (b), respectively. As we can see the frequency f^d in the perpendicular case is much lower than in the oblique case



(a)

(b)

Figure 2.4: First part of the experiment (metal ball): (a) signal s_1^b, perpendicular case; (b) signal s_2^b, oblique case;



(a)

(b)

Figure 2.5: Second part of the experiment (metal washer): (a) signal s_1^w, perpendicular case; (b) signal s_2^w, oblique case

even if the radar altitude h and speed v are the same. From this follows that the number of oscillations of the measured signal s, defined in (1.22) is essentially reduced. We suppose that under equal measurement conditions in the oblique case we receive more information about the defect than in perpendicular case.

In order to check the theoretical supposition mentioned above we perform an experiment. In the first part of the experiment we will compare an approximate value of the Doppler frequency in both cases. In order to ensure high echo power of the reflected signal during the whole scan we use a relatively large defect. It is a metal ball with the diameter of 12mm. The ball is placed in the middle of the line of scan. Measured signals in perpendicular and oblique cases (s_1^b and s_2^b, respectively) are presented in Figure 2.4(a) and (b). In both cases presence of the defect can be detected. Without further signal processing we can see that s_1^b has fewer oscillations than s_2^b. This can be explained in terms of the Doppler frequency. Obviously, s_2^b has the higher Doppler frequency than s_1^b what perfectly matches to the simulation results presented in Figures 2.3(a) and (b).

Let us determine an approximate ratio of the Doppler frequencies in both cases. We determine a fragment of s_1^b which contains, approximately, one period. In Figure 2.4(a) we denote it as δ^b. The defect is placed nearly in the middle of the interval δ^b. In Figure 2.4(b) we observe about five full periods which are located inside δ^b. Thus, the Doppler frequency of s_2^b is about five times higher than the frequency of s_1^b. It also matches to simulations given in Figures 2.3(a) and (b).

From the first part of the experiment we conclude that in the oblique case we receive more information about defect than in the perpendicular case. In Chapter 5 we utilize this quality of the oblique approach to increase spatial radar resolution.

Let us introduce the second part of the experiment that illustrates the advantages of the oblique approach in detection of small detects. The defect we use is a 1mm thin washer. Its radius is about 6mm. We perform both perpendicular and oblique measurements. Acquired signals s_1^w and s_2^w are represented in Figure 2.5(a) and (b) respectively. By analyzing signal s_1^w we can see that the presence of the defect can not be detected. This can be explained as follows. An antenna is perpendicular to the surface which reflects microwaves back. The washer also reflects microwave back. Since the defect is thin, the reflection from the surface is higher than the reflection from the defect. Therefore the contribution of the defect into the resulting signal is negligible.

In the oblique case (Figure 2.5(b)), reflection from the surface is not captured by the radar because of the slope. However, the defect still reflects microwaves back to antenna since is has sharp edges. While the reflection exists, the Doppler effect appears. The area of signal s_2^w which can be used to detect the defect is labeled as δ^w.

From the experiment discussed above we conclude:

- The theoretical model to explain and analyze the Doppler effect given in equation (2.5) is viable since it describes the experiment well.
- Oblique measurements retrieve more information about the object than perpendicular measurements.
- Oblique measurements are more sensitive to small defects.

2.4 Doppler Measurement System

In order to perform experiments based on the Doppler effect in microwaves, a prototype system has been developed and built. It uses CW radars for measurements. We refer to this system as to the *Doppler measurement system* or simply the *measurement system*. These are main requirements the measurement system have to satisfy:

- stability
- mobility
- high measurement speed
- ability to scan surfaces

Since the Doppler effect appears only with the motion, all the parts of the measurement system have to maintain *stability* except the one that moves. In order to keep the measured signal undamaged, any spontaneous shakings of the measurement system have to be prevented.

A term *mobility* means the support of a universal interface to connect different CW radars to the measurement system. The universal interface includes *power supply lines*, *output signals*, *control signals* etc. The mobility also includes easy adjustment of the spatial position of the radar.

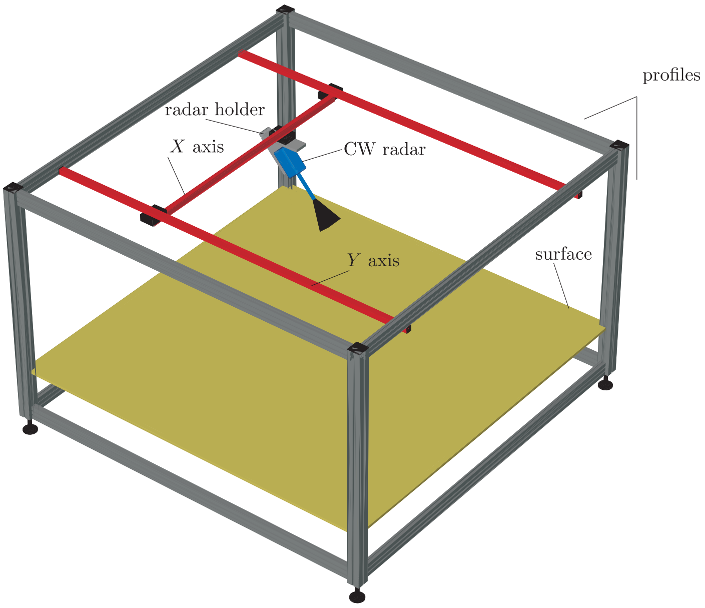


Figure 2.6: Doppler Measurement system

An ability to perform *fast measurements* is one of the main tasks of the measurement system. Speed of the measurements depends on the power of motion motors. It is also important to maintain a constant speed while the output signal is being acquired.

In this work we have developed the *Doppler imaging* as a new imaging technique in microwave NDT. Since 2D images are more informative then 1D signals the *surface scan* ability of the measurement system is required.

An accurate model of the Doppler measurement system is represented in Figure 2.6. Its *skeleton* is made from aluminium profiles which are joined to each other. In order to ensure stability of the skeleton every profile is additionally joined to all its neighbours by means of corner fastenings (not showed in Figure 2.6). At the top of the skeleton two positioning axes are fixed (axis X and Y). The CW Radar can move along axes with motion speed v. The object to be tested is placed on the surface located at the bottom of the skeleton.

In order to adjust the spatial position and orientation of the radar a *radar holder* is used, see Figure 2.7. The holder is fastened to the positioning axis X at its middle. The altitude of the radar h is adjusted by rotation of a *knob* which is placed at the top of the radar holder. The range of a *incidence angle* α varies from 0 to 90 degrees. The radar holder is equipped with scales for both h and α to adjust them precisely. It is also possible to rotate the radar around the z axis by using of a *turntable*. This may be helpful if there is a need to change the direction of scan. The radar is fastened on a *holding plate*. It was developed as a multipurpose part so that different radar types can be easily installed.

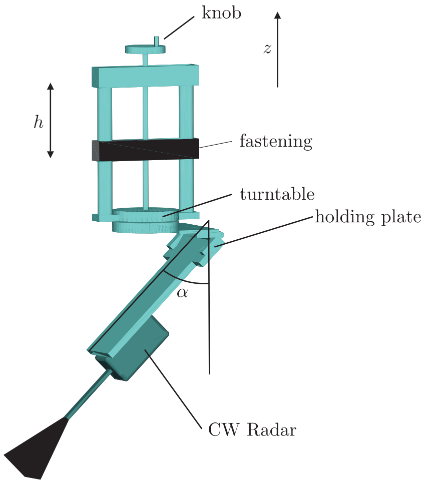


Figure 2.7: Radar Holder

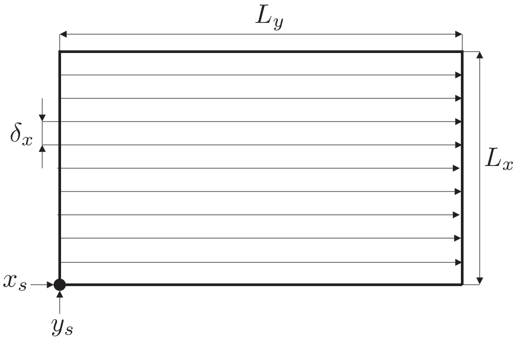


Figure 2.8: Motion Pattern

2.4.1 Motion Control Unit

The motion control of axes X and Y is performed by the computer. The axes motors have an intelligent interface. Through this interface the motors can be programmed to execute a particular sequence of motions. That sequence is denoted as a *motion pattern*.

In order to perform 2D surface scan we use a motion pattern which is shown in Figure 2.8. The area to be scanned has size $L_x \times L_y$. While the radar is being moved from start point (x_s, y_s) to end point $(x_s, y_s + L_y)$, the output signal is acquired. When measurement ends, the radar moves to position $(x_s + \delta_x, y_s)$. From $(x_s + \delta_x, y_s)$ to $(x_s + \delta_x, y_s + L_y)$ the measured signal is acquired again. It repeats until the last measurement from $(x_s + L_x, y_s)$ to $(x_s + L_x, y_s + L_y)$ is done. As we move the surface of interest is scanned line per line. Movement of the radar along axis Y is been performed at a constant speed. The size of the step, i.e. δ_x, depends on the size of defects and scan quality demands. In practice in order to choose δ_x a trade-off between quality and speed of measurement is needed.

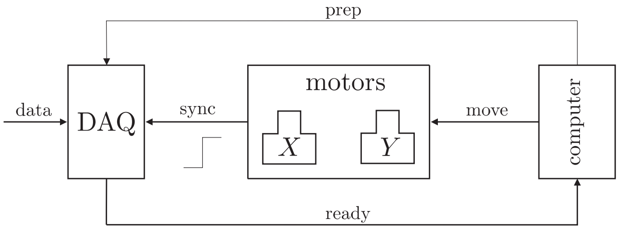


Figure 2.9: Data acquisition synchronization chart

2.4.2 Data Acquisition

While the radar is moving along a line of scan the Doppler signal is being acquired, (for reference see Section 2.4.1). Conversion of an analog sinal into its digital representation is given by the sampling procedure (see Definition 1.1.10). In order to perform a surface scan it is important to start data acquisition and radar motion at the same time for all line scans. It prevents the measured data from being shifted. To fulfill these requirements we use the *synchronization* technique represented in Figure 2.9.

The computer controls the data acquisition device (DAQ) and motors (controlling axis X and Y) of the measurement system. Before a line scan is performed the computer sends a preparation command *prep* in order to switch DAQ into the *waiting loop*. In that state DAQ waits for a positive edge of the trigger signal (*sync*) to start data acquisition. As soon as DAQ is activated (*ready* command), the computer sends the scanning parameters to the axis motors (*move* command). Trigger signal (*sync*) appears with the very first movement along the Y axis. It causes DAQ to capture the information from the data line immediately. Data acquisition finishes when the required number of samples is acquired. Finally, measured data are delivered for DAQ to the computer and saved there. The synchronization procedure repeats for every line scan.

The synchronization technique depicted in Figure 2.9 makes the process of data acquisition independent from operation system delays, motors and DAQ programming delays, equipment responses etc.

2.5 Doppler System

Many different stages are passed on the way from data acquisition to data representation. These includes data acquisition itself, measurement system control, data processing, computations etc. We organize all the stages into a *Doppler system* or simply *system* shown in Figure 2.10. The Doppler system consists of three main parts. These are *software*, *measurement*, and *computational* modules. Let us consider each part separately.

The measurement module includes data acquisition equipment (or DAQ), Doppler measurement system, and different radar types. We have already discussed the measurement module in Section 2.4. It retrieves raw measured data for the further processing.

The software module is managed by a *control unit*. It is a routine which is

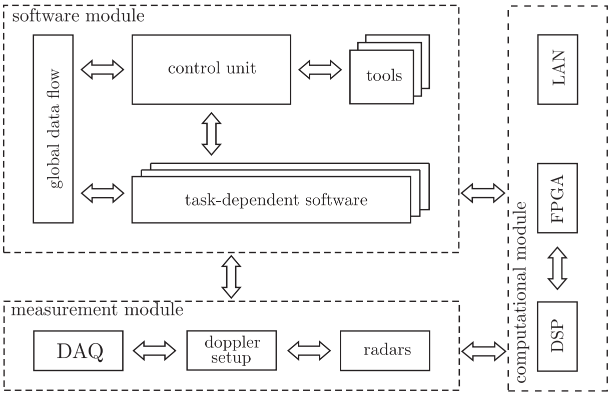


Figure 2.10: Doppler system

used to schedule *task-dependent software.* This software represents a number of different programs which communicate between each other by means of a *global data flow.* The task-dependent software includes signal processing procedures, software to control DAQ and the Doppler measurement system, user interface routines, distributed computation software etc. Generally, the Doppler system is controlled by the task-dependent software.

All the auxiliary routines are referred to as *tools.* These are data convertors, filter modules, simulation software etc.

The core of the software module is based on a *Modular Measurement System* (MMS) that has been developed in IZFP[1] for non-destructive testing purposes [24]. Nowadays MMS maintains many industrial devices and various measuring equipment. In this work we have developed the task-dependent software and tools compatible with MMS.

The computational module is used to perform external computations. Two types of communication are possible. The first is direct data flow from the measurement unit to FPGA and DSP devices. The second type of communication takes place if there are no external computational devices. This is the communication with a local network which provides the possibility to perform distributed computations. The Doppler system offers both broad functionality and modularity. This provide easy compatibility of new hardware and software modules with existing ones.

2.6 Doppler Resolution

We define *spatial resolution* of a CW radar to be the minimal distance between two defects that can be recognized as separate. In Chapter 1 we have already mentioned that CW radars do not retrieve any range information. This makes the determination of the target-radar distance and the size of the defect impossible. As a consequence, if there are many defects on some specimen, we can not determine their exact number.

[1] Fraunhofer Institute for Non-Destructive Testing, Germany, Saarbruecken

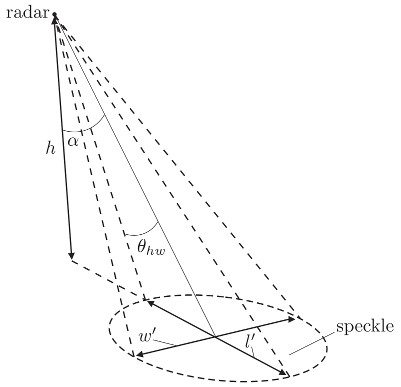


Figure 2.11: CW radar spatial resolution

A radar characteristic that deals with spatial resolution is a *speckle size*. We redefine it to be the area that the radar irradiates being placed in a particular spatial position (see Figure 2.11). Every point inside a speckle is irradiated by microwaves. A speckle is characterized by its length l' and width w'. Both l' and w' depend on radiation pattern θ_{hw}, radar latitude h, and incidence angle α, see Figure 2.11. Expressions which we use to evaluate dimensions of a speckle given as

$$\begin{aligned} l' &= h\left(\tan(\alpha + \theta_{hw}) - \tan(\alpha - \theta_{hw})\right) \\ w' &= \frac{2h}{\cos(\alpha)}\tan(\theta_{hw}) \end{aligned} \tag{2.6}$$

A small size of a speckle is preferable because the number of possible reflections is reduced within reflected area.

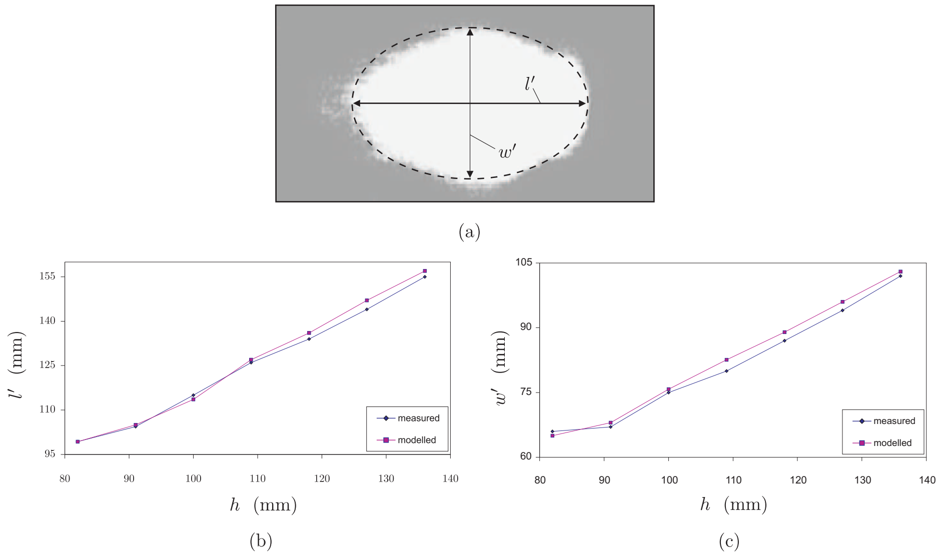
In Section 2.3 we discussed the advantage of oblique measurements over perpendicular ones. This implies an angle α to be in the interval $0 < \alpha < \pi/2$. Under that condition the length of a speckle l' is always larger than its width w', i.e. l' is the larger dimension.

Doppler measurements become efficient with a high Doppler frequency. In oblique measurements in order to receive high Doppler frequency f^d we have to increase angle α. When α is large, length l' rises so that the speckle dilates. The same effect is caused by increasing of h. Otherwise, large h ensures receiving of only a part of reflections that go straight back to the radar. Unfortunately, large h leads to decreasing of the intensity of microwaves what makes it difficult to recognize small defects. If h is too low (so that the very *near field* is reached), the response of the radar to the defect can be highly unpredictable. In practice a trade-off between *angle* α, *altitude* h, and *speckle length* l' must be found. Choice of parameters is usually done by a *calibration measurement* and *simulation*.

In the following subsection we confirm equation (2.6) practically.

2.6.1 Experiment Issue

We perform two different experiments. These are measurement with constant angle α at varying altitude h and vice versa. In this section we only represent



(a)

(b)

(c)

Figure 2.12: Radar speckle: (a) 2D measured radar speckle, $h = 100$ mm; (b) measured and modeled l' for all h; (c) measured and modeled w' for all h

the first part of the experiment because the results of the second experiment are similar. The radar altitude belongs to interval from 82mm to 136mm. Angle α is chosen to be 45 degrees.

As a defect we use a reflector (also called scatterer) of diameter d. In order to ensure back reflection at every h we chose d to be equal to the wavelength of the radiated microwaves. In the experiment we perform a 2D scan according to the schema given in Section 2.4. We determine the speckle size (i.e. l' and w') from acquired 2D data for every value of h as follows. We look for samples (which belong to the speckle) in the measured data. Thus, if for some value ξ which is associated with a sample holds

$$-10\text{db} \leq \xi \leq 0\text{db},$$

then we say that this sample is in the speckle. In the latter equation (db) stands for decibel. The decibel is a logarithmic unit used to describe a ratio. The feature of decibel scales is useful to describe very big ratios using numbers of modest size. The value ξ is given as

$$\xi = 20 \log \frac{\text{value of current sample}}{\text{maximum value over all samples}}.$$

In our case the lower border (i.e. -10db) is taken from antenna radiation pattern, which is expressed in db, at the angle θ_{hw} (see Section 1.4). Then, values of all samples belonging to the speckle assigned to 1 whereas the values of the others to 0. An example of a 2D speckle of the measured surface at $h = 100$mm is given in Figure 2.12(a). As we can see, the shape of the speckle resembles to an ellipse. From such a 2D figure we evaluate length l' and width w' of the ellipse. They are compared with corresponding length and width computed by equation (2.6).

In Figures 2.12(b) and (c) we present the measured and modeled l' and w' at the given altitude h. From these results we conclude that theoretical consideration and the experiments correspond to each other very well. It points at the correctness of the model we suggested to evaluate a CW radar speckle size given by equation (2.6).

We conclude that a speckle is determined by radar altitude h, incidence angle α, and an antenna radiation pattern θ_{hw}. If inside the speckle there is only one defect, which reflects back, then it will be detected without any ambiguity. In case of two and many defects detection becomes more difficult. The radar receives reflections from many defects at the same time. Every reflection represents a signal of different amplitude and Doppler frequency. Under these considerations we modify the output signal of the radar given in equation (1.22) to be a superposition over k scattered signals [25, p.8]:

$$s(t) = \sum_{0}^{k-1} A_k(t) \cos\left(2\pi \int_0^t f_k^d(t)\ dt\right), \tag{2.7}$$

where Doppler amplitudes A_k-th and Doppler frequencies f_k^d-th depend on the position of k-th defect inside the speckle. In practice it is extremely difficult to extract A_k-th and f_k-th. Often, the measured signal s is approximated as

$$s(t) = A(t) \cos\left(2\pi \int_0^t f^d(t)\ dt\right), \tag{2.8}$$

where A and f^d are some time-varying functions.

Let us introduce the extreme values of the Doppler frequency. The equations (1.15) and (1.16) introduces the Doppler frequency f^d which is observed at the radar when it lies on the same line of scan as the target. The Doppler frequency is maximal if the radar moves toward the target and is minimal if the radar moves in the opposite direction.

In the Doppler Measurement System the radar is located above the target so that the radar speckle is elliptical. In that case the Doppler frequency has its maximal value f_{max}^d when the radar starts irradiating the defect (see Figure 2.13(a) and (b), location p_1). It can be shown by the following argumentation. The frequency f_{max}^d can be computed from f^d by projection of the segment (p_0, p_1) onto axis x, see Figures 2.13(b). The angle of projection is given by the sum of oblique angle α and the angle θ_{hw} which characterize the antenna radiation pattern. Thus we have:

$$f_{max}^d = f^d \sin(\alpha + \theta_{hw}) = \frac{2v}{\lambda^{in}} \sin(\alpha + \theta_{hw}) \tag{2.9}$$

Let us show that the minimal value of the Doppler frequency f_{min}^d can be observed when the defect leaves the speckle, see location p_2 in Figures 2.13 (a) and (b). In that case the projection of the segment (p_0, p_2) onto x axis is given by the difference of angles α and θ_{hw}:

$$f_{min}^d = \frac{2v}{\lambda^{in}} \sin(\alpha - \theta_{hw}) \tag{2.10}$$

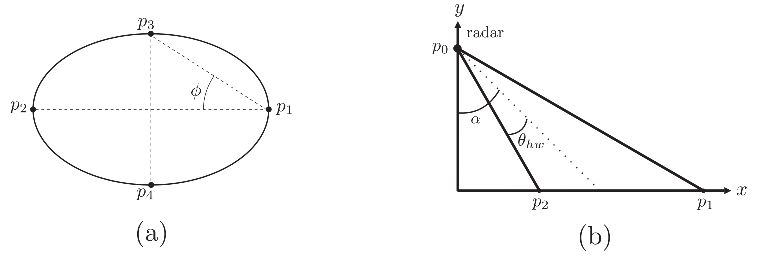


Figure 2.13: Extreme Doppler frequency values

We also show that if the position of the defect deviates from the line of scan by some angle ϕ (see location p_3 in Figure 2.13 (a)), then the maximal Doppler frequency decreases. In such a case a value of maximal Doppler frequency f^d_{max} is modified by introducing a projection of a segment (p_1, p_3) onto x axis as:

$$f^d_{max} = \frac{2v}{\lambda^{in}} \sin(\alpha + \theta_{hw}) \cos(\phi) \tag{2.11}$$

In general, both the Doppler amplitude A and the Doppler frequency f^d (from equation (2.8)) can be used to determine locations of defects inside the speckle. However, in case of many defects, the interference between cosines of amplitude A_k and frequency f^d_k is possible. This may impair the measured signal so that detection of locations of defects will not be possible.

In Chapter 3 we will present Doppler amplitude signal processing techniques. These techniques allow processing of the Doppler amplitude independently from Doppler frequency. Here we will discuss advantages and disadvantages of Doppler amplitude signal processing.

From equations (2.9), (2.10), and (2.11) we conclude that the maximal and the minimal Doppler frequencies vary with the position of the defect inside the speckle. The Doppler frequency is steadily dropping while the radar moves from p_1 to p_2. Such a variation of the Doppler frequency seems to be useful for detection of locations of defects. Unfortunately, because of speckle symmetry the reflection from p_3 will resemble the reflection from p_4 what definitely brings in ambiguity. A Doppler frequency processing will be discussed in Chapter 4 in detail.

Chapter 3

Doppler Imaging Technique

3.1 Definitions and Notations

Definition 3.1.1 *Let $\mathbb{T}^{m\times n}$ be a set of* ***matrices*** *of arbitrary type of size $m\times n$ such that $m, n \in \mathbb{N}^+$. Every entry (or element) of a matrix $\mathbf{A} \in \mathbb{T}^{m\times n}$ is addressed as $\mathrm{a}_{i,j} \in \mathbb{T}$, where $i \in [0 : m-1]$ and $j \in [0 : n-1]$. We assume $\mathbb{T}$ to be an arbitrary data type.*

In this work we denote a matrix by upper-case letter of a bold style. For addressing the particular matrix entry we use the corresponding low-case letter of a non-bold style.

Definition 3.1.2 *Let $\mathbf{V} \in \mathbb{T}^{m'\times n'}$, and $\mathbf{A} \in \mathbb{T}^{m\times n}$ be matrices. We define* ***extraction syntax*** *of a submatrix from $\mathbf{A}$ for $i_1, i_2, j_1, j_2 \in \mathbb{N}$, $m' = (i_2 - i_1 + 1) \leq m$, and $n' = (j_2 - j_1 + 1) \leq n$ as*

$$\begin{aligned}
&\mathbf{V} \leftarrow \mathbf{A}[i_1 : i_2][j_1 : j_2] \\
&\text{where} \quad \text{for all } i \in [0 : m'-1] \ \text{ and } j \in [0 : n'-1] \\
&\mathrm{v}_{i,j} = \mathrm{a}_{i_1+i, j_1+j}
\end{aligned}$$

Definition 3.1.3 *We introduce* ***update syntax*** *which is used to overwrite a submatrix of a given matrix. Let $\mathbf{A} \in \mathbb{T}^{m\times n}$ be a matrix to be updated with entries from $\mathbf{V} \in \mathbb{T}^{m'\times n'}$, then for some $\mathbf{A}' \in \mathbb{T}^{m\times n}$ and all the indices from* Definition 3.1.2 *we have*

$$\begin{aligned}
&\mathbf{A}' = \mathbf{A}[i_1 : i_2][j_1 : j_2] \leftarrow \mathbf{V} \\
&\text{where} \quad \text{for all } i \in [0 : m-1] \text{ and } j \in [0 : n-1] \\
&\mathrm{a}'_{i,j} = \begin{cases} \mathrm{v}_{i-i_1, j-j_1} & \text{if } i \in [i_1 : i_2] \ \wedge \ j \in [j_1 : j_2] \\ \mathrm{a}_{i,j} & \text{otherwise} \end{cases}
\end{aligned}$$

In the following we introduce shorthands for updating of $i-$th row and $j-$th column of matrix $\mathbf{A} \in \mathbb{T}^{m\times n}$:

$$\begin{aligned}
\mathbf{A}[i][] &\equiv \mathbf{A}[i : i][0 : n-1] \\
\mathbf{A}[][j] &\equiv \mathbf{A}[0 : m-1][j : j]
\end{aligned}$$

Definition 3.1.4 *Let* $\mathbf{v} \in \mathbb{T}^{n'}$ *and* $\mathbf{a} \in \mathbb{T}^{n}$ *be column vectors. We define* ***extraction syntax*** *of subvector from* $\mathbf{a}$ *for* $i, j \in \mathbb{N}$ *and* $n' = (j - i + 1) \leq n$ *as*

$$\begin{aligned} &\mathbf{v} \leftarrow \mathbf{a}[i:j] \\ &\text{where} \quad \text{for all } i' \in [0 : n'-1] \\ &\mathrm{v}_{i'} = \mathrm{a}_{i'+i} \end{aligned}$$

Definition 3.1.5 *Let* $\mathbf{a} \in \mathbb{T}^{n}$ *be a vector to be updated with entries from* $\mathbf{v} \in \mathbb{T}^{n'}$*. We introduce* ***vector update syntax*** *for some* $\mathbf{a}' \in \mathbb{T}^{1 \times n}$ *and all the indexes given in* Definition 3.1.4 *as*

$$\begin{aligned} &\mathbf{a}' = \mathbf{a}[i:j] \leftarrow \mathbf{v} \\ &\text{where} \quad \text{for all } i' \in [0 : n-1] \\ &\mathrm{a}'_{i'} = \begin{cases} \mathrm{v}_{i'-i} & \text{if } i' \in [i:j] \\ \mathrm{a}_{i} & \text{otherwise} \end{cases} \end{aligned}$$

In the following we define *continuous* and *discrete* versions of the *Fourier transform.* The *continuous Fourier transform* is used to operate with analog signals whereas the *discrete Fourier transform* operates with discrete ones. A general information about time-frequency analysis can be found in the literature [26–28].

Definition 3.1.6 *The* ***continuous Fourier transform*** (FT)

$$\mathcal{F}_x : \mathbb{R} \to \mathbb{C}$$

of an analog signal $x : \mathbb{R} \to \mathbb{T}$ *is defined as*

$$\mathcal{F}_x(f) = \int\limits_{-\infty}^{+\infty} x(t)\, e^{-j2\pi f t}\, dt,$$

where t *and* f *stand for time and frequency, respectively, and* $\mathbb{T}$ *is either* $\mathbb{R}$ *or* $\mathbb{C}$*. Here* x *is a time-domain signal.*

The ***inverse continuous Fourier transform*** (IFT)

$$\mathcal{F}_x^{-1} : \mathbb{R} \to \mathbb{C}$$

of an analog signal $x : \mathbb{R} \to \mathbb{C}$ *is defined as*

$$\mathcal{F}_x^{-1}(t) = \int\limits_{-\infty}^{+\infty} x(f)\, e^{j2\pi f t}\, df,$$

where x *is a frequency-domain signal.*

Definition 3.1.7 *Let* $\mathbf{x} \in \mathbb{T}^{n}$ *be a time-domain discrete signal. We define the* ***discrete Fourier transform*** (DFT)

$$\hat{\mathcal{F}}_{\mathbf{x}} : \mathbb{N}^n \to \mathbb{C}^n$$

such that for all $k \in [0 : n-1]$

$$\hat{\mathcal{F}}_{\mathbf{x}}(k) = \sum_{l=0}^{n-1} \mathrm{x}_l \exp\left(\frac{-j2\pi lk}{n}\right).$$

The ***inverse discrete Fourier Transform*** (DIFT)

$$\hat{\mathcal{F}}_{\mathbf{x}} : \mathbb{N}^n \to \mathbb{C}^n$$

of a frequency-domain discrete signal $\mathbf{x} \in \mathbb{C}^n$ *is defined for* $l \in [0 : n-1]$ *as*

$$\hat{\mathcal{F}}_{\mathbf{x}}^{-1}(l) = \frac{1}{n}\sum_{k=0}^{n-1} \mathrm{x}_k \exp\left(\frac{j2\pi kl}{n}\right)$$

Application of function **abs** to the FT is called **power spectrum** or **energy spectral density**. It shows how the energy of a signal is distributed with frequency [29].

In the following definition we introduce the Hilbert transform of a discrete signal (for reference see [30]).

Definition 3.1.8 *Let* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ *be discrete signals such that* $\mathbf{y} = \hat{\mathcal{F}}_{\mathbf{x}}$. *We define the* ***Hilbert transform*** $\hat{\mathcal{H}}_{\mathbf{x}} : \mathbb{R}^n \to \mathbb{C}^n$ *of* $\mathbf{x}$ for all $k \in [0 : n-1]$ *as*

$$\hat{\mathcal{H}}_{\mathbf{x}}(k) = \hat{\mathcal{F}}_{\mathbf{y}'}^{-1}(k), \quad \text{where} \quad \mathrm{y}'_k = \begin{cases} 0 & \text{if } k = 0, \\ -j\,\mathrm{y}_k & \text{if } 1 \le k < \dfrac{n}{2}, \\ 0 & \text{if } k = \dfrac{n}{2}, \\ j\,\mathrm{y}_k & \text{if } n/2 < k \le n-1 \end{cases}$$

We note that the signal $\mathbf{y}'$ *is the modified signal* $\mathbf{y}$ *such that in Frequency domain positive frequencies are multiplied with* $-j$ *and negative frequencies with* j.

The Hilbert transform is extensively used for signal processing purposes by the radar people. By using the Hilbert transform we can compute, for example, the amplitude $a \in \mathbb{R}$, $a > 0$ and the linear-varying phase $\boldsymbol{\phi} \in \mathbb{R}^n$ of a cosine signal $\mathbf{x} \in \mathbb{R}^n$. Let for all $k \in [0 : n-1]$:

$$\mathrm{x}_k = a\,\cos(\phi_k),$$

According to [30, page 688] and applying the Hilbert transform to the signal $\mathbf{x}$ we have:

$$\begin{cases} \mathrm{x}_k = a\,\cos(\phi_k) \\ \hat{\mathcal{H}}_{\mathbf{x}}(k) = a\,\sin(\phi_k) \end{cases}$$

From the latter equation we derive the amplitude a and the phase $\boldsymbol{\phi}$ for any k as

$$a = \sqrt{(\mathrm{x}_k)^2 + (\hat{\mathcal{H}}_{\mathbf{x}}(k))^2} \tag{3.1}$$

and

$$\phi_k = \arctan\left(\frac{\hat{\mathcal{H}}_{\mathbf{x}}(k)}{\mathrm{x}_k}\right) \tag{3.2}$$

Equations (3.1) and (3.2) introduce the main idea of the Hilbert transform for detection of the amplitude and the phase of a cosine signal. In practice the radar signals are not truly cosine (i.e. the amplitude a is a constant and the phase $\boldsymbol{\phi}$ is a linear-varying function). Very often the amplitude and the phase are arbitrary-varying signals. Such a situation if the amplitude and the phase may lead to errors caused by the Hilbert transform. We discuss the condition when the Hilbert transform fails in the following in detail.

In signal processing it is convenient to operate with so-called analytic signal which is formed from a real signal and its Hilbert transform. The definition of the analytic signal is given below.

Definition 3.1.9 *Let* $\mathbf{x} \in \mathbb{R}^n$ *be a real signal defined* for all $k \in [0 : n-1]$ *as*

$$\mathrm{x}_k = \mathrm{a}_k \cos(\phi_k),$$

where the amplitude $\mathbf{a} \in \mathbb{R}^n$ *and the phase* $\boldsymbol{\phi} \in \mathbb{R}^n$ *are arbitrary real signals. The* ***analytic signal*** *associated with* $\mathbf{x}$ *is a complex-valued signal* $\mathbf{x}^+ \in \mathbb{C}^n$ *such that* for all $k \in [0 : n-1]$ *holds:*

$$\mathrm{x}_k^+ = \mathrm{x}_k + j\hat{\mathcal{H}}_{\mathbf{x}}(k) = \mathrm{a}_k\left(\cos(\phi_k) + j\sin(\phi_k)\right) = \mathrm{a}_k e^{j\phi_k} \tag{3.3}$$

The real part of an analytic signal is called the *in-phase component*, whereas its imaginary part is referred to as the *quadrature component.*

The function **abs** applied to the analytic signal $\mathbf{x}^+$ retrieves the amplitude $\mathbf{a}$, which is referred to as a *complex envelope* or simply *envelope* [30, page 692]. Use of the function **abs** on $\mathbf{x}^+$ is equivalent to equation (3.1).

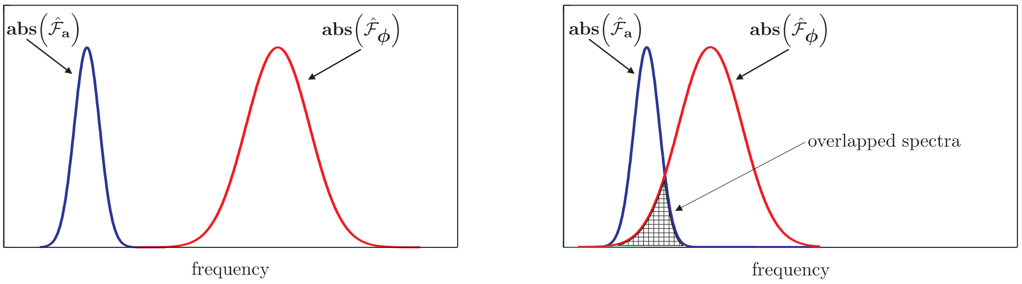
The function **arg** applied to $\mathbf{x}^+$ retrieves the phase $\boldsymbol{\phi}$. In the following we will refer it to as an *instantaneous phase.* Use of the function **arg** on $\mathbf{x}^+$ is equivalent to equation (3.2).

In practice, equation (3.3) does not always hold. This problem was firstly investigated for complex signals in [31] and [32]. It was shown that equation (3.3) is valid if $\mathbf{abs}(\hat{\mathcal{F}}_{\mathbf{a}})$ (i.e the power spectrum of the signal $\mathbf{a}$) and $\mathbf{abs}(\hat{\mathcal{F}}_{\boldsymbol{\phi}})$ (i.e. the power spectrum of the signal $\boldsymbol{\phi}$) are non-overlapped in the frequency domain, see Figure 3.1(a). If these spectra are overlapped then the Hilbert transform will be computed with errors. In such a case equation (3.3) is modified as

$$\mathrm{x}_k^+ = \mathrm{a}_k e^{j\phi_k} \ \pm \ \varepsilon_k,$$

where $\boldsymbol{\varepsilon} \in \mathbb{C}^n$ is some error signal. Figure 3.1(b) introduces the example when the amplitude and the phase power spectra are overlapped.

We will use the analytic signal and its envelope in the Doppler signal processing in order to detect an approximate position of the defect, see Section 3.2. The analytic signal plays an important role in *frequency analysis.* In Chapter 4 we also utilize the instantaneous phase for defect detection purposes.



(a) (b)

Figure 3.1: An example of non-overlapped and overlapped spectra: (a) Power spectra $\mathbf{abs}(\hat{\mathcal{F}}_\mathbf{a})$ and $\mathbf{abs}(\hat{\mathcal{F}}_\boldsymbol{\phi})$ are non-overlapped; (b) Power spectra $\mathbf{abs}(\hat{\mathcal{F}}_\mathbf{a})$ and $\mathbf{abs}(\hat{\mathcal{F}}_\boldsymbol{\phi})$ are overlapped

3.1.1 Doppler Imaging

In this section we introduce a formal description of a *Doppler imaging*. Under the term Doppler imaging we understand formation of 2D images from the data derived by means of the Doppler effect in microwaves. We already discussed the 2D scan technique in Section 2.4. The scan is performed by separate line-scans over the measured surface. Every line-scan is performed under particular conditions. These include motion pattern, motion speed, number of samples, sampling interval etc. Let us define performing of a line-scan in terms of the following function.

Definition 3.1.10 *Let* $\mathbf{p} \in \mathbb{R}^k$ *be a vector containing all the parameters which are used to perform Doppler measurements. We define an* ***acquisition*** *function*

$$\text{acquire} : \mathbb{R}^k \times \mathbb{N} \to \mathbb{R}^n,$$

such that m*-th line-scan* $\mathbf{x}^m \in \mathbb{R}^n$ *under parameters* $\mathbf{p}$ *is given as*

$$\mathbf{x}^m = \text{acquire}\,(\,\mathbf{p}, m\,)$$

In fact, line-scans can be combined into a matrix so that the measured surface is represented as 2D image. Generally, not only measured data but also the result of any signal processing procedure can be introduced to the user in a 2D form.

Definition 3.1.11 *We summarize our discussion above and define* ***Doppler image*** *to be a* 2D *representation of any Doppler data. In this work we introduce Doppler images in terms of matrices. We refer to the entry of a Doppler image as a pixel.*

Definition 3.1.12 *We call a* ***boolean Doppler image*** *or simply* ***boolean image*** *any Doppler image* $\mathbf{A} \in \mathbb{B}^{m \times n}$*, where* $\mathbb{B} \in \{1, 0\}$.

As the next step we define a function which forms a Doppler image from separate line-scans.

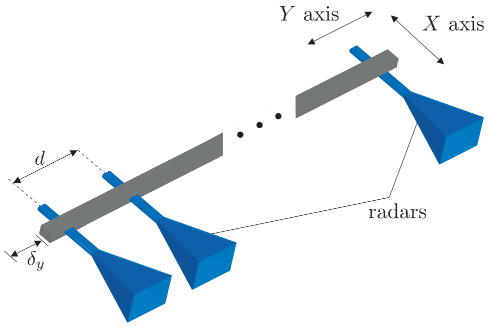


Figure 3.2: Multi-channel Doppler measurement system

Definition 3.1.13 *Let $m \in \mathbb{N}^+$ be a number of line-scans, $n \in \mathbb{N}$ be the length of a measured signal, $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a Doppler image, and $\mathbf{p} \in \mathbb{R}^k$ be a vector of parameters which is required by the function* **acquire**, *see* Definition 3.1.10. *We define a function*

$$\text{getData} : \mathbb{R}^k \times \mathbb{N}^+ \rightarrow \mathbb{R}^{m \times n}$$

such that its result

$$\mathbf{A} = \text{getData}(\mathbf{p}, m)$$

for all $i \in [0 : m - 1]$ is defined as

$$\mathbf{A}[i][] \leftarrow \text{getData}(\mathbf{p}, i)$$

As we can see the function **getData** forms and returns the Doppler image for constant $\mathbf{p}$ and $m \geq 1$.

Definition 3.1.13 introduces the output of the *single-channel* Doppler measurement system such that every line-scan in the resulting Doppler image is acquired sequentially. This certainly increases data acquisition time. In practice in order to accelerate measurements *multi-channel* Doppler measurement systems are under consideration. Such systems are able to acquire k line-scans simultaneously, where k denotes the number of used radars.

Realization of a multi-channel Doppler measurement system is relatively simple. A number of radars are organized into an *radar array* so that they are equally spaced with distance d between them, see Figure 3.2.

As we already mentioned the quality of a Doppler image depends on the step value between line-scans. In a multi-channel measurement system the radar array can be shifted by the step δ_y along axis Y, see Figure 3.2. This helps to archive arbitrary scan resolution of the Doppler image.

In order to detect defects (or features) of the analyzed surface, signal processing of the Doppler image is required. In the following section we discuss *standard signal processing*. It is a set of techniques which do not need any supplementary computational hardware and they are fast enough to perform *real-time* computations. The main drawback of standard signal processing techniques is their low ability to extract information from the measured data. As the result of this we observe a low spatial resolution of CW radar. Later in this thesis we compare standard signal processing techniques with advanced ones which we present in Chapters 4 and 5.

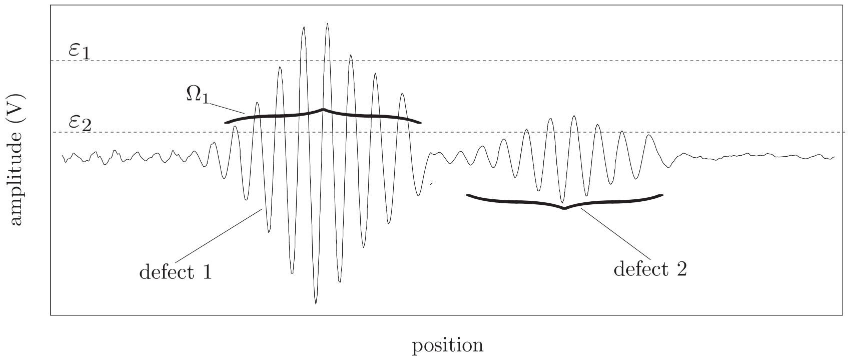


Figure 3.3: Signal thresholding

3.2 Standard Signal Processing Techniques

3.2.1 Threshold Evaluation

In this section we discuss a *threshold evaluation* technique. This implies splitting of the input data (Doppler image) into two parts according to some condition. A pixel belongs to the first group if it satisfies the condition. If the condition does not hold, the pixel is put into another group.

Definition 3.2.1 *Let* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *be a Doppler image and* $\varepsilon \in \mathbb{R}$ *be some threshold value. We define* ***thresholding*** *function*

$$\mathrm{iThreshold^b} : \mathbb{R}^{m \times n} \times \mathbb{R} \to \mathbb{B}^{m \times n}$$

such that its result

$$\mathbf{B} = \mathrm{iThreshold^b}(\mathbf{A}, \varepsilon) \tag{3.4}$$

for all $i \in [0 : n-1]$ and $j \in [0 : m-1]$ *is defined as*

$$\mathrm{b}_{i,j} = \begin{cases} 1 & \text{if} \quad \mathrm{a}_{i,j} \geq \varepsilon \\ 0 & \text{otherwise,} \end{cases}$$

where entries $\mathrm{a}_{i,j}$ *contain the measured doppler amplitude.*

In the latter definition a pixel is labeled as 0 if its amplitude $\mathrm{a}_{i,j}$ is less than the threshold. In this case we say that the area of the measured surface associated with the pixel does not contain any defects. Similarly, an area with defects is associated with a pixel labeled by 1.

In practice it is not obvious how to select a proper threshold value. Thus, if the threshold it too low then many pixels will be drawn to 1. This may lead to loosing information about defects making all the pixels indifferent. As an example we can consider the Doppler amplitude. Since it falls as the radar passes the defect, then the highest amplitude will correspond to the defect location. In this case in order to identify the defect location it is better to keep pixel values which are higher than the threshold. In the following we introduce a function which performs this operation.

Definition 3.2.2 *Let* $\mathbf{A} \in \mathbb{R}$ *be a Doppler image and* $\varepsilon \in \mathbb{R}$ *be some threshold value. We define **thresholding** function*

$$\text{iThreshold}^{\text{v}} : \mathbb{R}^{m \times n} \times \mathbb{R} \rightarrow \mathbb{R}^{m \times n}$$

such that its result

$$\mathbf{B} = \text{iThreshold}^{\text{v}}(\mathbf{A}, \varepsilon)$$

for all $i \in [0 : n-1]$ and $j \in [0 : m-1]$ *is defined as*

$$\mathrm{b}_{i,j} = \begin{cases} \mathrm{a}_{i,j} & \text{if} \quad \mathrm{a}_{i,j} \geq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

The thresholding procedure fails if the amplitude of the Doppler image significantly varies. In that case choice of a threshold becomes a difficult task. If it is too large, then small defects will not be detected, see Figure 3.3 for threshold value ε_1 and *defect 2*. A low threshold value (see ε_2) causes widening of an area of already detected defects (see area Ω_1 for *defect 1*), what certainly leads to degradation of spatial resolution.

In order to overcome this problem we use a *multi-thresholding* technique as given in the following. The Doppler image is split into segments so that every segment is processed by the thresholding procedure. An image segmentation can be done both in manual and automatic manner.

3.2.2 Image Closing

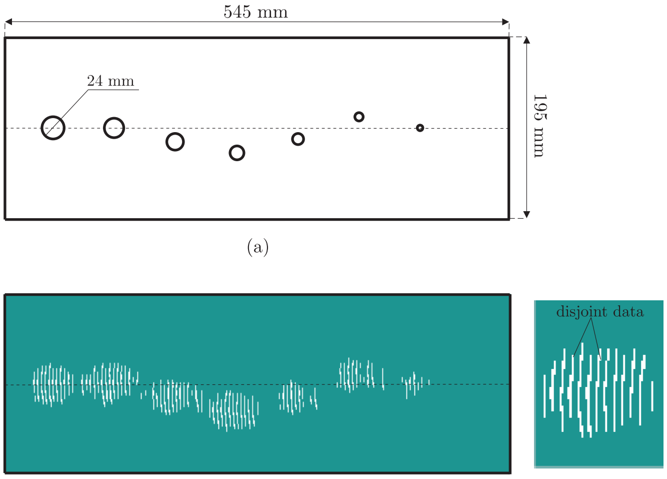
We utilize the *image closing* technique in order to improve microwave Doppler imaging. This is one of the basic image processing techniques which is also known as *morphology* [33], [34]. We apply image closing to boolean Doppler images computed by procedure **iThreshold**$^{\text{b}}$.

Because of the oscillating nature of the Doppler signal, pixels of the image which characterize a defect are disjoint. This complicates understanding of the resulting image. For example, detection of defect locations and their shape become not obvious. In order to demonstrate such a situation we perform Doppler imaging of the specimen represented in Figure 3.4(a). It is a metal plate with round holes of different diameters. Holes are not ordered along a centerline of the object and they rather deviate from it with different distances. Figure 3.4(b) represents the 2D Doppler image derived by procedure **iThreshold**$^{\text{b}}$. Even all the defects are detected well, the data which belong to the particular defect are disjoint, see zoomed picture.

In general the image closing procedure includes two steps. These are *dilation* and *erosion*. Let us define these procedures.

Definition 3.2.3 *Let* $\mathbf{A} \in \mathbb{B}^{m \times n}$ *be a boolean Doppler image and* $R \in \mathbb{N}^{+}$ *be some constant. We define the **dilation function** as*

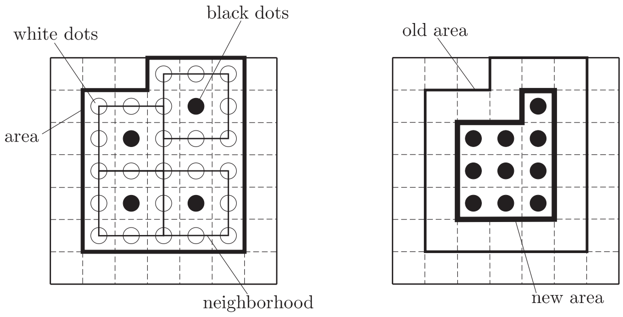
$$\text{ImgDilation} : \mathbb{B}^{m \times n} \times \mathbb{N}^{+} \rightarrow \mathbb{B}^{m \times n}$$



(a)

(b)

Figure 3.4: Image thresholding: (a) specimen; (b) thresholded Doppler image



(a)

(b)

Figure 3.5: Example of dilation and erosion: (a) dilation procedure output; (b) erosion procedure output

Let $\mathbf{B} \in \mathbb{R}^{m \times n}$ *be an output boolean image of* $\mathrm{ImgDilation}(\mathbf{A}, R)$*, then* for all $i \in [0 : m-1]$ *and* $j \in [0 : n-1]$ *we have*

$$\mathrm{b}_{i,j} = \begin{cases} \bigvee\limits_{i',j' \in [-R\,;\,R]} \mathrm{a}_{i+i',j+j'} & \text{if} \quad i \in [R : m-1-R] \wedge \\ & \qquad j \in [R : n-1-R] \\ \mathrm{a}_{i,j} & \text{otherwise} \end{cases}$$

According to Definition 3.2.3 pixels (or points) which belong to the neighborhood $[-R\,;\,R]$ of some non-zero pixel (i,j) are updated by 1. An example of the dilation procedure output is given in Figure 3.5(a). Initially this image only had four pixels set to 1, see *black dots* in Figure 3.5(a). The dilation procedure updates pixels in the neighborhood of existing black ones by 1, see *white dots*. The number of updated pixels around the existing one is determined by R. Here we assume $R = 1$. In Figure 3.5(a) we explicitly show the *neighborhood* of every initially set pixel.

A disadvantage of the dilatation procedure is widening of the resulting area

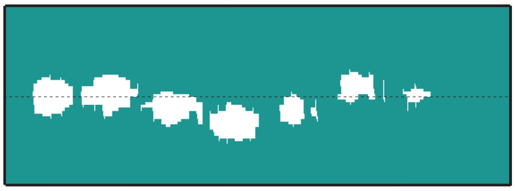


Figure 3.6: Image closing result

because separate defects might be overlapped. In order to suppress this effect we apply the *erosion* procedure.

Definition 3.2.4 *Let* $\mathbf{A} \in \mathbb{B}^{m \times n}$ *be a boolean Doppler image which is a result of the dilation procedure. We define an* ***erosion function*** *for kernel* $\mathbf{K} \in \mathbb{K}^{p \times p}$ (*see* Definition (3.2.3)) *as:*

$$\text{ImgErosion} : \mathbb{B}^{m \times n} \times \mathbb{B}^{p \times p} \longrightarrow \mathbb{B}^{m \times n}$$

Let $\mathbf{B} \in \mathbb{R}^{m \times n}$ *be an output boolean image for* $\text{ImgErosion}(\mathbf{A}, \mathbf{K})$*, then* for all $i \in [0 : m-1]$ *and* $j \in [0 : n-1]$ *we have:*

$$\mathrm{b}_{i,j} = \begin{cases} \bigwedge\limits_{i'=-R}^{R} \ \bigwedge\limits_{j'=-R}^{R} \quad \mathrm{k}_{i+R,j+R} \ \wedge \ \mathrm{a}_{i-R,j-R} & \text{if} \quad i \in [R : m-1-R] \ \wedge \\ & \qquad j \in [R : n-1-R] \\ 0 & \text{otherwise} \end{cases}$$

The erosion procedure only leaves a non-zero pixel if in its neighborhood all other pixels are non-zero. The result of this procedure for data from Figure 3.5(a) is represented in Figure 3.5(b). We notice that the *old area* shrinks to the smaller *new area*.

Dilation and erosion procedures allow us to combine disjoint pixels in the boolean image without widening of the resulting area. In the next definition we combine dilation and erosion procedures to one function.

Definition 3.2.5 *We introduce an* ***image closing*** *procedure for the input given in* Definitions (3.2.3) *and* (3.2.4) *as:*

$$\begin{aligned} &\text{ImgClosing} : \mathbb{B}^{m \times n} \times \mathbb{B}^{p \times p} \longrightarrow \mathbb{B}^{m \times n}, \quad \text{where} \\ &\mathbf{B} = \text{ImgClosing}(\mathbf{A}, \mathbf{K}) = \text{ImgErosion}(\text{ImgDilation}(\mathbf{A}, \mathbf{K}), \mathbf{K}) \end{aligned}$$

In Figure 3.6 we represent the result of the image closing procedure which is computed for the boolean Doppler image represented in Figure 3.4(b). As we can see image closing combines disjoint Doppler information. This certainly improves the resulting Doppler image, compare Figures 3.4 and 3.6.

It can be also very useful to apply the image closing procedure with different kernel sizes. These images can be used for better understanding of structure of defects.

3.2.3 Image Resizing

In practice the relation of width n to height m of the Doppler image is not equal to the relation of width L_y to height L_x of the measured surface, see Section 2.4. Here n is the length of a line-scan in samples, m is a number of line-scans equally spaced over L_x. On the one hand, value n should be high enough to digitalize a Doppler signal correctly. On the other hand, high m will slow down surface measurements.

Because of this reason we perform surface measurements for large n and small m. Then, we process a measured Doppler image by the *image resizing* technique. The goal is to transform a $(m \times n)$ image into a resized $(m' \times n)$ image such that

$$\frac{L_y}{L_x} = \frac{n}{m'}$$

In *digital signal processing* and *computer vision* an image resizing technique is known as *image scaling* or *image zooming and shrinking.* There are many different algorithms to perform image resizing. They differ in the way of interpolation of unknown pixel by existing ones. The most popular algorithms among others are

- nearest-neighbour interpolation
- bilinear interpolation
- bicubic interpolation

Nearest-neighbour interpolation basically makes the pixels bigger. The color of a pixel in the new image is the color of the nearest pixel of the original image. This technique offers pure image quality and does not introduce any *anti-aliasing. Bilinear* and *bicubic* interpolation determines the value of a new pixel based on the weighted average of the 4 and 16 pixels in the nearest neighbourhood, respectively. The averaging has an anti-aliasing effect and ensures relatively smooth images. In all the experiments we did not notice a great difference between bilinear and bicubic interpolations. In this work we utilize bilinear interpolation[1] because of its low cost, only $\mathcal{O}(\max(m', n)^2)$. More information about image resizing techniques can be found in [35], [36].

3.2.4 Peak Detection

Peak search technique can be useful for feature extraction from a Doppler image. Since a Doppler image contains line-scans, where a line-scan is considered as a signal, we define the *peak search* procedure for a signal. The idea of defect detection by using the peak search is based on the evidence that the Doppler amplitude rises while the radar passes the defect. The task of the procedure is to localize peaks of the amplitude. Under the term peak we understand a sample with a high amplitude value surrounded by samples with low amplitude values.

The peak search procedure is given in Definition 3.2.6. Its inputs are the signal to be analyzed and a threshold value. Often a Doppler signal is corrupted

[1] here we used Mathlab$^{\text{TM}}$image processing toolbox

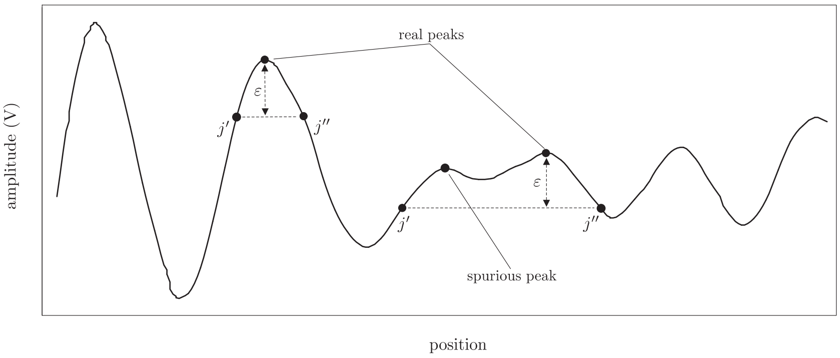


Figure 3.7: Peak Search Algorithm

by noise what causes *spurious peaks* to appear. We use the threshold value to exclude them from the result. A graphical representation of the result of the peak search algorithm is given in Figure 3.7.

Definition 3.2.6 *We define a* ***peak search*** *procedure as follows:*

$$\text{pSearch} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$$

Let $\mathbf{x} \in \mathbb{R}^n$ *be a vector containing a Doppler signal and* $\varepsilon \in \mathbb{R}$ *be a threshold. Then,* for all $i \in [0 : n-1]$

$$\text{pSearch}(\mathbf{x}, \varepsilon)_i = \begin{cases} 1 \quad \text{if} & \exists\, j' {\in} [0 : i-1] \quad \mathrm{x}_i - \mathrm{x}_{j'} \geq \varepsilon \wedge \qquad (1) \\ & \exists\, j'' {\in} [i+1 : n-1] \quad \mathrm{x}_i - \mathrm{x}_{j''} \geq \varepsilon \wedge \qquad (2) \\ & \forall\, k' {\in} [j' : i-1] \quad \mathrm{x}_i \geq \mathrm{x}_{k'} \wedge \qquad (3) \\ & \forall\, k'' {\in} [i+1 : j''] \quad \mathrm{x}_i \geq \mathrm{x}_{k''} \qquad (4) \\ 0 \quad \text{otherwise} & \end{cases}$$

In order to identify whether the current sample i is a peak we use four predicates, see Definition 3.2.6. In line (1) we check if there exists such a sample j' on the left from i such that $\mathrm{x}_i - \mathrm{x}_{j'}$ is not lower than threshold ε, i.e. we define the left border of the peak. Similarly, in line (2) we determine the right border j'' of a peak. Often, it is not sufficient to have only first two conditions to be true, to ensure that the processed point is a peak. An example of a fault situation is represented in Figure 3.7. We can see that the right border j'' of a spurious peak is located after the real peak. In order to avoid selectin of the spurious peak as a real one, we check whether a peak candidate (at sample i) is the maximum on the interval $[j' : j'']$, see lines (3) and (4). If these conditions hold the sample i becomes a peak.

In our experiments we observed that the peak search procedure works well even if an analyzed signal is corrupted by noise. In practice threshold ε is determined experimentally. The next definition introduces a function for peak detection in a Doppler image.

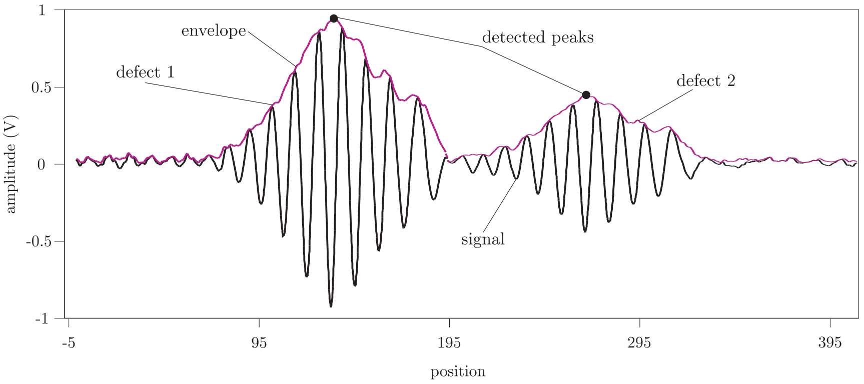


Figure 3.8: Envelope peak search

Definition 3.2.7 *We define an **image peak-search** function*

$$\mathrm{pImgSearch} : \mathbb{R}^{m\times n} \times \mathbb{R} \to \mathbb{R}^{m\times n}$$

which applies the linear peak search function above line by line to the Doppler Image $\mathbf{A} \in \mathbb{R}^{m\times n}$*. For* $\mathbf{B} = \mathrm{pImgSearch}(\mathbf{A}, \varepsilon)$ *and* $j \in [0 : m-1]$ *we define*

$$\mathbf{B}[j][] \leftarrow \mathrm{pSearch}(\mathbf{A}[j][], \varepsilon)$$

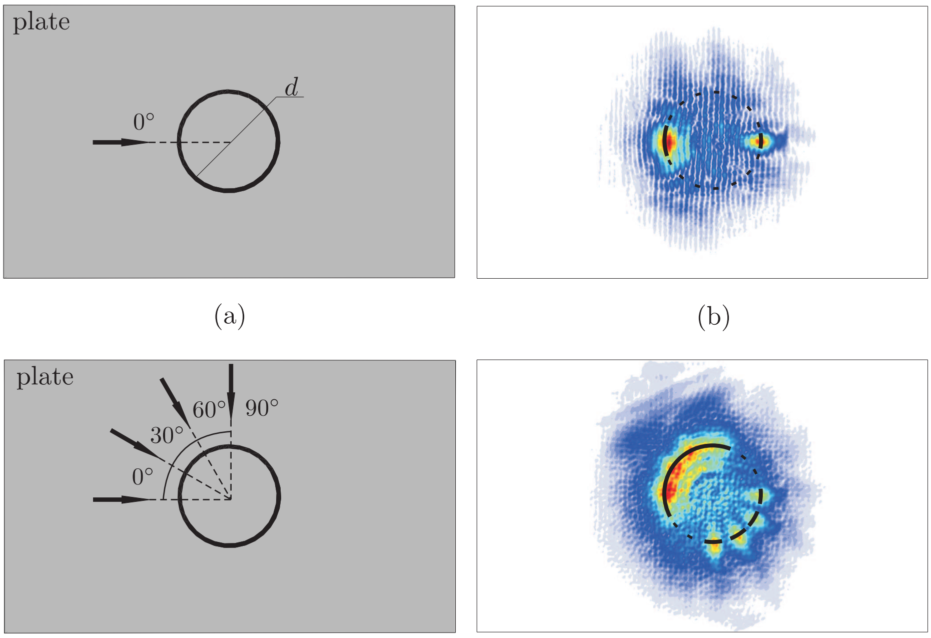
If we apply the peak search to the signal represented in Figure 3.3, then many peaks will be detected. It happens because of oscillating nature of the Doppler signal. In order to reduce the number of oscillations we build the *envelope* of the incoming signal using the Hilbert transform (see Definition 3.1.8) and apply the algorithm to it. The result of such a modification is represented in Figure 3.8, where the detected peaks are labeled by black dots. We note that the location of the detected peaks almost correspond to locations of the defects.

3.3 Multi-Angle Doppler Imaging

In the current section we introduce a novel approach for measurement and processing of the Doppler images in order to receive additional information about the specimen.

Often, Doppler image of the specimen does not show all the defects. This is not caused by an insufficient signal processing rather by the physics. It happens when the reflection from a defect does not propagate back to the radar. This is the case when the defect can not be detected even if it is situated inside the radar speckle. Obviously, the direction of propagation of reflected microwaves depends on geometry and shape of the particular defect.

We demonstrate such a case in the following experiment. For simplicity, we assume the geometry of the defect to be very simple. It is a circle of diameter d = 150mm made of thin metal wire 1mm thick. In order to ensure microwave reflection only from a circle, we place it on a flat metal surface, which is free from any defects, see Figure 3.9(a).



(a)

(b)

(c)

(d)

Figure 3.9: Multi-angle Doppler imaging with simple geometry: (a) scheme at 0° degrees scan; (b) Doppler image of (a); (c) scheme at $0°, 30°, 60°$ and 90° degrees scan; (d) Doppler image of (c)

The Doppler image is acquired by a surface scan from left to right in the direction pointed by the arrow at 0° degrees, see Figure 3.9(b). As we can see the circle is detected partially. The reflected signal only appears on the left and right edges of the circle whereas the rest is lost. In order to overcome this problem we propose to perform Doppler imaging from different view directions. In the context of the Doppler measurement system it is implemented by rotation of the sample around its center. We perform four surface scans, namely at $0°, 30°, 60°$ and 90° degrees, see Figure 3.9(c). For every line in the measured image we compute the envelope. In order to compute the resulting image the processed images at $0°, 30°, 60°$ and 90° degrees are arithmetically added to each other. We present the resulting image in Figure 3.9(d). We conclude that by further measurements all parts of the circle will be detected.

Multi-angle Doppler imaging can be done in different ways. It depends on type the of application, possible solutions of the problem, and cost. In general, the method of multi-angle Doppler imaging works with any method of single-angle Doppler imaging. In the following section we give a formal description of multi-angle Doppler imaging which suits any system implementation. We also discuss algorithms and functions which are required to produce the resulting image.

Let us denote a *sequence* to be a number of Doppler images which are obtained by multi-angle Doppler imaging of some specimen. Every image in a sequence corresponds to a specific angle of the scan. In order to indicate which angle is

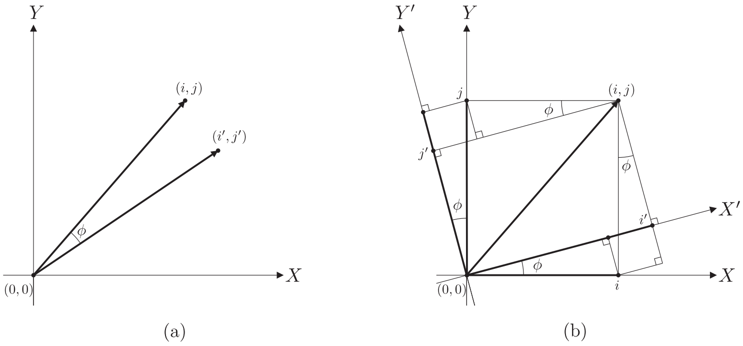


Figure 3.10: Image rotation procedure

associated with a Doppler image we use the following notation:

$$\mathbf{A}^{\xi} \in \mathbb{R}^{m \times n},$$

where $\mathbf{A}$ is a Doppler image and ξ is the angle at which it was acquired.

A definition of a multi-angle Doppler imaging function is given below. Its output is a vector which contains a sequence. We utilize a vector of Doppler images to save the whole sequence what significantly simplifies further data processing.

Definition 3.3.1 *Let* $\mathbf{p} \in \mathbb{R}^l$ *be a vector of parameters required by the function* getData*, see* Definition 3.1.13. *Let* $k \in \mathbb{N}^+$ *be a number of scans at different angles* $\xi_0, \xi_1, \ldots, \xi_{k-1}$*. The size of the acquired doppler image is* $(m \times n)$*. We define a* ***multi-angle Doppler imaging*** *function*

$$\text{getDataM} : \mathbb{R}^l \times \mathbb{N}^+ \rightarrow \left(\mathbb{R}^{m \times n}\right)^k$$

which returns the vector of Doppler images. For $\mathbf{v} = \text{getDataM}(\mathbf{p}, m)$ *and* $i \in [0 : k-1]$ *we define:*

$$\text{v}_i = \mathbf{A}_i^{\xi_i}, \text{ where } \quad \mathbf{A}_i^{\xi_i} = \text{getData}(\mathbf{p}, m)$$

3.3.1 Image Rotation

For further processing of a sequence of Doppler images the *back-rotation* is required. Under this term we understand rotation of all entries of a Doppler image around specific one at the given angle. In Figure 3.10(a) we introduce an example of rotation of a point (i, j) around a point $(0, 0)$ at the angle ϕ. The new point has coordinates (i', j'). The rotation of the point (i, j) can be considered in terms of rotation of the original plane XY. In Figure 3.10(b) we denote the rotated plane as $X'Y'$. Using simple trigonometrical transformations and Figure 3.10(b) we derive:

$$\begin{aligned} i' &= i \ \cos(\phi) + j \ \sin(\phi) \\ j' &= j \ \cos(\phi) - i \ \sin(\phi) \end{aligned} \tag{3.5}$$

Below we define the procedure to perform the back rotation.

Definition 3.3.2 *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *be a Doppler image to be rotated around some point* $(x, y) \in \mathbb{R}^2$ *and* $\xi \in \mathbb{R}$ *is a rotation angle. We define the* ***image rotation*** *function*

$$\mathrm{ImgRotate} : \mathbb{R}^{m\times n} \times \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^{m\times n}$$

which returns the rotated Doppler image. For $\mathbf{B} = \mathrm{ImgRotate}(\mathbf{A}, (x, y), \xi)$*,* $i \in [0 : m-1]$ *and* $j \in [0 : n-1]$ *we define:*

$$b_{i,j} = \begin{cases} a_{i',j'} & \text{if} \quad i' \in [0 : m-1] \ \wedge \ j' \in [0 : n-1] \\ 0 & \text{otherwise} \end{cases}$$

where i', j' *are computed according to* ***affine transformation*** *(see for reference* [37]*) and equation* (3.5) *as*

$$\begin{aligned} i' &= (i-x)\cdot\cos(\xi) + (j-y)\cdot\sin(\xi) + x, \\ j' &= (j-y)\cdot\cos(\xi) - (i-x)\cdot\sin(\xi) + y \end{aligned}$$

The latter equation is an updated version of equation 3.5*. Here, the rotation is performed around an arbitrary point* (x, y)*.*

Back rotation must be performed for every image from a sequence at the rotation angle associated with it. Let us define a function which performs back rotation of every image in a given sequence.

Definition 3.3.3 *We define a* ***sequence rotation*** *function*

$$\mathrm{seqRotate} : \left(\mathbb{R}^{m\times n}\right)^k \times \mathbb{R}^2 \to \ \left(\mathbb{R}^{m\times n}\right)^k.$$

Let $\mathbf{v} \in \left(\mathbb{R}^{m\times n}\right)^k$ *be a sequence such that* $\mathbf{v} = \left(\mathbf{A}_0^{\xi_0}, \mathbf{A}_1^{\xi_1}, \ldots, \mathbf{A}_{k-1}^{\xi_{l-1}}\right)$ *and* $(x, y) \in \mathbb{R}^2$ *be a rotation point. For* $\mathbf{v}' = \mathrm{seqRotate}(\mathbf{v}, (x, y))$ *and* $i \in [0 : k-1]$ *we define:*

$$\mathrm{v}_i' = \mathrm{ImgRotate}(\mathbf{A}_i^{\xi_i}, (x, y), -\xi_i)$$

In Definition 3.3.3 a negative rotational angle ξ ensures back rotation.

3.3.2 Image Merging

In the current section we define the *image merging* function. This function combines all the images from a sequence into the resulting Doppler image. The output Doppler image depicts information about defects of a specimen scanned at different angles of view. Generally, implementation of a particular merging procedure depends on Doppler image content. Further in the thesis we give a definition of arithmetical merging. It performs pixel-by-pixel arithmetical summation of the Doppler images.

Definition 3.3.4 *Let* $\mathbf{v} \in \left(\mathbb{R}^{m\times n}\right)^k$ *be a sequence of back-rotated Doppler images. We define an* ***arithmetical summation*** *function*

$$\mathrm{lSum} : \left(\mathbb{R}^{m\times n}\right)^k \to \mathbb{R}^{m\times n}.$$

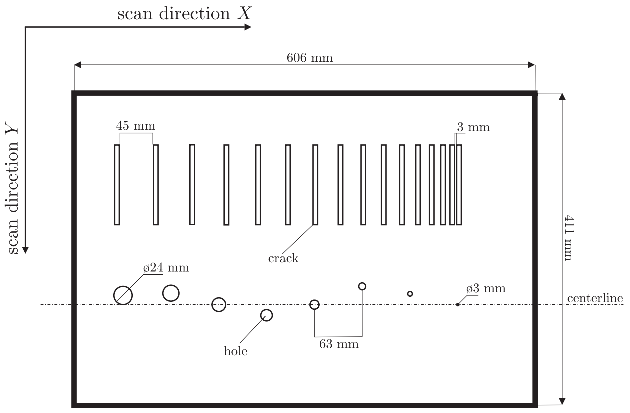


Figure 3.11: An artificial specimen, plate with holes and cracks

For $\mathbf{B} = \mathrm{lSum}(\mathbf{v})$ *we define:*

$$\mathbf{B} = \sum_{i=0}^{k-1} \mathrm{v}_i$$

If the sequence of images to be merged contains boolean Doppler images, then the definition of function **lSum** must be slightly updated. It can be done by changing of the arithmetical summation on to the logical summation.

3.4 Multi-Angle Doppler Imaging Results

We perform the multi-angle Doppler imaging by measuring an artificial specimen. It is a plate made of aluminium with artificial defects such as long cracks and holes, see Figure 3.11. Long cracks are placed on the surface at the stepwise increasing distances between them: from 3mm to 45mm (step is 3mm). The length, width, and depth of cracks are 100mm, 6mm, and 6mm respectively. The holes of the specimen have different diameters. They vary from 3mm to 24mm (step is 3mm). The placemen of holes slightly deviates from the centerline (see Figure 3.11).

Certainly, the distribution of defects represented in Figure 3.11 is not real. For example, it is unlikely that cracks (or defects) are nested very close to each other over the whole specimen. In this experiment both cracks and holes will help to investigate spatial resolution ability of the Doppler image. Thus, cracks are used to test the resolution in scan direction X. Holes will help to investigate the resolution ability in scan direction Y. Here we are interested wheatear the deviation of holes from the centerline will be seen in the resulting image.

We perform the experiment in two stages. Firstly, we check what is the quality of the doppler image when the microwaves directly irradiate the surface of the specimen. Then, we worsen measurement conditions covering the surface of the specimen by the semi-transparent (with regard to microwaves) material[2].

In the following we present colored and binary Doppler images of the experiment. The color of the image is proportional to the amplitude of the measured

[2] we use Polyvinyl Chloride (PVC) plate of thickness of 3mm

Doppler signal. The highest amplitude has a red color, whereas the lowest amplitude has a blue color.

In the Figures 3.12(b), (c), (d), and (e) a number of Doppler images is represented. Each image was measured without covering material at angles 0°, 45°, 90°, and 135° degrees, respectively.

Figures 3.12(b) and (d) discover vertical and horizontal borders of the specimen. In the image taken at 0° we can perfectly see holes whereas cracks appear partially so that only their faces are recognizable. In the image taken at 90° we distinguish holes and upper and lower edges of the cracks. In Figures 3.12(c) and (e) taken at 45° and 135° degrees we also can see holes and partially cracks.

We rotate and merge images. The resulting image is given in Figure 3.12(f). Here, all the holes except for the very last one are perfectly detected. We can not see the smallest hole because its size is beyond the radar detection ability. The first six cracks are also well recognizable. Although it is difficult to separate the following cracks we still have an evidence of defects. In Figure 3.12(g) we present a binary Doppler image. It is computed by thresholding and closing of Figure 3.12(f) (see Section 3.2). This gives additional information that is difficult to recognize on the colored image. In practice, both colored and binary Doppler images are usually considered.

As the next step we perform multi-angle Doppler imaging of the same specimen covered with semi-transparent material (PVC plate). The resulting colored and binary images are presented in Figures 3.13(a) and (b), respectively. Here, disturbances and amplitude fading are caused by the plastic plate. Figure 3.13(a) introduces significant quality degradation in comparison with the image in Figure 3.12(f). Nevertheless in both cases almost all the surface defects are well noticeable.

As we can see, in the experiment the cracks were not detected as well as the holes. The reason is that the cracks are situated very close to each other. It causes multi-reflections such that an acquired signal is heavily corrupted. We distinguish two types of corruption. In the first case spurious oscillations appear when reflection is caused by neighbors defects. It becomes difficult to determine real positions of the defect but it is still possible to recognize its physical presence. In the second case an acquired signal can be nearly zero. This can happen when reflected signals exhibit the opposite phase to each other so that their summation is zero. This will be interpreted as an absence of the reflected signal by the radar, i.e. as a non-defected surface what is certainly wrong. Using this consideration we can easily explain amplitude fading in the middle range of the cracks, see Figures 3.12(f) and 3.13(a).

In addition to disturbances of the acquired signal, Figures 3.12(f) and 3.13(a) also demonstrate low resolution ability of standard signal processing techniques. For example, we can see that closely spaced cracks form massive spots of defects so that details of the structure of the defects is lost. Within standard signal processing the resolution can not be improved. In this work we discuss a proposal for the resolution improvement in Chapters 4 and 5.

In all the experiments multi-angle Doppler imaging discovers very good stability and reputability of results. This certainly lets us expect its possible successful application in practice.

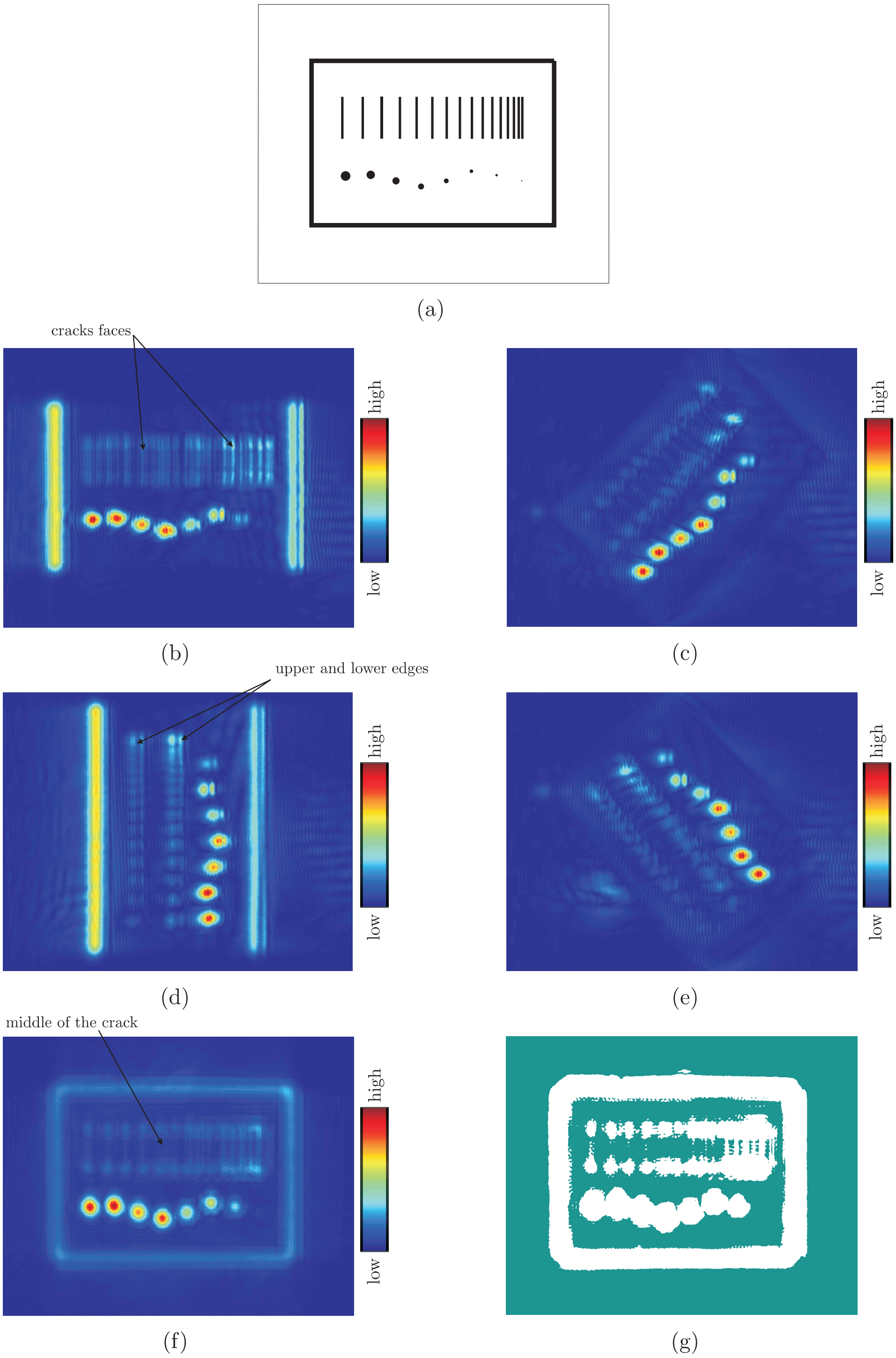


Figure 3.12: Doppler Imaging, specimen without covering: (a) specimen diagram; (b) 0° - Doppler image; (c) 45° - Doppler image; (d) 90° - Doppler image; (e) 135° - Doppler image; (f) merged Doppler image; (g) binary Doppler image

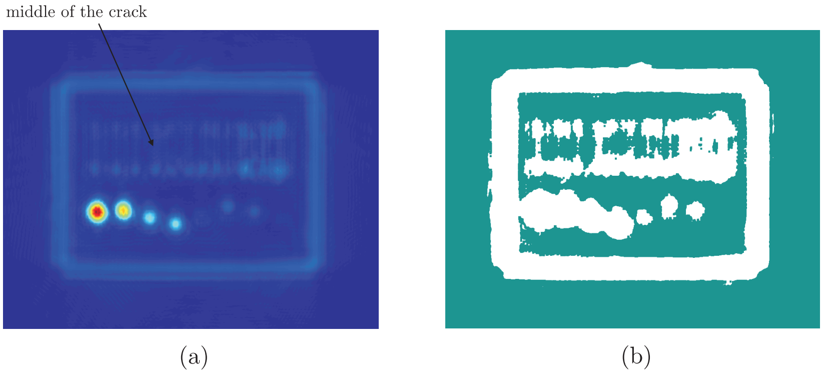


Figure 3.13: Doppler Imaging, specimen with covering, PVC plate of thickness of 3 mm; (a) merged image; (b) binary image

Chapter 4

Frequency Analysis

In this chapter we discuss Time-Frequency Signal Analysis and Processing (TFSAP). TFSAP includes theory and algorithms which are used for processing and analysis of *non-stationary signals*, i.e. signals having time-varying amplitude and frequency [38]. In the following we examine several TFSAP algorithms, namely a number of *Time-Frequency Distributions*, *Polynomial-Phase Transform*, methods of *Adaptive Frequency Estimation*, *Least-Square* techniques etc. The goal of this Chapter is to make a comparative study of different methods in order to discover the most suitable algorithms for Doppler signal processing.

4.1 Definitions and Notations

Definition 4.1.1 *The* ***total energy*** *(or simply energy)* $\mathcal{E}_{\mathbf{x}}$ *of a signal* $\mathbf{x}$ *is defined as a sum of all squared sample values:*

$$\mathcal{E}_{\mathbf{x}} = \sum_{i=0}^{n-1} \mathrm{x}_i{}^2$$

We note that in the case of continuous signals the energy is defined through the continuous integral.

Definition 4.1.2 *The* ***power*** $\mathcal{P}_{\mathbf{x}}$ *of the signal* $\mathbf{x}$ *is defined as its average energy per sample:*

$$\mathcal{P}_{\mathbf{x}} = \frac{\mathcal{E}_{\mathbf{x}}}{n} = \frac{1}{n} \sum_{i=0}^{n-1} \mathrm{x}_i{}^2$$

Definition 4.1.3 *The* ***signal-to-noise ratio*** (SNR) *is 10 times the logarithm ration of signal power to noise power and is measured in decibel* (or db)*:*

$$\mathrm{SNR} = 10 \log \frac{\mathcal{P}_{\mathbf{x}}}{\mathcal{P}_{\mathbf{n}}},$$

where $\mathbf{x}$ *and* $\mathbf{n}$ *are the signal and the noise, respectively. We note that signal-to-noise ration is used to estimate the influence of the noise onto the signal. Low* SNR *is associated with high signal corruption, whereas high* SNR *implies weak signal damage.*

Definition 4.1.4 *The* ***mean square error*** MSE *of two signals is used as a measure of their unsimilarity. Thus, the lower* MSE *is the more similar are the signals. Let* $\mathbf{x}$ *and* $\mathbf{x}'$ *be two signals, then* MSE *for them is*

$$\frac{1}{n}\sum_{i=0}^{n-1}\sqrt{(\mathrm{x}_i - \mathrm{x}'_i)^2}$$

4.2 Instantaneous Frequency (IF)

The frequency of the cosine signal is a well defined quantity. It determines the number of oscillations within a given time period. An example of such a signal is a cosine signal defined as

$$s(t) = a(t)\,\cos(2\pi f t + \theta), \tag{4.1}$$

where f and θ are frequency and initial phase, respectively. In general, amplitude $a : \mathbb{R} \to \mathbb{R}$ is an arbitrary function of time t. In the literature signals which satisfy equation (4.1) are called *stationary signals.* However, in general, real signals are not stationary. Spectral characteristics of such signals, in particular frequency, are time-varying. These signals are referred to as *non-stationary signals.* Frequency variation of a non-stationary signal is determined by *instantaneous frequency* (IF). In many situations such as seismic, telecommunication, medicine, radar, speech processing, or biomedical applications IF is used for description of physical phenomena [39].

The harmonic oscillation of non-stationary nature is written as

$$s(t) = a(t)\,\cos\phi(t), \tag{4.2}$$

where the argument $\phi : \mathbb{R} \to \mathbb{R}$ is called *instantaneous phase* and is defined as

$$\phi(t) = 2\pi\int\limits_0^t f(t)\,dt + \theta. \tag{4.3}$$

This leads us to the following definition of the instantaneous frequency $f : \mathbb{R} \to \mathbb{R}$:

$$f(t) = \frac{1}{2\pi}\frac{d\phi}{dt} \tag{4.4}$$

In the following we often refer the reader to equation (4.2) even if we operate with discrete signals. Under this circumstances we have in mind the discrete version of equation (4.2).

In Section 1.2 we have already mentioned that CW radars are widely used in applications connected with motion. In a great number of such applications it is assumed that an acquired Doppler signal is stationary. In order to evaluate the Doppler frequency of a stationary signal various algorithms were suggested [40]. Among others are such techniques as *zero-crossing technique*, *notch-filter*, *CSGN algorithm*, *Phase-Locked-Loop (PLL)*, *autocorrelation technique*, and *maximum of the Fourier transform.*

Doppler signals measured with the Doppler measurement system are non stationary. The instantaneous frequency of these signals carries information about geometrical properties of a defect and its location. In order to extract this information an exact evaluation of the Doppler frequency is required. This includes detection of the frequency variation in time with high frequency resolution. Algorithms which we mentioned above do not work with non-stationary signals properly. To overcome this problem we examine and compare a number of TFSAP methods for instantaneous frequency (IF) analysis of real and simulated data. Tasks which can benefit from exact Doppler frequency evaluation are:

- better understanding of the nature of the Doppler frequency
- application of the Doppler frequency approach in pattern recognition applications

The former task includes evaluation and detailed examination of Doppler frequency in different measurement conditions. A latter task implies distinguishing of defects using the instantaneous frequency as a recognition pattern. In both cases precise Doppler frequency estimation is required. That certainly depends on algorithm effectiveness and properties of the analyzed signal.

4.3 Doppler Signal Modeling

A concept of instantaneous frequency had been first examined in the late thirties by Carson and Fry [41] and then in 1946 by Van der Pol [42] and Gabor [43]. Nowadays the analysis of any non-stationary signal begins from the evaluation of its instantaneous frequency. Rising popularity of IF has caused development of various algorithms for its extraction and evaluation.

Choice of a specific algorithm for IF extraction depends on many conditions. These are signal type[1], its SNR, instantaneous frequency behavior etc. For example, it can be difficult for some algorithm to detect rapid IF variations. Often, at low SNR, smoothing of the signal in the time and frequency domain is required. All above mentioned reasons can essentially reduce the number of algorithms that can be applied for IF extraction purposes.

All modern algorithms for IF extraction requires the analytic signal as the input. As we already discussed the analytic signal is computed with the Hilbert transform (for reference see Definition 3.1.8). We have also mentioned that the Hilbert transform applied to the non-stationary signal (4.2) may introduce errors. It happens if the amplitude power spectra $\mathbf{abs}(\hat{\mathcal{F}}_a)$ and the phase power spectra $\mathbf{abs}(\hat{\mathcal{F}}_\phi)$ are overlapped. Errors in the analytic signal may lead to the wrong IF estimation.

In order to test the ability of the particular algorithm to estimate IF we propose the following approach. We model the amplitude and frequency signals such that their behavior remain the behavior of real ones. By using modeled amplitude and frequency we produce a discrete non-stationary signal according to equation

[1] the signal can be real, complex, stationary, or non-stationary etc. Many practical algorithms are very sensitive to the type of the input signal. Thus, in practice the signal is transformed into an appropriate type before the algorithm is applied.

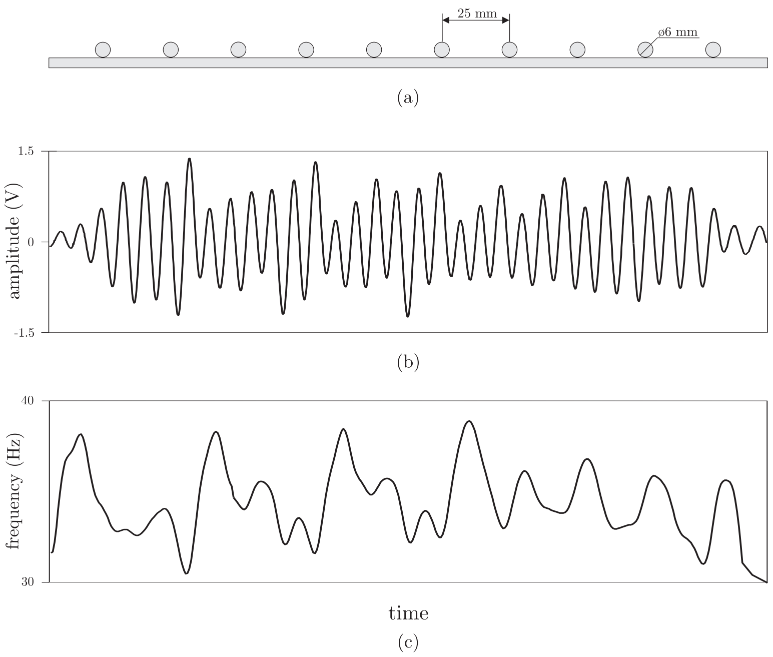


Figure 4.1: Multi-defects experiment: (a) artificial specimen; (b) Doppler acquired signal; (c) Doppler instantaneous frequency

(4.2). Applying the Hilbert transform to that signal we compute the analytic signal and extract its instantaneous frequency. We compare extracted and modeled instantaneous frequencies and make a conclusion about applicability of the algorithm for IF extraction purposes.

In order to observe alternation of the Doppler frequency of a real signal we perform measurements on an artificial specimen. This is a plate with point scatterers equally spaced along the line of scan, see Figure 4.1(a). A large number of defects (point scatterers) allows us to produce extremely varying Doppler frequency.

The acquired Doppler signal and its instantaneous frequency are represented in Figures 4.1(b) and 4.1(c), respectively. As we expected, the Doppler frequency exhibits non-trivial alternating. In order to model similar behavior of an instantaneous frequency a special technique is required.

4.3.1 Doppler Frequency Modeling

Literature offers different approaches for modeling of IF. According to the approach given in [44] it is proposed to model a non-stationary Doppler signal as the following function

$$z(t) = e^{-\frac{t^2}{\alpha^2}} e^{j2\pi f_0 t}, \tag{4.5}$$

where f_0 is the middle Doppler frequency and α is some real-valued coefficient. From equation (4.5) we conclude that the time-varying Doppler frequency changes linearly. Extraction of the linear instantaneous frequency is discussed in [45, 46]. Clearly, equation (4.5) can not describe the real IF signal shown in Figure 4.1(c).

In [47] *Young Bok Ahn* and *Sing Bai Park* proposed to model a power spectrum

r of a non-stationary signal through the Gaussian centered at the frequency f_m with standard deviation σ:

$$r(f) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(f - f_m)^2}{2\sigma^2}\right). \tag{4.6}$$

In order to make the power spectrum r more real it is artificially corrupted by noise so that the modified power spectrum P is given as

$$P(f) = -\ln g(f) \left(\frac{10^{\frac{\mathrm{SNR}}{10}} r(f)}{\sum\limits_{f} r(f)} \right), \tag{4.7}$$

where g returns random numbers uniformly distributed between zero and one. The modeled Doppler signal z in time domain is computed from the power spectra P applying the inverse Fourier transform as:

$$z(t) = \mathcal{F}_P^{-1}(f)$$

The latter modeling approach can be used to examine influence of the noise onto the Doppler frequency. A significant drawback of the approach is that the definition of power spectra r and P does not allow to model time-varying Doppler frequency.

Another approach to model non-stationary signals is based on the assumption that instantaneous frequency of the signal has sinusoidal form [48]. On the one hand analysis of such signals explores the ability of algorithms to extract dynamically-varying instantaneous frequency. On the other hand the sinusoidal instantaneous frequency does not describe a non-trivial behavior of the frequency of a real measured signal.

In [49] *Yuanyuan Wang* and *Peter J. Fish* proposed to model an instantaneous frequency f by filtering the *white gaussian noise*, for reference see [49, 50]. In the following we denote the noise by n. An example of a white gaussian noise is shown in Figure 4.2(a). The function which is also used in the modeling is called *impulse-response h*. The function $h : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defines parameters of the filter[2] which is not constant but time-varying. Since the noise n occupies the whole *power spectrum*, then by its filtering with time-varying filter we achieve non-stationary behavior of the resulting signal (i.e. filtered signal). Thus, the instantaneous frequency is modeled as follows:

$$f(t) = \int n(t - \tau) h(\tau, t)\, d\tau \tag{4.8}$$

A definition of the time-varying impulse response is given in terms of the Gaussian as follows:

$$h(\tau, t) = a \exp\left(-\frac{\tau^2}{2\sigma_t^2(t)}\right). \tag{4.9}$$

[2] power spectrum of the filter, its width and the central frequency positions

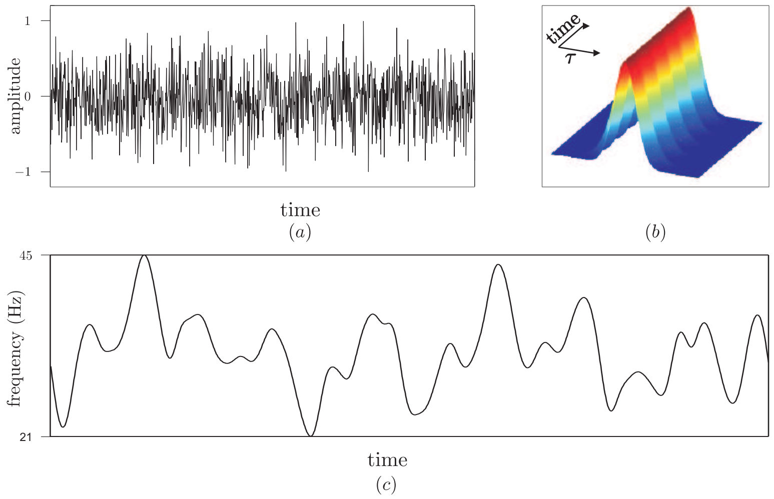


Figure 4.2: Modeled Doppler frequency: (a) an example of a white gaussian noise n; (b) an example of impulse response h; (c) modeled doppler frequency f

In equation (4.9) the width σ_t of a gauss function depends on time. Such a time-varying nature of σ_t makes it possible to model extremely non-stationary signals. According to [51] the function σ_t is defined as

$$\sigma_t(t) = \frac{1}{2\sqrt{2\pi\sigma_f(t)}}, \tag{4.10}$$

where σ_f is called the gaussian bandwidth and defines the time-varying width of the power spectrum of h in frequency domain. In this work we assume that

$$\sigma_f(t) = \cos(2\pi f' \, t) + c$$

where constants $f', c \in \mathbb{R}$ are set experimentally until the demanded form of f appears. An example of the function h for σ_t computed by using the latter equation is shown in Figure 4.2(b). As we can see it is the Gaussian which changes its width with time according to a harmonic law.

Usually extreme values of the modeled Doppler frequency do not correspond to the extreme values of the real Doppler frequency. Thus, the modeled Doppler frequency must be normalized in order to match a real frequency range. The definition of the normalization function is given below. The lowest value of f is assumed to be equal to the minimal Doppler frequency f^d_{min}, see equation (2.10). Analogously, the maximum value of f corresponds to f^d_{max}, which is given in equation (2.9). In Figure 4.2(c) we present the modeled and then normalized instantaneous Doppler frequency. We notice that the modeled Doppler frequency represented in Figure 4.2(c) behaves similarly to the real Doppler frequency represented in Figure 4.1(c).

Definition 4.3.1 *We define the* ***normalization*** *function*

$$\text{mnorm} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n.$$

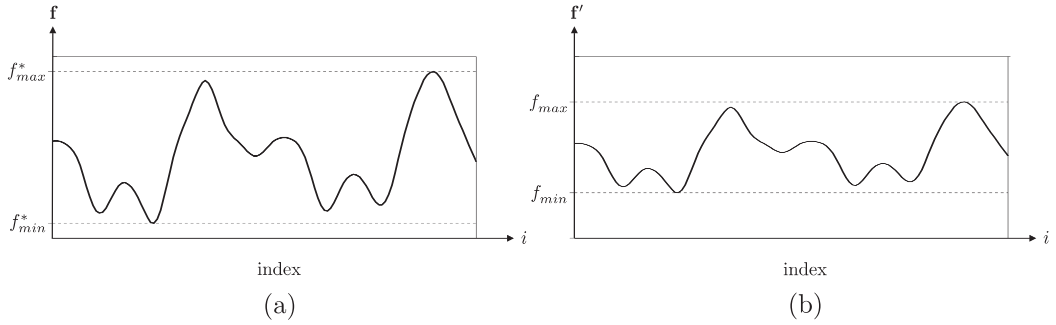


Figure 4.3: Doppler frequency normalization example: (a) modeled Doppler frequency; (b) normalized Doppler frequency

Let $\mathbf{f} \in \mathbb{R}^n$ *be the modeled Doppler frequency and* $f^*_{max} = \max(\mathbf{f})$ *and* $f^*_{min} = \min(\mathbf{f})$ *are its maximal and minimal values, see* Figure 4.3(a). *Let* $f_{max}, f_{min} \in \mathbb{R}$ *be the maximum and the minimum of the normalized modeled Doppler frequency*

$$\mathbf{f}' = \mathrm{mnorm}(\mathbf{f},\, f_{max},\, f_{min}),$$

see Figure 4.3(b). *The entries of* $\mathbf{f}'$ *are defined* for all $i \in [0 : n-1]$ *as*

$$\mathrm{f}'_i = f_{min} + \frac{f_{max} - f_{min}}{f^*_{max} - f^*_{min}} \cdot (\mathrm{f}_i - f_{min})$$

4.3.2 Doppler Amplitude Simulation

As we already discussed in the beginning of the current section the modeling of the non-stationary Doppler signal given in equation (4.2) requires modeling not only the instantaneous frequency f (i.e. the Doppler frequency) but also the amplitude a. The modeling of the amplitude can be successfully done by using equation (4.8). The only difference with frequency modeling is that we compute not the frequency f but the amplitude a. Here, the gaussian bandwidth function σ_f which is used in computation of impulse response h is linearly varying i.e.

$$\sigma_t(t) = a \cdot t + b,$$

where coefficients $a, b \in \mathbb{R}$ are chosen by the user.

In Figures 4.4(a) and (b) we compare measured and modeled Doppler signals. As we can see the behavior of these signals is very similar. The amplitude of both signals rises and falls so that separate shapes appear. The frequency varies what leads to the changing time distance between oscillations. Similar behavior of both signals lets us to expect that the method for Doppler signal modeling was chosen correctly.

In the following we represent comparative analysis of different algorithms for IF estimation. We examine the ability of algorithms to extract the instantaneous frequency of a modeled Doppler signal at different noise levels. For that reason we add the white gaussian noise of different levels to the modeled signal given in Figure 4.4(b). The range of Signal to Noise Ratio (SNR) changes from -10db to

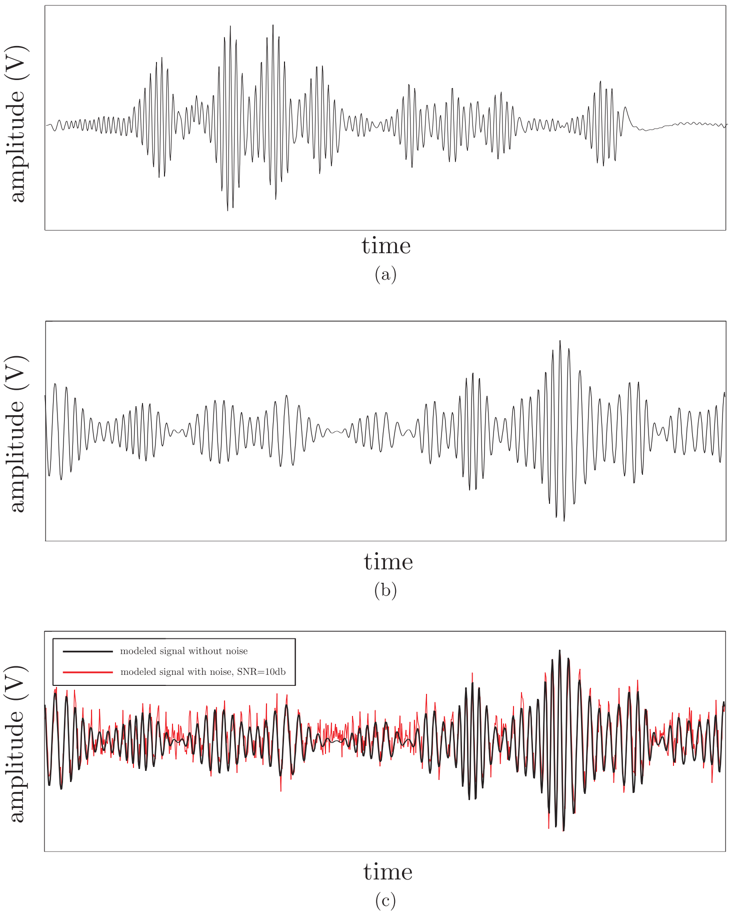


Figure 4.4: Measured and modeled Doppler signals: (a) measured Doppler signal; (b) modeled Doppler signal; (c) modeled Doppler signal with noise, SNR=10db

35db. In Figure 4.4(c) we represent an example of the noised modeled Doppler signal with SNR=10db.

Algorithms to be researched take as an input the analytic signal computed from the modeled Doppler signal with noise by using the Hilbert transform. Then, we extract the instantaneous frequency and compare it with the modeled one, given in Figure 4.2(c).

4.4 Frequency Estimation Techniques

4.4.1 Zero-Crossing IF estimator

In the literature it was shown that the Zero-Crossing estimator is highly inaccurate at extraction of instantaneous frequency of a non-stationary signal, see for reference [52]. Thus, for example, the variations of instantaneous frequency within a period of the signal cannot be estimated at all. The reason why we introduce Zero-Crossing IF estimator is the following. In many situations it is impossible to estimate instantaneous frequency of a signal by applying the frequency extraction algorithm to the whole signal. The only solution is to split the signal into data segments of some length and estimate IF frequency for every segment separately. We call such an approach a *segmentation technique*. In this thesis we utilize the Zero-Crossing estimator as a preprocessing technique for computation of the length of a data segment which is used as an input by more advances algorithms.

The Zero-Crossing estimator does not require any supplementary preprocessing except smoothing, see for reference [53]. Smoothing reduces spurious zero-crossings by rejecting the higher frequencies. The Zero-crossing estimator procedure seeks in the input signal for entries which pass through zeros, i.e. *zero crossings*. The Double time distance between two successive zero-crossings is reciprocal to a frequency in this interval. The following definition of Zero-Crossing procedure is based on the assumption that an input signal has at least two zero-crossings.

Definition 4.4.1 *We define a* ***zero-crossing*** *function*

$$\mathrm{ZeroCrossing} : \mathbb{R}^n \to \mathbb{R}^m.$$

Let $\mathbf{x} \in \mathbb{R}^n$ *be a signal to be processed,* $m < n$. *The entries of the signal* $\mathbf{z} = \mathrm{ZeroCrossing}(\mathbf{x})$ *are defined for all* $i \in [0 : m-1]$ *as:*

$$\begin{aligned} &\mathrm{z}_i = k, \quad \text{where} \\ &(\mathrm{x}_k \geq 0 \ \wedge \ \mathrm{x}_{k+1} < 0) \ \vee \ (\mathrm{x}_k < 0 \ \wedge \ \mathrm{x}_{k+1} \geq 0) \ \wedge \ k \neq 0 \ \wedge \ k \neq n-1. \end{aligned}$$

The predicate given above is used to find zero-crossings by checking the relation at (k)-th and $(k+1)$-th samples of the input signal $\mathbf{x}$. We notice that its 0-th and $(n-1)$-th samples are not considered.

We compute instantaneous frequency $\mathbf{f^x} \in \mathbb{R}^n$ of the input signal $\mathbf{x}$ for all $i \in [0 : n-1]$ as

$$\mathrm{f}_i^{\mathbf{x}} = \frac{1}{2 \ \triangle t \ (\mathrm{z}_{j+1} - \mathrm{z}_j)} \quad \text{if} \quad \mathrm{z}_j \leq i < \mathrm{z}_{j+1}, \text{ where } j \in [0 : m-2] \tag{4.11}$$

Instantaneous frequency at boundaries (i.e. $\mathbf{f}^{\mathbf{x}}[0 : \mathrm{z}_0 - 1]$) and $\mathbf{f}^{\mathbf{x}}[\mathrm{z}_{m-1} : n-1]$ are filled from their closest already computed samples.

We assume the length of a data segment w to be the *average period* over the Doppler signal which is computed as follows:

$$w = \frac{2 \sum\limits_{j=0}^{m-2} (\mathrm{z_{j+1}} - \mathrm{z_j})}{m-1} \tag{4.12}$$

In the following we will discuss length of a data segment and the segmentation technique in more details.

4.4.2 Adaptive IF Estimation

The adaptive IF estimation includes a great number of different techniques for instantaneous frequency extraction. One of the basic techniques is called *Phase Locked Loop* (or shortly PLL). This technique is widely used in Doppler applications such as traffic control [25]. PLL's drawback is its inability to track rapid changes in the instantaneous frequency. A general information about PLL technique can be found in [54, 55].

Another group of adaptive IF estimators is based on *linear prediction* of measured data. In this work we examine the *Least-Mean Square* (LMS) adaptation algorithm which was introduced in [56]. The idea of the LMS algorithm is to a find vector of *prediction coefficients* $\mathbf{g}^k = (\mathrm{g}_0^k, \mathrm{g}_1^k, \ldots, \mathrm{g}_{L-1}^k) \in \mathbb{C}^L$, where $L \in \mathbb{N}^+$, for every sample x_k of the input analytic signal $\mathbf{x} \in \mathbb{C}^n$. These coefficients are used to compute so-called prediction sample x'_k so that the difference (or error) between predicted sample x'_k and original sample x_k is minimal i.e.:

$$\min_{\mathbf{g}^k \in \mathbb{R}^L} \ \epsilon = (\mathrm{x}_k - \mathrm{x}'_k)^2 \quad \text{for all} \quad k \in [0 : n-1] \tag{4.13}$$

According to the definition given in [56] prediction sample x'_k is computed from a data segment $\mathbf{x}[k - L : k - 1]$ of length L as follows:

$$\mathrm{x}'_k = \sum_{l=0}^{L-1} \mathrm{g}_l^k \, \mathrm{x}_{k-l-1}, \tag{4.14}$$

In general, the vector of prediction coefficients $\mathbf{g}^k$ determines spectral characteristics of k-th sample of the input analytic signal $\mathbf{x}$. One of these characteristics is the instantaneous frequency. In order to compute the k-th instantaneous frequency we have to solve the following maximization task:

$$\mathrm{f}_k^{\mathbf{x}} = \frac{p}{m \cdot \triangle t} \quad \text{such that} \quad \max_{p \in [0:m-1]} \mathbf{q}^{(\mathbf{g}^k)}, \tag{4.15}$$

where $p \in \mathbb{N}$ is a sample index, $m \in \mathbb{N}$ is a number of frequencies, $\triangle t \in \mathbb{R}$ is a sampling interval, and $\mathbf{q}^{(\mathbf{g}^k)} \in \mathbb{R}^m$ is called a *prediction spectrum*. The prediction spectrum is unique for every vector of prediction coefficients $\mathbf{g}^k$ and can be computed for all $p \in [0 : m-1]$ as

$$\mathrm{q}_p^{(\mathbf{g}^k)} = \left| 1 - \sum_{l=0}^{L-1} \mathrm{g}_l^k \exp\left(-\frac{j2\pi p\,(l+1)}{m} \right) \right|^{-2}.$$

The choice of a number of frequencies m depends on desired frequency resolution of the problem (4.15). In practice m is determined experimentally. Usually it is assumed to be some large value which ensures smoothness of prediction spectrum $\mathbf{q}^{\mathbf{g}^k}$.

A computation scheme for finding the prediction vector $\mathbf{g}^k$ utilizes a *steepest descent* approach. This approach works in an iterative way:

$$\mathbf{g}^{k,j+1} = \mathbf{g}^{k,j} - \big(\mu \cdot (\mathrm{x}_k - \mathrm{x}'_k)\big)\ \mathbf{x}[k-L:k-1], \tag{4.16}$$

where μ is a convergence parameter, $\mathbf{g}^{k,j}$ denotes value of $\mathbf{g}^k$ at iteration j. The computation of k-th prediction vector starts from the assignment $\mathbf{g}^{k,0} = \mathbf{g}^{k-1}$ where $\mathbf{g}^{k-1}$ denotes prediction vector computed for the previous data sample $(k-1)$. The process (4.16) stops when ϵ becomes less or equal to some constant ε and satisfies the following condition:

$$\epsilon \leq \varepsilon. \tag{4.17}$$

Since the number of unknown coefficients L is usually small the steepest descent procedure (4.16) converges quite well. A detailed analysis and theoretical fundamentals of LMS algorithm are given in [57, pages 231-319]. A short survey of steepest descent methods can also be found in Chapter 5 of this thesis.

Selection of length L, the initial values $\mathbf{g}^{0,0}$, and constant μ is studied in [56]. In practice, L is usually chosen experimentally analyzing measured and simulated data. According to the latter reference we should choose

$$\mu = \frac{\alpha}{L}, \tag{4.18}$$

where α is in range from 0.1 to 1.0. The initial value of prediction vector affects the convergence speed of LMS algorithm. In the thesis we let $\mathbf{g}^{0,0}$ to be vector of zeros.

An important advantage of the LMS algorithm is its ability to evaluate IF only from L known previous samples of the input signal so that the length L is very short[3]. Thus, the LMS algorithm does not require information about the whole signal length and can evaluate IF with every new received data sample not waiting for the end of data acquisition.

One more advantage of the LMS algorithm is its computational efficiency. This algorithm is considerably faster than other IF algorithms we will consider later. Unfortunately, real signals are not perfectly predictable, in particular, because of a presence of noise. This reason causes extremely slow convergence of the algorithm in the sense of condition (4.17). In order to prevent endless algorithm repetitions it is stopped when a certain number of iterations *maxIter* is reached. Of course this leads to wrong instantaneous frequency estimation.

We tested the LMS algorithm at different levels of noise from SNR=-10db to SNR=35db (SNR is given in Definition 4.1.3). We added the noise to the modeled Doppler signal[4] (for reference see Sections 4.3.1 and 4.3.2) whose instantaneous frequency is known. By using the Hilbert transform we computed the

[3] L is referred to as short in comparison with, for example, length of a data segment which is required by such techniques as the Fourier Transform, Zero-Crossing, or other algorithms we will consider further in the thesis

[4] the length of the modeled Doppler signal is 1024 samples

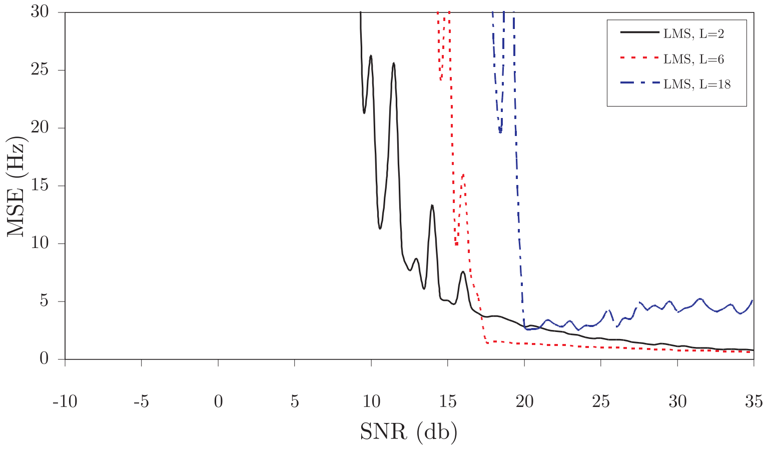


Figure 4.5: LMS algorithm testing

analytic signal and processed it with the LMS algorithm in order to compute the instantaneous frequency. Afterwards, we computed the mean square error (MSE) between modeled and computed instantaneous frequency for every noise level, see Definition 4.1.4. In Figure 4.5 we introduce the result of the LMS algorithm for different data segment lengths L. The test shows that the LMS is very sensitive to the presence of noise. The noise causes incorrect estimation of the instantaneous frequency from SNR=-10db to about SNR=20db. A sufficient low average MSE is only reached from SNR=25db to SNR=35db. Here MSE is about 0.95Hz at the length $L = 6$. Such a high noise sensitivity makes LMS useless in real Doppler applications which are connected with instantaneous frequency estimating.

In the simulations the best frequency approximation was derived at $L = 6$. If L is too low (see in the figure $L = 2$), then IF cannot be estimated because of the lack of information in the prediction spectrum. Otherwise, if L is too high (see $L = 18$), then IF is over-estimated, i.e. the prediction does not have pronounced peak corresponding to the instantaneous frequency.

4.4.3 Linear Least Squares IF Estimation LLS

The Linear Least Squares algorithm (or shortly LLS) belongs to the group of algorithms which are based on *polynomial phase modeling* of the analyzed signal. This introduces a discrete instantaneous phase $\boldsymbol{\phi} \in \mathbb{R}^n$ defined from equation (4.2) for $i \in [0 : n-1]$ as

$$\phi_i = \phi(i \cdot \triangle t) = \mathrm{b}_0 + \mathrm{b}_1\, i + \mathrm{b}_2\, i^2 + \ldots + \mathrm{b}_p\, i^p = \sum_{k=0}^{p} \mathrm{b}_k\, i^k, \tag{4.19}$$

where $\triangle t$ is a sampling interval, p denotes a *polynomial order* and $\mathbf{b} \in \mathbb{R}^{p+1}$ is a *vector of polynomial coefficients*.

A polynomial approximation of the instantaneous phase requires an implicit assumption about the nature of the analyzed signal. For example, high p is used to model a rapidly varying phase, whereas low p is responsible for a slow-varying one. In order to decide which polynomial p can be used in the case of a Doppler signal we perform comparative analysis of real and modelled data.

By substitution of (4.19) into (4.4) we derive a definition of instantaneous frequency in terms of polynomial coefficients:

$$\mathrm{f}_i = \frac{1}{2\pi} \sum_{k=1}^{p} k\, \mathrm{b}_k\, i^{k-1}. \tag{4.20}$$

Before the LLS algorithm starts, a rough approximation of the instantaneous phase is performed. It can be done by applying function **arg** (see Definitions 1.1.7 and 3.1.8) to an analytic signal $\mathbf{z} \in \mathbb{C}^n$, which is associated with the analyzed Doppler signal, as

$$\boldsymbol{\phi}' = \arg(\mathbf{z}). \tag{4.21}$$

The entries of the instantaneous phase $\boldsymbol{\phi}'$ are computed for all $i \in [0 : n-1]$ as given below

$$\phi'_i = \arctan\left(\frac{\mathrm{Im}(\mathrm{z}_i)}{\mathrm{Re}(\mathrm{z}_i)}\right)$$

The phase $\boldsymbol{\phi}'$ defined in (4.21) contains discontinuities when $\mathrm{Re}(\mathrm{z}_i)$ is zero. To remove these discontinuities a *phase unwrapping* procedure is applied to $\boldsymbol{\phi}'$. The definition of an unwrapping phase procedure is given in Appendix A.0.1. In the following we denote $\boldsymbol{\phi}$ to be the unwrapped version of $\boldsymbol{\phi}'$.

The LLS algorithm derives the vector of polynomial coefficients $\mathbf{b}$ which is used to compute IF through equation (4.20) by solving the following system of linear equations:

$$\boldsymbol{\phi} = \mathbf{H}\,\mathbf{b}, \tag{4.22}$$

The solution of equation (4.22) is proposed in [58, p.49] as follows:

$$\mathbf{b} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\boldsymbol{\phi},$$

where entries of matrix $\mathbf{H}$ are defined for $i \in [0 : n-1]$ and $j \in [0 : p]$ as

$$\mathrm{h}_{i,j} = i^j.$$

Let us denote a function which implements the LLS algorithm i.e. it returns computed instantaneous frequency according to equation (4.22) such as

$$\mathrm{LLS} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, \tag{4.23}$$

where its first and second arguments are the unwrapped phase and the polynomial order p. A general idea behind LLS is to approximate and smooth the instantaneous phase $\boldsymbol{\phi}$ by a polynomial. Smoothing is needed since in general $\boldsymbol{\phi}$ contains errors because of two main reasons. First of all in real situations the instantaneous phase is corrupted by noise. Thus, even small noise in the denominator of function **arg** may cause significant errors in the resulting signal $\boldsymbol{\phi}'$. Clearly, errors in $\boldsymbol{\phi}'$ will also appear in $\boldsymbol{\phi}$. The second reason of corruption of $\boldsymbol{\phi}$ can be explained through errors introduced by the Hilbert transform (for more information see Definition 3.1.8).

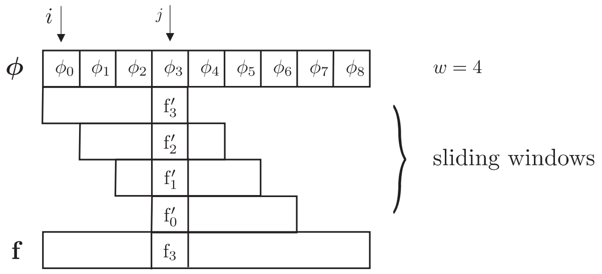


Figure 4.6: Segmentation algorithm

We note, that the modeled frequency represented in Figure 4.2(c) cannot be approximated by any polynomial along its entire length. Instead, we suggest to split the input signal into segments and approximate each segment with some polynomial. The segment size (window size) w is assumed to be equal to the average period computed by Zero-Crossing IF estimator, see equation (4.12). Wrong choice of the window size w may lead to incorrect frequency estimation. Hence, at long segments the computed instantaneous frequency (output of the LLS algorithm) may be heavily over-smoothed what leads to degradation of frequency resolution. With short segments the computed instantaneous frequency may be detected completely wrong because of insufficient information about the analyzed signal.

Definition 4.4.2 *We define segmentation function*

$$\mathrm{segLLS} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{N}^+ \rightarrow \mathbb{R}^n$$

Let $\boldsymbol{\phi}$ be the unwrapped instantaneous phase of an analytic signal, w be a segment size, and p be a polynomial order. We define the instantaneous frequency $\mathbf{f} \in \mathbb{R}^n$ is defined as

$$\mathbf{f} = \mathrm{segLLS}(\boldsymbol{\phi},\, p,\, w),$$

$$\text{where} \quad \text{for all } j \in [0 : n-1\,] \;\; \wedge \;\; i = \max(0,\, j - w + 1) \quad (1)$$

$$\mathrm{f}_j = \frac{\sum\limits_{\forall k \in [i:j]} \mathrm{LLS}\big(\,\boldsymbol{\phi}[k : k{+}w{-}1], p\,\big)_{j-k}}{(j{-}i{+}1)} \qquad (2)$$

Let us consider the segmentation technique by the example given in Figure 4.6, where $w = 4$. The window slides along $\boldsymbol{\phi}$ one by one sample. For every segment an instantaneous frequency is computed using LLS algorithm. Thus, at specific sample j there will be computed at most w frequency values f'_3, f'_2, f'_1, and f'_0, see in the figure. Then, j-th instantaneous frequency value f_j is computed as their average.

According to Definition 4.4.2 the function **segLLS** is executed into two stages for every j. At first, see line (1), intervals of phase signal entries with base $k \in [i : j]$ are determined. These entries participate in the forming of k-th frequency sample. Then, in line (2) the intervals mentioned above are processed by **LLS** and f_j-th frequency sample is computed.

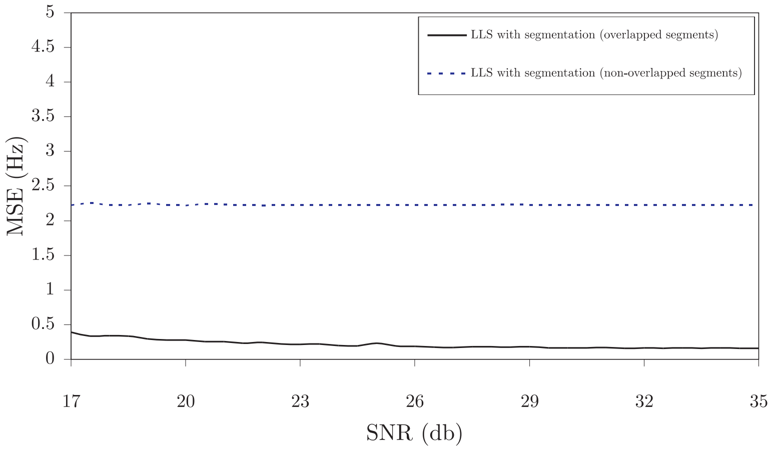


Figure 4.7: LLS algorithm testing

In the following we test the ability of the LLS algorithm to extract the instantaneous frequency. Testing methodic and the modeled data are the same we used to test the LMS algorithm in the previous section.

In all tests the LLS algorithm without segmentation did not extract the instantaneous frequency correctly even at high polynomial order p for all noise levels. It can be explained in terms of the non-trivial behavior of the modeled instantaneous frequency given in Figure 4.2(c).

Applying the segmentation technique we achieve better performance of the LLS algorithm. However in the tests the LLS algorithm with segmentation did not show high resistance to noise. It only retrieves good estimation of the instantaneous frequency at about SNR = 17db and higher.

In general the segmentation can be done in two ways. The first, is co-called overlapped segmentation (given in Definition 4.4.2). In overlapped segmentation the window is shifted by one sample. Such segmentation increases both the computational time and resolution of the estimated frequency. The second, is co-called non-overlapped segmentation. In non-overlapped segmentation the window slides by its length what can significantly decrease the frequency resolution . In Figure 4.7 the average MSE difference between curves (overlapped and non-overlapped segmentation) is about 2Hz. Such a value is significant since it may be equivalent up to 10% of overall instantaneous frequency range variation. The overlapping segmentation is particulary efficient for long data segments. In the following we always use the segmentation technique for instantaneous frequency extraction.

4.4.4 Polynomial Phase Difference IF Estimator PPD

The PPD technique also belongs to polynomial phase modeling algorithms. It can be considered as the advanced version of the conventional *phase-difference* technique for instantaneous frequency estimation. Its main advantage is high noise resistance. The phase difference technique is entirely based on equation (4.4), where $\boldsymbol{\phi}$ is the unwrapped phase. This equation requires computation of the time derivative of $\boldsymbol{\phi}$. The approximation of the derivative in the case of discrete signal can be obtained by using *finite differences*. The most appropriate form of these is the *central finite difference* (CFD) [52]. The central finite difference $\mathbf{f}^*$ (i.e. the instantaneous frequency) can be derived for every sample of an instantaneous unwrapped phase $\boldsymbol{\phi} \in \mathbb{R}^n$ for all $i \in [0 : n-1]$ and sampling interval $\triangle t$ as:

$$\mathrm{f}_i^* = \begin{cases} \frac{1}{2\pi}\frac{1}{\triangle t}\sum\limits_{j=0}^{q} \mathrm{b}_j\,\phi_{i-q+j} & \text{if} \quad q/2 \le i \le n - q/2 - 1 \\ 0 & \text{otherwise} \end{cases}, \tag{4.24}$$

where q is a derivative order, and $\mathbf{b} \in \mathbb{R}^{q+1}$ is a vector of derivative coefficients (see Table below). The order of CFD is assumed to be even i.e. $q \in [2, 4, \ldots]$. The coefficients of CFD $\mathbf{b}$ can be computed from discrete Taylor expansion for particular q, for reference see [59]. Below we represent derivative coefficients for the first three even CDF's orders:

order q	vector of coefficients $\mathbf{b}$						
2	$-\frac{1}{2}$	0	$\frac{1}{2}$				
4	$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$		
6	$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$

With high q an instantaneous frequency becomes more exactly evaluated. In practice CFD of 4-th or 6-th order is preferable. Increasing of q above does not affect the result, rather $q/2$ boundary samples become undefined, for details see equation (4.24).

The idea of PPD technique is to find a vector of real-valued coefficients $\mathbf{a} = (\mathrm{a}_1, \mathrm{a}_2 \ldots, \mathrm{a}_p)^T$ such that an instantaneous frequency $\mathbf{f} \in \mathbb{R}^n$ is defined for all $i \in [0 : n-1]$ as

$$\mathrm{f}_i = \mathrm{a}_1 + 2\,\mathrm{a}_2\,i + 3\,\mathrm{a}_3\,i^2 + \ldots + p\,\mathrm{a}_p\,i^{p-1}, \tag{4.25}$$

where p is the polynomial order (similarly to LLS algorithm). The problem of estimation of the parameters $\mathbf{a}$ is solved in [52] as:

$$\mathbf{a} = (\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{C}^{-1}\mathbf{f}^*), \tag{4.26}$$

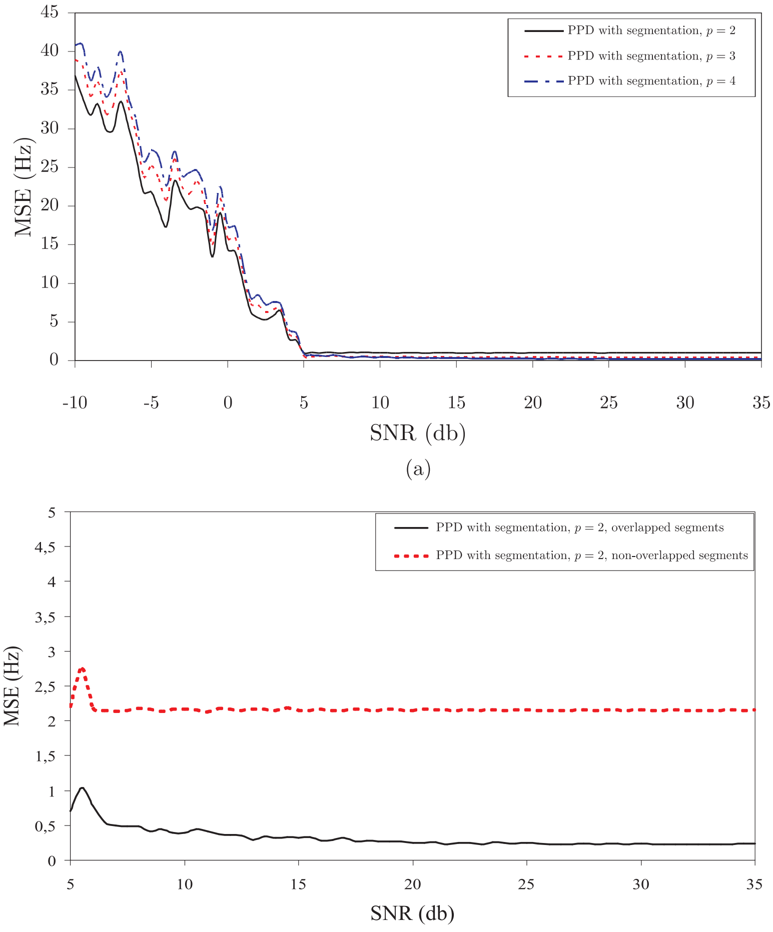
where $\mathbf{C}$ and $\mathbf{X}$ are known as *covariance* and PPD-*core* matrixes, respectively.

Similarly with the LLS algorithm a segmented version of PPD is preferable, see Section 4.4.3. This requires reformulation of $\mathbf{C}$ and $\mathbf{X}$ for the particular data segment size w:

$$\mathbf{C}_{i,j} = \begin{cases} \sum\limits_{k=0}^{q-|j-i|} \mathrm{b}_k\,\mathrm{b}_{k+|j-i|} & \text{if} \quad |j-i| \le q \\ 0 & \text{otherwise} \end{cases}, \tag{4.27}$$

where $\mathbf{C} \in \mathbb{R}^{(w\times w)}$ and $i, j \in [0 : w-1]$. A PPD-core matrix $\mathbf{X} \in \mathbb{R}^{(w\times p)}$ is defined for $i \in [0 : w-1]$ and $j \in [1 : p]$ as

$$\mathbf{X}_{i,j} = j \cdot (i)^{j-1}. \tag{4.28}$$



(a)

(b)

Figure 4.8: PPD algorithm testing: (a) general testing of the PPD algorithm; (b) testing of the overlapped and non-overlapped segmentation

We do not introduce a complete definition of PPD algorithm since it can be done analogously with LLS algorithm, see Definition 4.4.2. Here LLS function is directly substituted by PPD function. We only notice that in the case of segmented PPD equation (4.26) uses instead of full $\mathbf{f}^*$ its particular data segment of length w.

The only difference between PPD and LLS techniques is that in PPD an instantaneous frequency is fitted by the polynomial where its approximate value is taken from finite derivative. In the LLS algorithm computation of instantaneous frequency is not required. Instead, the unwrapped phase is approximated by a polynomial.

In Figure 4.8(a) we test the PPD algorithm with segmentation. For SNR higher than 5db it estimates an instantaneous frequency very well. We have examined PPD for different polynomial orders (namely 2, 3 and 4). Deviation of the estimated IF from the original one is 1, 0.45, and 0.3Hz, respectively. The following increasing of the polynomial order degrades IF estimation.

In Figure 4.8(b) we examine the PPD algorithm for overlapped and non-overlapped segmentation. Similarly with the LLS algorithm the PPD with overlapped segmentation shows the highest frequency resolution. The advantage of the overlapped segmentation over non-overlapped one was observed for all polynomial orders.

Apart from PPD and LLS there are two more algorithms which belong to the group of polynomial phase modeling techniques. These are *ML Based Polynomial Coefficient Estimation Procedure* (MLBP) and *Discrete Polynomial Phase Transform* (DPPT). MLBP is referred to as an extremely high-cost technique. It is also very difficult to adjust its parameters. Because of this we do not discuss MLBP in this work. For more information reader can find its description in the following literature [52, 59, 60] and [58, pages 45-48]. The DPPT algorithm will be defined and examined in Section 4.4.6.

4.4.5 Time-Frequency Distribution (TFDs)

Generally, in signal processing there are two classical representations of a signal. These are time-domain representation $x : \mathbb{R} \to \mathbb{C}$ where the argument of x is time t and the *frequency-domain* representation given by the Fourier Transform as $\mathcal{F}_x : \mathbb{R} \to \mathbb{C}$, where its argument is frequency. The purpose of TFDs is to combine time and frequency representations so that a signal can be expressed in terms of a function which takes time and frequency as its arguments. Let us denote $p_x : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ to be a TFD of a given signal x. Such a two-dimensional representation of a signal might be very helpful in extraction of its different characteristics. These are instantaneous frequency, number of frequency components which correspond to particular time etc.

There is a number of properties of TFDs which are studied in detail in [61, page 60]. We are especially interested in two of them. The first is the *energy conservation* property:

$$\mathcal{E}_{\mathbf{x}} = \int\limits_{-\infty}^{+\infty} |x(t)|^2\, dt \; = \int\limits_{-\infty}^{+\infty} |\mathcal{F}_x(f)|^2\, df \; = \int\limits_{-\infty}^{+\infty}\int\limits_{-\infty}^{+\infty} p_x(t,f)\; dt\, df\,, \tag{4.29}$$

where x is an analytic signal, $\mathcal{E}_{\mathbf{x}}$ is its energy, and p_x is some TFD of a signal. The first two terms of equation (4.29) represent conservation of energy of a signal in time and frequency domains, respectively. The latter term implies the energy conservation property of any Time Frequency Distribution of a given signal.

A typical Doppler signal and its TFD[5] are represented in Figure 4.9(a) and (b), respectively. Here we can see that the major part of the energy is concentrated inside some area. Its size is determined by time and frequency intervals. Thus, in time interval $[t_1 : t_2]$ the signal is non-zero. The frequency interval $[f_1 : f_2]$ includes all the frequencies which participate in the signal formation.

The second property of TFD we are interested in is the instantaneous frequency determination. For continuous analytic signals x, their instantaneous frequency

[5] this TFD was computed by using Time-Frequency Toolbox [62]

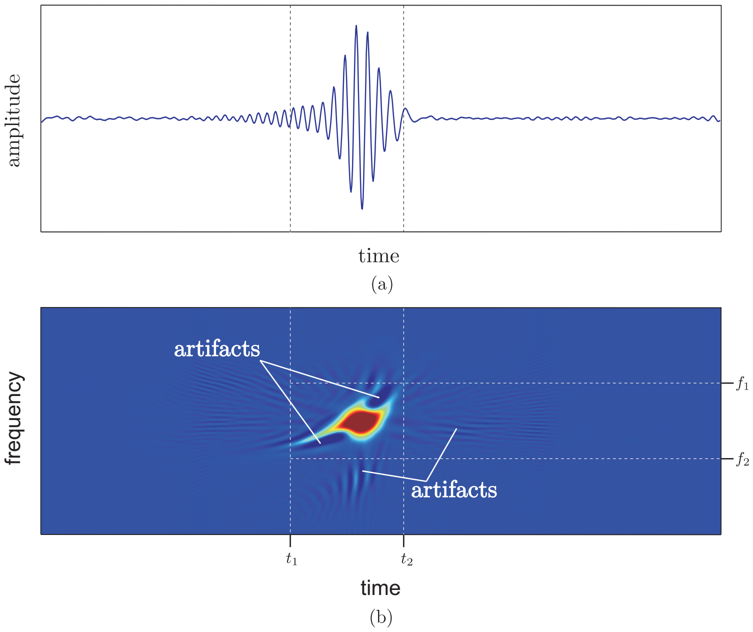


Figure 4.9: TFD distribution: (a) real part of analytic Doppler signal: Re($\mathbf{z}$); (b) TFD of $\mathbf{z}$

f_x is defined as the *first moment* of the particular TFD [63]:

$$f_x(t) = \frac{\int\limits_{-\infty}^{+\infty} f\, p_x(t,f)\, df}{\int\limits_{-\infty}^{+\infty} p_x(t,f)\, df} \tag{4.30}$$

In literature it has been shown that a great weakness of TFDs is a presence of *artifacts*, see Figure 4.9(b). The main reason why artifacts appear can be explained through the quadratical nature of TFDs (see definitions of TFDs below). Let the analyzed signal x be a sum of two elementary signals a and b. The TFD computes the square of the input signal such that $x^2 = (a+b)^2 = a^2 + b^2 + 2ab$. Thus the result of this is a three-term equation such that the first two terms are simply squared elementary signals. The third term (also called *cross term*) introduces multiplication of this signals what leads to degradation of the TFD, i.e. artifacts.

Because of presence of cross-terms the instantaneous frequency computation through equation (4.30) may be unreliable. Instead, it is suggested to estimate instantaneous frequency as follows.

Definition 4.4.3 *Let us define the* ***maximum*** **TFD** *function*

$$\text{maxTFD} : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^n$$

Let $\mathbf{P_z} \in \mathbb{R}^{m\times n}$ *be a* TFD *where m and n stand for number of frequency and time* *samples, respectively. We compute the instantaneous frequency* $\mathbf{f} \in \mathbb{R}^n$ *is given as*

$$\begin{aligned}
&\mathbf{f} = \mathrm{maxTFD}(\mathbf{P_z}) \\
&\text{where} \quad \text{for all } i \in [0 : n-1] \\
&\mathrm{f}_i = j \quad \text{such that} \quad \text{for all } j, k \in [0 : m-1] \;\text{ and }\; k \neq j \\
&\mathrm{p}_{\mathbf{z}_{j,i}} > \mathrm{p}_{\mathbf{z}_{k,i}}
\end{aligned}$$

The latter definition represents the search of such a frequency j at every time instant i of the TFD so that it exhibits maximal value. In all experiments instantaneous frequency estimation with function **maxTFD** showed higher stability and reliability than with equation (4.30).

The most popular group of TFDs belongs to Cohen's class of distributions. A basic TFD which is usually under consideration is the Wigner Distribution (WD) called in honor of its discoverer. Let us discuss the principle of detection of the instantaneous frequency of a unit-amplitude analytic signal by means of the WD.

The unit-amplitude analytic signal is defined as:

$$z(t) = e^{j\phi(t)},$$

where ϕ is an instantaneous phase (for reference see equation (4.3)). The instantaneous frequency of z is given as

$$f(t) = \frac{1}{2\pi}\frac{\partial \phi}{\partial t}. \tag{4.31}$$

In [61] it is suggested to use the following approximation of the derivative $\partial\phi/\partial t$:

$$\frac{\partial \phi}{\partial t} \approx \frac{1}{\tau}\left[\phi\left(t+\frac{\tau}{2}\right) - \phi\left(t-\frac{\tau}{2}\right)\right], \quad \text{where} \;\; \tau \in \mathbb{R} \tag{4.32}$$

The Wigner Distribution $p_z : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ of the signal z is defined in [61, pages 30-31] as:

$$p_z(t, f') = \int\limits_{-\infty}^{+\infty} e^{j2\pi f(t)\tau} \; e^{-j2\pi f'\tau} d\tau \tag{4.33}$$

By substitution equations (4.31) and (4.32) into equation (4.33) we derive:

$$\begin{aligned}
p_z(t, f') &= \int\limits_{-\infty}^{+\infty} e^{j\left[\phi\left(t+\frac{\tau}{2}\right)-\phi\left(t-\frac{\tau}{2}\right)\right]} \; e^{-j2\pi f'\tau} d\tau \\
&= \int\limits_{-\infty}^{+\infty} z(t+\frac{\tau}{2})\, z^*(t-\frac{\tau}{2}) \; e^{-j2\pi f'\tau}\, d\tau
\end{aligned} \tag{4.34}$$

where $z^*(t)$ denotes complex conjugate number for $z(t)$.

An idea behind equation (4.34) is the following. The integral over all τ for the Wigner Distribution p_z at specific instance (t, f') has the highest value if the instantaneous frequency $f(t)$ is equal to f' (see equation (4.33)). This consideration holds for all instances $t, f' \in \mathbb{R}$.

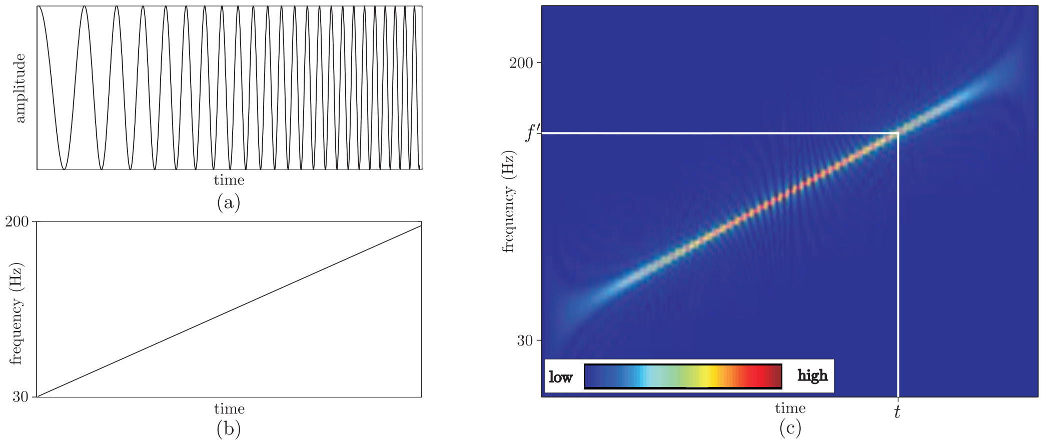


Figure 4.10: An example of the Wigner Distribution of the signal with liner-varying instantaneous frequency: (a) a real part of the unit-amplitude artificial signal s; (b) the linear-varying instantaneous frequency of the signal s of range from 30 Hz to 200 Hz; (c) the Wigner Distribution p_s of the signal s

In Figure 4.10(a) we introduce an example of some unit-amplitude artificial signal $s : \mathbb{R} \to \mathbb{C}$. This signal has a linear-varying frequency $f : \mathbb{R} \to \mathbb{R}$ of range form 30 Hz to 200 Hz, which is shown in Figure 4.10(b). In Figure 4.10(c) we represent the Wigner Distribution p_s of the signal s. Here we can see that $p_s(t, f') > p_s(t, f'')$ for all $f', f'' \in \mathbb{R}$ such that $f' \neq f''$ and $f' = f(t)$. The latter inequality makes it possible to determine the instantaneous frequency of the given signal from its Wigner Distribution.

In the literature it was shown that equation (4.34) has significant limitations. This includes inability of WD to estimate non-linear instantaneous frequency. Generally, WD shows low frequency resolution and high sensitivity to the cross-terms. These limitations make difficult estimation of the instantaneous frequency of a real Doppler signal, for reference see [64]. In order to increase performance of WD, a natural solution is to apply smoothing windows for both frequency and time. It has given reason for the formulation of *Pseudo Wigner-Ville* (PWV) and smoothed Pseudo Wigner-Ville (SPWV) distributions. In the following we will study different TFDs by example of PWV and SPWV since other known distributions are defined in a similar ways. We will also compare performance of the distributions in instantaneous frequency estimation on real and modelled Doppler data.

As we have already mentioned, PWV can be derived from (4.34) by extension with *frequency smoothing window* in the time-domain. In signal processing there is a grate number of already defined windows. They have different shape and can be used to adjust the smoothing to the analyzed data. In Figure 4.11(a) and (b) we introduce the Hamming and the flat top window, respectively. As we can see, for example, the Hamming window is only positive defined, whereas the flat top windows also has negative values. The reader can find more information about smoothing windows and their application in [65]. In this this work we used the Hamming windows (Figure 4.11(a)) because it helped to reduce the artifacts the

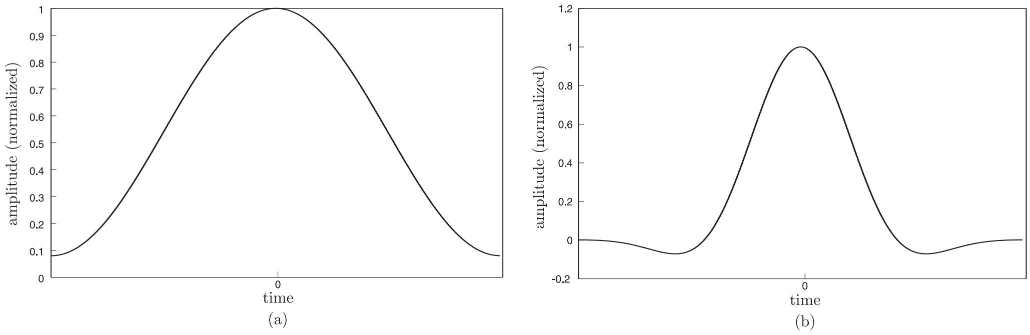


Figure 4.11: An example of smoothing windows (a) the Hamming window; (b) the flat top window

best among the other windows.

By taking in to account the frequency smoothing window h in time domain the PWV distribution is defined as:

$$p_z^{PWV}(t,f) = \int_{-\infty}^{+\infty} h(\tau)\, z(t+\frac{\tau}{2})\, z^*(t-\frac{\tau}{2})\, e^{-2\pi f\tau}\, d\tau \tag{4.35}$$

In (4.35) the range of τ is not $[-\infty : +\infty]$ rather is limited by the width of the window function h. This increases frequency resolution at every time t when $h(\tau)$ is large, by letting values of the signal z be irrelevant outside of the window h. A detailed description of the windowing technique is given in [65].

In the following we show that PWV is a good choice for a relative high SNR. If a signal to noise ratio is low, the Smoothed Pseudo Wigner-Ville distribution is preferable. SPWV distribution is defined as

$$p_z^{SPWV}(t,f) = \int_{-\infty}^{+\infty} h(\tau) \int_{-\infty}^{+\infty} g(t'-t)\, z(t'+\frac{\tau}{2})\, z^*(t'-\frac{\tau}{2})\, dt' \quad e^{-2\pi f\tau}\, d\tau \tag{4.36}$$

The equation (4.36) represents smoothing both in frequency and time, where the smoothing window g is a *time smoothing window* in the time-domain. In this work we also let the window function g to be the Hamming window. The time-smoothing window g works as follows. For a specific τ an averaging-like procedure is performed for a number of samples of z around t. Then the frequency smoothing is applied (see function h). Such a two-way smoothing technique may sufficiently reduce the influence of the noise to an instantaneous frequency. Let us examine it by the following example. In Figure 4.12(a), (b), and (c) we present WD, PWVD, and SPWVD of a modelled Doppler signal. The signal is corrupted by a white gaussian noise of 5db. The WD also looks heavily corrupted. It looks unlikely to be possible to estimate the instantaneous frequency correctly. In Figure 4.12(b) the energy of the signal becomes more apparent but it is still corrupted by the noise. Already at SNR of about 0db PWV distribution fails to estimate Doppler frequency. The SPWV distribution, see Figure 4.12(c), shows the best

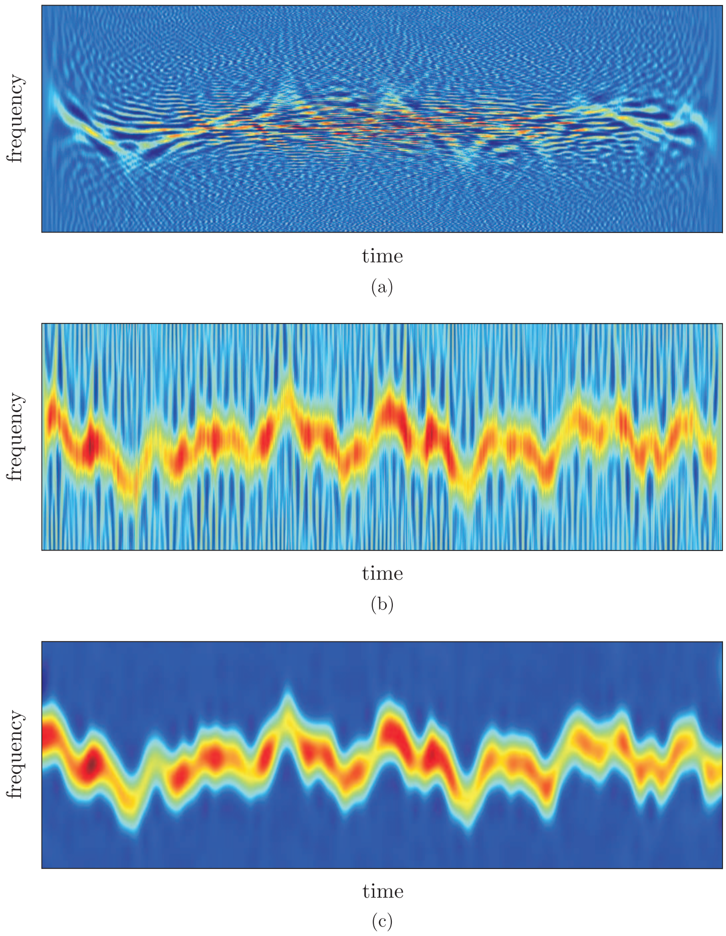


Figure 4.12: Comparison between Wigner-family distributions: (a) the Wigner distribution; (b) PWV distribution; (c) SPWV distribution

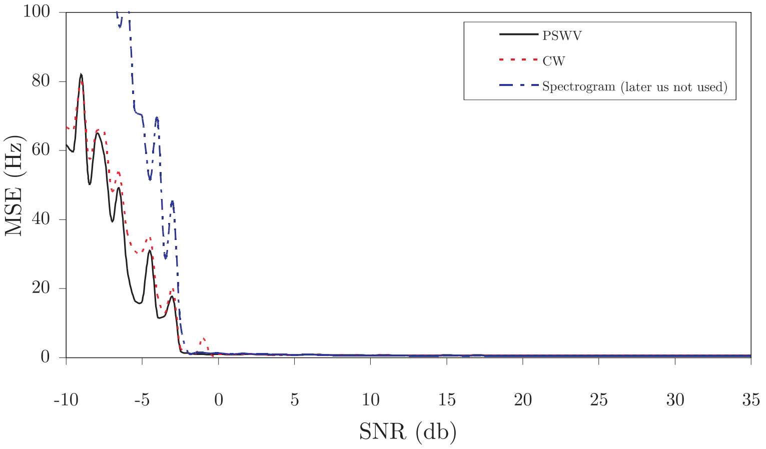


Figure 4.13: Performance in noise of PSWV, CWD, and Spectrogram distributions

performance for noise reduction. Later we will also show that time-frequency smoothing of SPWV also may set back the resolution of the estimated frequency.

On the basis of WD many similar time-frequency distributions have been developed. The only difference between them is an individual formulation of window functions. In practice, functions h and g utilize properties and physical background of the process to be investigated. For example, in [66] the *Choi-Williams* CW distribution was applied to enhance frequency images of magnetohydrodynamic models in plasma observations. It was shown that CWD offers improved frequency resolution in comparison to standard *spectrogram plots*.

Bilin Zhang and Shunsuke Sato have shown that combination of Choi-Williams and *Margenau-Hill* kernels may be successfully applied to speech signal processing. It makes possible to eliminate some former inevitable cross-terms [67].

In practice, in order to solve a specific task different TFDs are tested. In this work we do not discuss all known distributions in detail. This information can be found in [61]. In this work we will examine a number of well-known distributions for Doppler IF extraction with different SNR. As before, for this purpose we utilize the modelled Doppler signal.

In Figure 4.13 we can see a comparison of SPWV, CW, and Spectrogram distributions. These distributions precisely estimate the instantaneous frequency even at high noise, see threshold about -2db. The average MSE from 0 to 35db is 0.76, 0.74, and 0.69Hz, respectively. There are two more distributions that have shown similar results in the simulation. These are the *Zhao-Atlas-Marks* and *Born-Jordan* distributions.

In the simulation we have discovered distributions which are less stable and have worse frequency estimation than the distributions mentioned above. These are the PWV and Pseudo Page (PP) distributions, see Figure 4.14. We can see that PWV and PP are not as stable as SPWV. A simulation shows that the instantaneous frequency cannot be correctly estimated up to SNR $\approx$ 5db. An average MSE from 5db to 35db is 0.5Hz and 1.7Hz for PWV and PP, respectively. Apparently, PSWV represents the best performance. By comparing PSWV and PWV we notice that smoothing in time and frequency domains increases stability

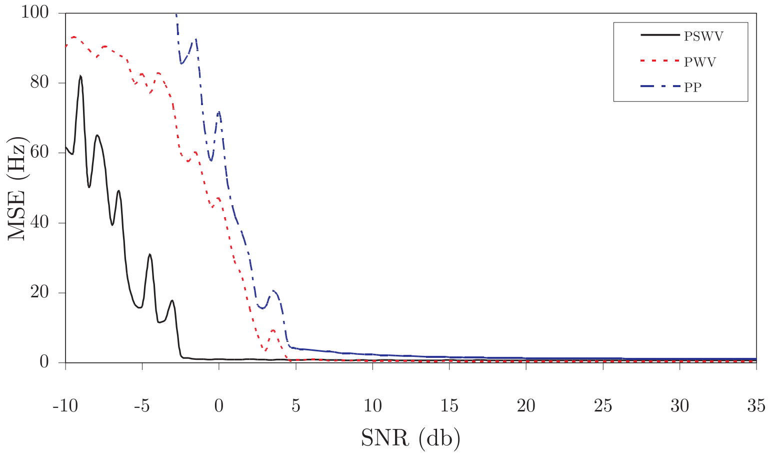


Figure 4.14: Performance in noise of PWV and PP distributions

but decreases frequency resolution.

4.4.6 Polynomial Phase Transform PPT

The Polynomial-Phase Transform has been discovered in the earlier 90-th. It was suggested to use it in different engineering applications including analysis of radar signals [68]. The PPT was introduced as a mean to estimate IF of polynomial phase signals with constant amplitude. A definition of such a signal is given below:

$$\mathrm{z}_i = b_0\, e^{j\phi_i} = b_0\, e^{j \sum\limits_{m=0}^{p} \mathrm{a}_m\, (\triangle t\, i)^m}, \tag{4.37}$$

where $\triangle t$ is a sampling interval, b_0 is an amplitude, $\boldsymbol{\phi}$ is a discrete polynomial instantaneous phase, and polynomial order p determines a number of polynomial coefficients $\mathbf{a} = (\mathrm{a}_1, \ldots, \mathrm{a}_p)$. The task of the PPT transform is to find polynomial coefficients $\mathbf{a}$ if the measured signal $\mathbf{z}$ is known and then to compute the instantaneous phase using the vector $\mathbf{a}$. Before the PPT transform is applied, it is assumed that p is known [69].

Let us define a *polynomial operator* $\mathcal{DP}$ of order p as

$$\mathcal{DP}^{p,\tau,i}(\mathbf{z}) = \prod_{k=0}^{p-1} \left(\mathrm{z}^{\blacktriangle}_{i-k\tau}\right)^{\binom{p-1}{p}}, \tag{4.38}$$

where

$$\binom{K}{k} = \frac{K!}{k!(K-k)!}.$$

In (4.38) the black triangle denotes a mapping of i-th element as

$$\mathrm{z}^{\blacktriangle}_i = \begin{cases} \mathrm{z}_i & \text{if} \quad k \ \text{even} \\ \mathrm{z}^*_i & \text{otherwise} \end{cases}, \tag{4.39}$$

and z_i^* denotes the complex conjugate for z_i. If in equation (4.39) index i is negative, then we reassign value z_i as follows

$$\mathrm{z}_i = \begin{cases} \mathrm{z}_i & \text{if} \quad i \in [0 : n-1] \\ 0 & \text{otherwise} \end{cases}$$

In equation (4.38) parameter $\tau \in \mathbb{N}$ is called a time delay. Its optimal value was suggested in [70, p.36] as $\tau = n/p$. At such values of τ PPT reaches its highest noise resistance.

Let us define the PPT as the Fourier transform of a vector $\mathbf{v}$ as:

$$\boldsymbol{\mathcal{DPT}}^{\mathbf{z},\mathbf{p},\boldsymbol{\tau}} = \hat{\mathcal{F}}_{\mathbf{v}}, \tag{4.40}$$

where vector $\mathbf{v}$ is defined by using polynomial operator for given polynomial order p and $h = (p-1)\tau$ as:

$$\mathbf{v} = \left(\mathcal{DP}^{p,\tau,h}(\mathbf{z}),\, \mathcal{DP}^{p,\tau,(h+1)}(\mathbf{z}),\, \dots\, , \mathcal{DP}^{p,\tau,(n-1)}(\mathbf{z})\right)$$

The PPT of a polynomial order p defined in equation (4.40) is used to compute p-th polynomial coefficient of the vector $\mathbf{a}$ (i.e. a_p) as follows. Let n' be the length of the vector $\mathbf{v}$ at given polynomial order p and

$$\boldsymbol{\mathcal{DPT}}^{\mathbf{z},\mathbf{p},\boldsymbol{\tau}} = \left(\mathcal{DPT}_0^{z,p,\tau}, \mathcal{DPT}_1^{z,p,\tau}, \dots, \mathcal{DPT}_{n'-1}^{z,p,\tau}\right).$$

Let the index $k' \in [0 : n'-1]$ maximizes the vector $\mathbf{abs}(\boldsymbol{\mathcal{DPT}}^{\mathbf{z},\mathbf{p},\boldsymbol{\tau}})$ such that for all $i \in [0 : n'-1]$ and $k' \neq i$ holds

$$\mathbf{abs}\left(\mathcal{DPT}_{k'}^{z,p,\tau}\right) > \mathbf{abs}\left(\mathcal{DPT}_i^{z,p,\tau}\right) \tag{4.41}$$

We define the index $k \in \mathbb{N}$ which introduce explicitly whether it corresponds to negative or positive frequency as

$$k := \begin{cases} k' & \text{if} \quad k' \leq \left\lfloor \frac{n'}{2} \right\rfloor \\ -(n'-k') & \text{otherwise} \end{cases}$$

By using k determined from the latter equation we can compute a_p as follows:

$$\mathrm{a}_p = \frac{2\pi k}{n\; p!\; (\triangle t\, \tau)^{p-1}}. \tag{4.42}$$

In order to find all the polynomial coefficients from vector $\mathbf{a}$ we proceed recursively in the following way. The highest coefficient a_p, from the initial signal $\mathbf{z}$, is computed as we discussed above. Then, the signal $\mathbf{z}$ is reassigned according to so-called *de-chirping* procedure which is given in (4.43) for $i \in [0 : n-1]$ as

$$\begin{aligned} \mathrm{z}_i &:= \mathrm{z}_i\, e^{-j\mathrm{a}_p(\triangle t\, i)^p} \\ &= b_0\, e^{j \sum\limits_{m=0}^{p} \mathrm{a}_m\, (\triangle t\, i)^m - j\mathrm{a}_p(\triangle t\, i)^p +} \\ &= b_0\, e^{j \sum\limits_{m=0}^{p-1} \mathrm{a}_m\, (\triangle t\, i)^m} \end{aligned} \tag{4.43}$$

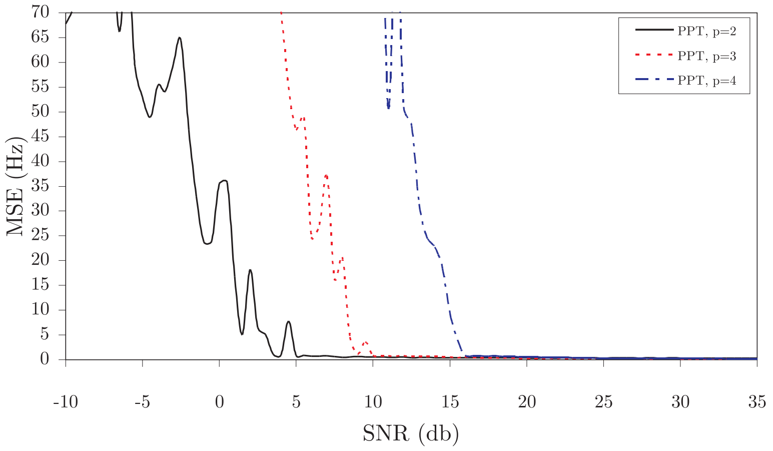


Figure 4.15: Performance of PPT in noise

According to [70] the de-chirping procedure filters out the highest coefficient a_p of the instantaneous phase $\boldsymbol{\phi}$ of signal $\mathbf{z}$ defined in equation (4.37). In order to estimate the coefficient a_{p-1} we apply equation (4.40) to the reassigned signal computed in equation (4.43). We stop this procedure when the coefficient a_1-th is known.

The vector of polynomial coefficients **a** is used to estimate the instantaneous frequency as follows

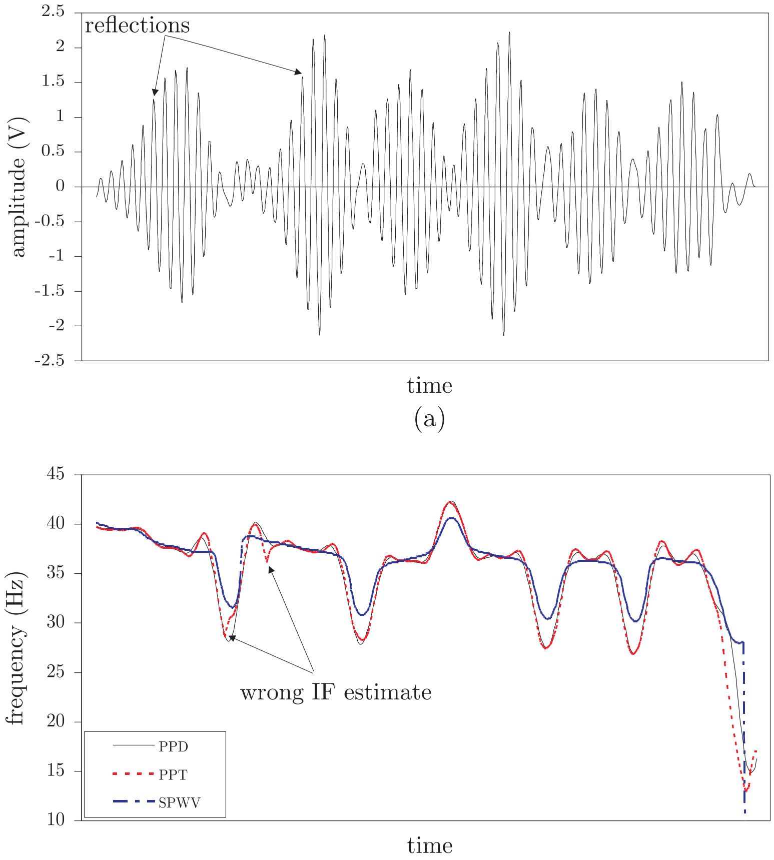
$$\mathrm{f}_i = \frac{1}{\triangle t\, 2\pi} \sum_{m=1}^{p} m\, \mathrm{a}_m (\triangle t\, i)^{m-1}. \tag{4.44}$$

As long as the instantaneous frequency has a polynomial form, the performance of PPT is good. A non-polynomial frequency behavior damages IF estimation [71]. In order to suppress errors we apply for the first time the segmentation technique as it was discussed in Section 4.4.3. In Figure 4.15 we examine the performance of the PPT algorithm in noise. PPT estimates an instantaneous frequency well above 5db noise. The average MSE at polynomial order $p = 2$ from 5db to 35db is 0.43Hz. Simulations show high sensitivity of PPT to the value of p. Increasing of the polynomial order impairs the noise resistance of the algorithm. At $p = 4$ the PPT algorithm seems unlikely to be able to estimate the instantaneous frequency correctly.

In the simulation we also showed that the segmentation technique improves frequency resolution of the PPT algorithm.

4.5 Comparative Issue

In Section 4.4 we represented a number of algorithms which can be used for extraction of instantaneous frequency from non-stationary signals. In the previous sections we discussed their mathematical description, implementation aspects, and tested their noise sensitivity on simulated data. In the current section we examine these algorithms for real data and decide which of them is suitable for Doppler frequency extraction.



(a)

(b)

Figure 4.16: Comparison of PPD, PPT, and PSWV distributions (first case): (a) Doppler signal; (b) IF estimated from the PPD, PPT, and PSWV algorithms

4.5.1 Real Signal IF Estimation

First of all we concentrate our attention on two algorithms which belong to the polynomial phase techniques. These are the PPD and PPT algorithms, see Sections 4.4.4 and 4.4.6, respectively. For relatively low polynomial order $p = 2, 3$ these algorithms showed both good noise resistance and exact frequency estimation.

Another group of algorithms based on Wigner-Ville Distribution also behaves very stable at high noise. These are PSWV, CWD, Spectrogram etc, see Sections 4.4.5. Since all these algorithms use the same basic technique and the analysis of the modeled data have not demonstrated big difference in their results we will only check the PSWV.

In Figure 4.16(a) a real Doppler signal is represented. Here we can see closely spaced oscillations caused by back-reflection from the measured surface. The instantaneous frequencies defined from PPD, PPT, and PSWV are given in Figure 4.16(b). Here PPD gives the best result. It is smooth and does not contain any spurious frequency alternations as, for example PPT (see sharp peaks pointed by arrows). We observed such an unreal frequency variation[6] almost in all estimations derived from the PPT technique. It happens, in particular, when a polynomial order exceeds some acceptable value. Apart from this a frequency resolution of PPD and PPT is almost equal.

The PSWV curve represents bad frequency resolution. It can be explained in terms of smoothing window (function h and g, see equations (4.35) and (4.36), respectively) in frequency and time domains. Since any window function introduces averaging between neighboring samples, then the result of the windowing will be averaged.

The example given in Figure 4.17 introduces a typical situation when PPT and PSWV may fail. In particular, it happens when the Doppler signal does not cross zero-value. In such a case the frequency of the regarded data segment is wrongly interpreted to be much lower then the expected minimal Doppler frequency, see the data segment in Figure 4.17(b) labeled as "wrong IF estimate". We remind that the expected Doppler frequency values can be computed for the given experiment according to equations (2.9) and (2.9),

In the scope of the work more than twenty unique experiments were done in different measurement conditions. The experiments also were repeated in order to exclude measurement errors. In all the experiments the PPD algorithm showed the most reliable estimation of the Doppler frequency.

4.5.2 Complexity Issue

In the previous subsection we have discovered that the PPD technique may be a good choice to estimate the instantaneous Doppler frequency. It showed the best performance in all the experiments we have performed (see previous Section). The only drawback is that it is not as stable to noise as the PSWV algorithm (for reference see Figures 4.8, 4.13 and 4.14). Generally, in practice, SNR lower than $4-5$db is not very common. Of course, with really noisy signals the only solution is the PSWV.

Let us consider the complexity of the PPD, PPT, and PSWV algorithms. The core of PPD is given in (4.26). It is a product of $\mathbf{C}$, $\mathbf{X}$, and $\mathbf{f}$, where $\mathbf{f}$ is a segment of $\mathbf{f}^*$ of size w. A covariance matrix $\mathbf{C}$ depends on the window size w and the order q of the central finite difference. In practice, w is some constant. It depends on the Doppler frequency range, which is a function of measurement conditions, ultimate defects' size etc. CFD order q is also a constant which mostly depends on SNR. Its value can be estimated off-line by using simulated data. In this work we did not notice any improvements at $q > 6$.

In order to compute $\mathbf{X}$, window length w and polynomial order p' are required, see (4.28). Usually p' is determined experimentally. In this work estimation of instantaneous frequency was successfully done at low polynomial order. Let us assume that the polynomial order is limited i.e. $p' \leq p$. Then $\mathbf{X}$ can be precomputed for constant w and all $p' \in [1:p]$.

Let us denote by $\mathbf{X}_{p'}$ a matrix $\mathbf{X}$ computed for a certain p'. Then, from (4.26),

[6] because of the nature of the frequency it is physically incorrect the frequency has sharp edges

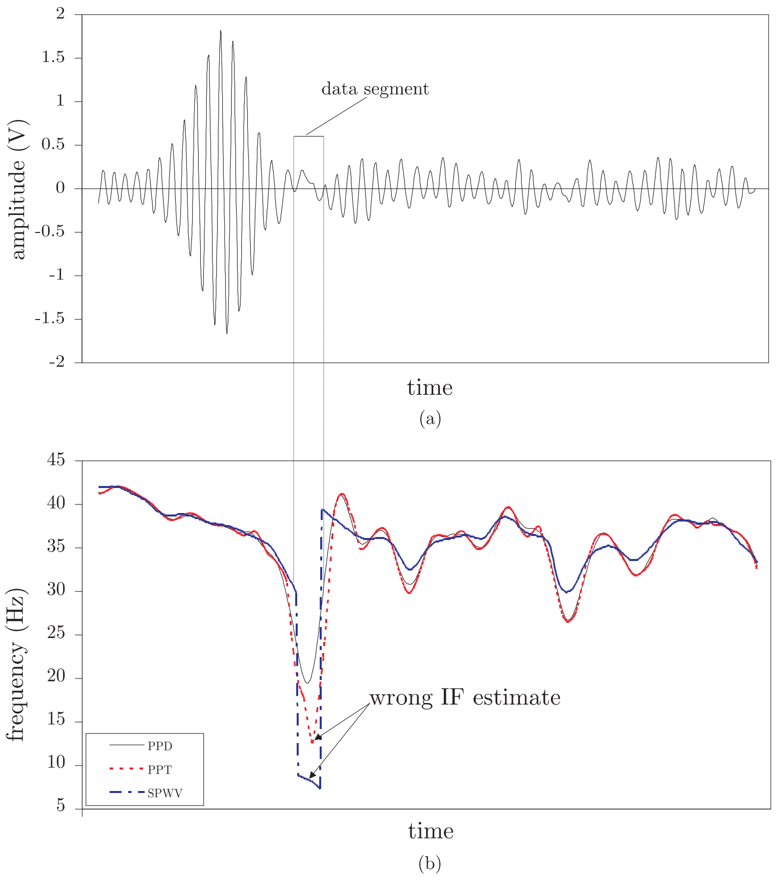


Figure 4.17: Comparison of PPD, PPT, and PSWV (second case): (a) Doppler signal; (b) IF estimated from the PPD, PPT, and PSWV algorithms

we have two products which can be computed as

$$\begin{aligned}\mathbf{A}_{p'} &= (\mathbf{X}_{p'}^{T}\,\mathbf{C}^{-1}\,\mathbf{X}_{p'})^{-1}\\ \mathbf{B}_{p'} &= \mathbf{X}_{p'}^{T}\,\mathbf{C}^{-1}\end{aligned} \tag{4.45}$$

In order to pre-save $\mathbf{A}_p$ and $\mathbf{B}_p$ for all p the memory consumption for $w > 1$ is

$$\sum_{p'=1}^{p} \left(p'^2 + w\,p'\right) \approx\; \mathcal{O}(p^3 + p^2 w) \tag{4.46}$$

As we can see at low p the memory consumption is not expensive.

Let us derive the computational cost of PPD at given polynomial order p and pre-computed $\mathbf{A}_p$ and $\mathbf{B}_p$. We assume the offset between segments to be minimal (i.e. one sample) so that the number of segments is $s = n - w + 1$. Then, the computational cost for $p < w$ is defined as:

$$s(p^2 + w\,p) \approx \mathcal{O}\left(n \cdot (p^2 + pw)\right) \tag{4.47}$$

Since $n \gg p$ and $n \gg w$, PPD is not computationally expensive.

We represent the cost of the PPT algorithm which includes computation of the $\mathcal{DPT}$ operator for data segment size w and polynomial order p. We assume that the number of segments is equal to the number of samples in the whole analyzed signal and denote it as n. Using the discrete fast Fourier transform [72] cost of the PPT with the de-chirping procedure is given as follows:

$$\mathcal{O}(2np^2 w + np\,N\log N), \tag{4.48}$$

where N denotes the number of the frequencies in the discrete Fourier Transform, see Definition 3.1.7. Since the segment size (number of samples) w is usually short, the Fourier Transform of the segment also has w frequencies. If the frequency resolution w is too low, the number of frequencies of the Fourier Transform can be increased to N such that $N > w$. This can be done by adding zero samples at the right hand size of the analyzed segment in the time domain. This procedure is called *zero-padding*. Afterwards the Fourier Transform is applied to the reassigned data segment. Usually N is relatively big so that the second term of (4.48) becomes expensive. In general PPT can be considered as sub-quadratical on N.

Generally, the computation of a Time Frequency Distribution is a difficult task. The need of efficient computation of TFD has motivated the development of a number of algorithms and techniques. A real-time implementation of Wigner-Ville distribution was discussed in [73]. It was concluded that the best solution to reach high frequency and time resolutions is the use of *parallel microprogrammed system.* Later in [74], this approach was enhanced by introducing FFT algorithms which suppress multiplications with zero-crossings which are understood as non-valuable operations. A first parallel implementation of the Choi-Williams distribution was represented in [75]. In order to achieve real-time frequency estimation up to 5 parallel processors were used. Each processor operated according to a specific computational scheme which was developed for this distribution [76].

Generally, the application of a Choi-Williams distribution for instantaneous frequency estimation is restricted by rigid h and g functions. This caused the development of fast algorithms which can handle pseudo- and reassigned versions of the Wigner-Ville distribution. Such implementations were introduced in [77]. In latter references first of all the authors discuss algorithm for the *spectrogram* method. Then, this algorithm is generalized to Cohen's class methods. It has been claimed that the algorithm requires some restrictions on windows and is difficult to be used in practice.

In the general case computation of the PSWV distribution consumes quadratical time. A possible way to speed up this algorithm is to decrease the number of time and frequency samples of the 2D image. A clever choice of the specific window functions h and g can also reduce computational demand.

By comparing computational efficiency, performance in noise, and reliability of the PPD, PPT, and PSWV algorithms we conclude that the PPD algorithm is the most efficient one for Doppler signal processing tasks. It introduces good performance on both simulated and real data. PPD also offers quite good noise resistance at very low computational cost.

4.5.3 Instantaneous Frequency Imaging

Doppler imaging as it was defined in Chapter 3 can be referred to as amplitude Doppler imaging. This is because the physical quantity we use to depict defects is the Doppler amplitude. Analogously we can define frequency Doppler imaging where instantaneous Doppler frequency is used to portray the surface of a specimen. It is estimated from the measured data using the PPD algorithm. In Figure 4.18(a) and (b) we represent amplitude[7] and frequency Doppler images at $\alpha = 135^\circ$.

Both images are corrupted with artifacts which are understood as measurement errors. These do not have much influence on amplitude image since the error waveform has low amplitude. Contrariwise, the frequency image shows great sensitivity to these artifacts. It can be explained through oscillatory behavior of the error waveform. Thus, the PPD algorithm detects frequency of both error and valid signals. In order to clear the frequency image from spurious information the amplitude image can be used. We perform this by simple re-mapping of the frequency image for all $i, j \in [0 : n-1]$:

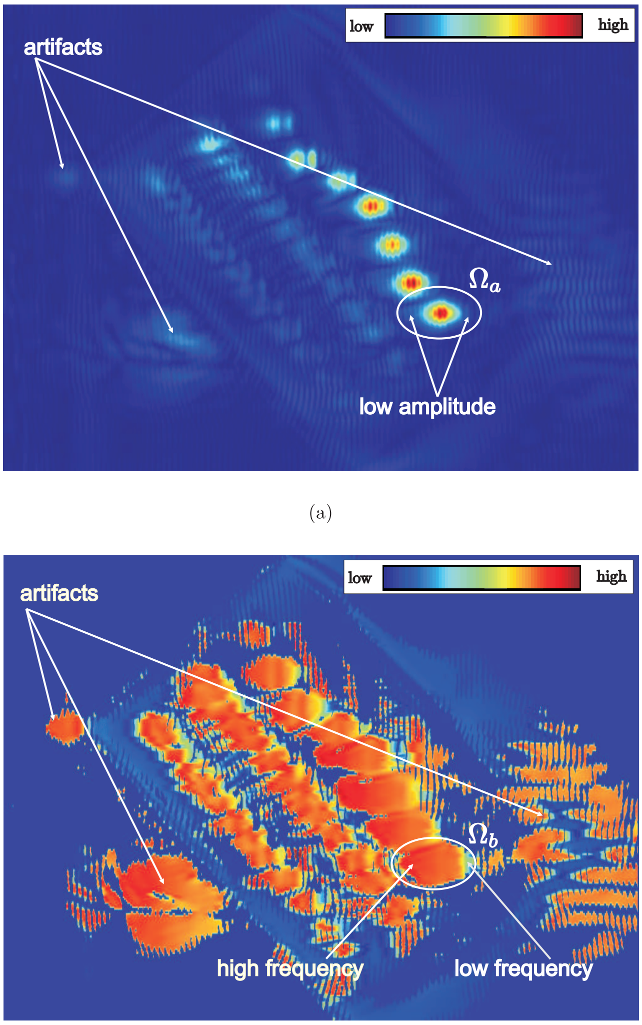
$$\mathrm{a}_{i,j} := \begin{cases} \mathrm{a}_{i,j} & \text{if} \quad \mathrm{b}_{i,j} \geq \epsilon \\ 0 & \text{otherwise,} \end{cases} , \tag{4.49}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ are frequency and amplitude images, respectively, and ϵ is some constant (i.e. threshold).

In practice equation (4.49) does not give a great difference in comparison to the amplitude Doppler image which was obtained by the thresholding technique. From the experiment introduced in Figures 4.18(a) and (b) we conclude that in general amplitude imaging is much more suitable for defect evaluation than frequency imaging.

Although, the frequency image is sensitive to noise and is not as reliable as the amplitude image, it carries some new information about defects. Let us introduce

[7] we discussed that image in the context of amplitude Doppler imaging in Chapter 3 in Figure 3.12(e)



(a)

(b)

Figure 4.18: Instantaneous Doppler frequency imaging: (a) amplitude image; (b) frequency image; (c) 1D signal from the middle of Ω_a (amplitude); (d) 1D signal from the middle Ω_b (frequency)

areas Ω_a and Ω_b which are circled in Figures 4.18(a) and (b). The Doppler signal and its envelope belonging to the middle of Ω_a are represented in Figure 4.18(c). The instantaneous frequency along the main axis of the ellipse Ω_b is given in Figure 4.18(d). As the radar starts irradiating the defect, the signal has some low amplitude, whereas the frequency is high. While the radar is passing the defect the amplitude reaches its maximal value and falls again. The frequency is steadily decreasing with moving of the radar. It drops heavily as the radar leaves the defect. We observe this effect with all defects represented in the picture, see yellow spots on the frequency image. Thus the amplitude and the frequency behavior gives us an idea that the specific combination of amplitude and frequency information can be used to improve the Doppler imaging technique. This question will be discussed in detail in Chapter 5.

4.6 Conclusion

In this chapter we introduced the concept of Doppler instantaneous frequency. We discussed its nature, conditions which cause instantaneous frequency appearance and its relation with defects. An approach for modeling of a typical Doppler signal from its instantaneous amplitude and frequency is represented. The modelled signal is used to select the most appropriate algorithm for Doppler frequency estimation. Then, we examined the performance of the algorithms on the measured data. A comparison of algorithms performance and complexity has discovered that PPD algorithm is the most efficient one for our needs.

In Section 4.5.3 we introduced Doppler frequency imaging and compared it with amplitude imaging. We discovered that frequency information alone can not be used for defect detection since it is very sensitive to spurious waveforms. It is only useful when the amplitude of the Doppler signal does not have pronounced peaks i.e. stays relatively constant. In all other cases the amplitude information is preferable. In the next chapter we consider a signal processing technique which utilizes both frequency and amplitude information. There we will discuss in detail the resolution ability of the radar and possible ways to improve it.

Chapter 5

Maximum Entropy Deconvolution Approach

5.1 Spatial resolution of CW Radar

In Chapters 3 and 4 we discussed two approaches for Doppler signal analysis. These approaches include processing of the Doppler amplitude and Doppler frequency. We have shown that the amplitude of the Doppler signal can be successfully used for detection of defects. There is a great number of industrial applications where high spatial resolution of the radar is not necessary. Often, it is good enough to make a conclusion whether a specimen contains any defects. In such applications approximate locations of defects can be found by analyzing the Doppler amplitude.

However, in general, analysis of the Doppler amplitude and frequency separately, does not allow to reach high spatial resolution. In the following we examine the resolution ability of the Doppler amplitude and frequency experimentally. We perform a series of single-line experiments for a number of defects placed along the line of scan at different distances between them. The defects are taken to be point scatterers (steel balls of diameter of 8mm). The task will be to test how well the actual distances can be computed from measured signals by analyzing Doppler amplitude and frequency. We compute the distance between two defects in the measured signal $\mathrm{x} \in \mathbb{R}^n$ as

$$\triangle t \cdot v \cdot \mathrm{abs}(i - j), \tag{5.1}$$

where x_i and x_j are samples which correspond to defects locations, $\triangle t$ is a sampling interval, and v is a speed of the radar.

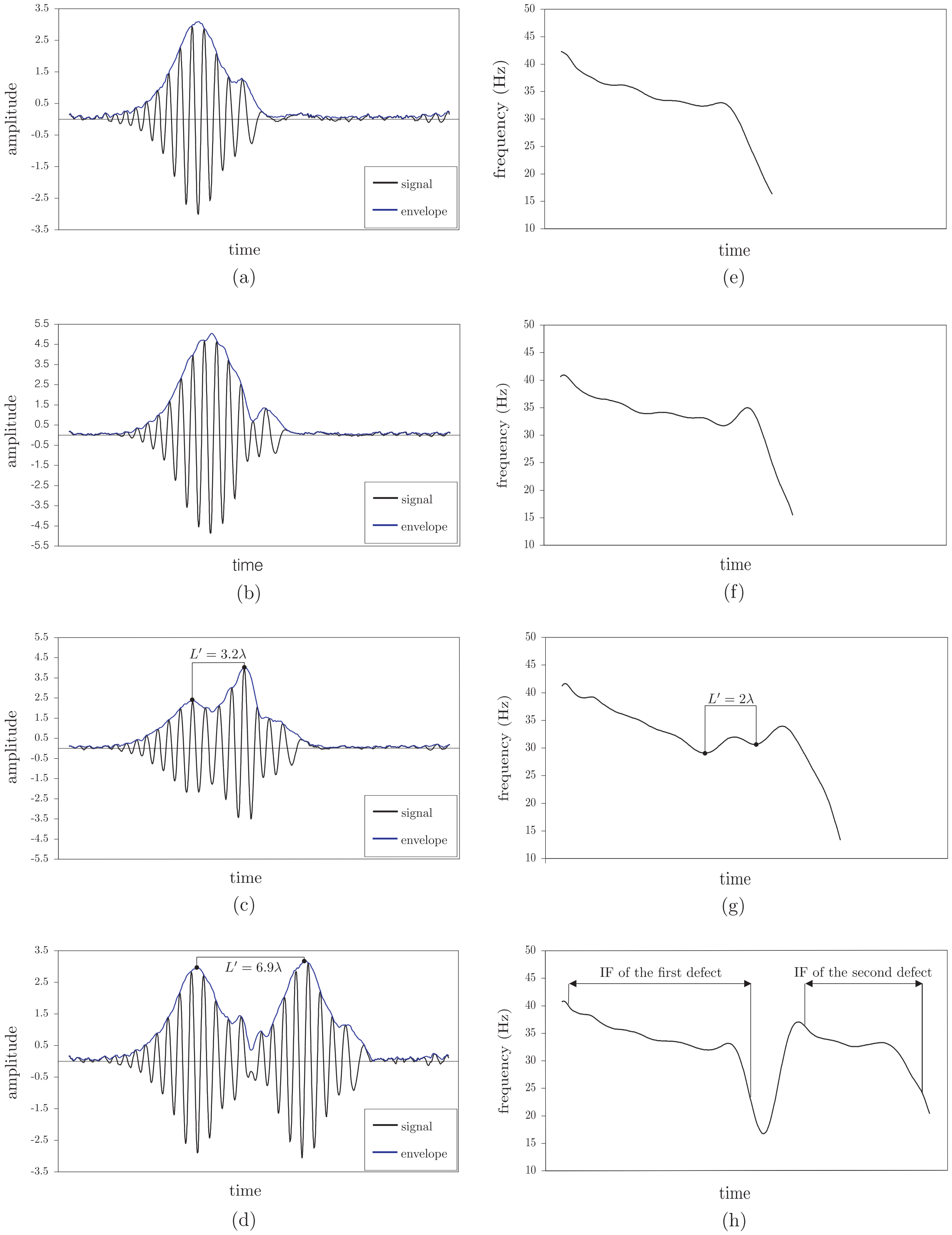
The first experiment consists of several single-line measurements for two equal defects placed along the line of scan for different distances between them. The amplitude of the measured signals at every distance (which we denote as L) will be compared with the amplitude of a so-called reference signal measured for one such defect. This comparison will help us to understand what is the influence of the second defect on the measured signal. We present the reference signal and its envelope in Figure 5.1(a), we recall that $\lambda = 12$mm denotes wavelength and depends on the value of the transmitted frequency of the radar. The Doppler signals and their envelopes at distances $L = 1.5\lambda$, 2.5λ, and 6.5λ are given in Figures 5.1(b), (c), and (d), respectively. By comparing these signals we notice that the second defect at $L = 1.5\lambda$ does not change the measured signal strongly. The differences can be noticed in amplitude rising and time-spreading. With the following increasing of L defects become more separable. Thus, in Figure 5.1(c) at $L = 2.5\lambda$ we notice two overlapped shapes indicating the presence of defects. By applying the peak search algorithm to the envelope we can evaluate the distance L' between peaks, i.e. distance between defects. Here, peak analysis does not retrieve the correct distance between defects. It exceeds the initial distance value and is equal to $L' = 3.2\lambda$. The following increasing of distance L discovers almost full spatial separation of the defects, see Figure 5.1(d). By analyzing peaks of the envelope we conclude that the distance L is also estimated incorrectly. In this case it is about $L' = 6.9\lambda$.

In the first experiment we also analyze whether the Doppler frequency (in addition to the Doppler amplitude) can be used to estimate distance between defects. We use the PPD algorithm to compute the instantaneous frequency of the signals mentioned above. Similarly to the Doppler amplitude, the Doppler frequency seems unlikely to be helpful in improvement of the radar spatial resolution. For one defect (represented in Figure 5.1(e)), we observe a typical Doppler frequency behavior. It rises first and steadily drops as the radar passes the defect. In the case of two defects at $L = 1.5\lambda$, the frequency is different, see Figure 5.1(h). Even if it rises just before falling down we can not estimate the number of defects. The Doppler frequency at $L = 2.5\lambda$ has two cavities but the estimated distance between the defects is also incorrect. Its value is about $L' = 2\lambda$. At the distance between defects $L = 6.5\lambda$ both defects are well-separated. The Doppler frequency curve in 5.1(h) is obviously (visibly) separated into two parts where each of them corresponds to IF of one defect given in Figure 5.1(e).

The second experiment is closely to the real situation. We increase the number of defects so that four defects are placed at equal distances $L = 2.5\lambda$ along the line of scan. In that way we heavily damage a measured signal making the radar to receive reflection from all the defects at the same time. The signal, its envelope, and its Doppler frequency are represented in Figure 5.2(a) and (c), respectively. In this situation only instantaneous frequency carries valuable information about defects. Here we can see four cavities so that distances between their minima almost corresponds to distances between defects.

In the third experiment we slightly change conditions, see Figure 5.2(b) and (c). Here we have three defects located at distances $L = 1.25\lambda$ (between the first and the second) and $L = 1.6\lambda$ (between the second and the third). As we can see it is not obvious to determine the number of defects and distances between them in both amplitude and frequency curves.

From the experiments represented in Figures 5.1 and 5.2 we conclude that in general, distinct analysis of Doppler amplitude and frequency can not be used for improvement of spatial resolution. As we have already discussed in Chapter 4, joint analysis of the amplitude and frequency may possibly help to overcome that problem.



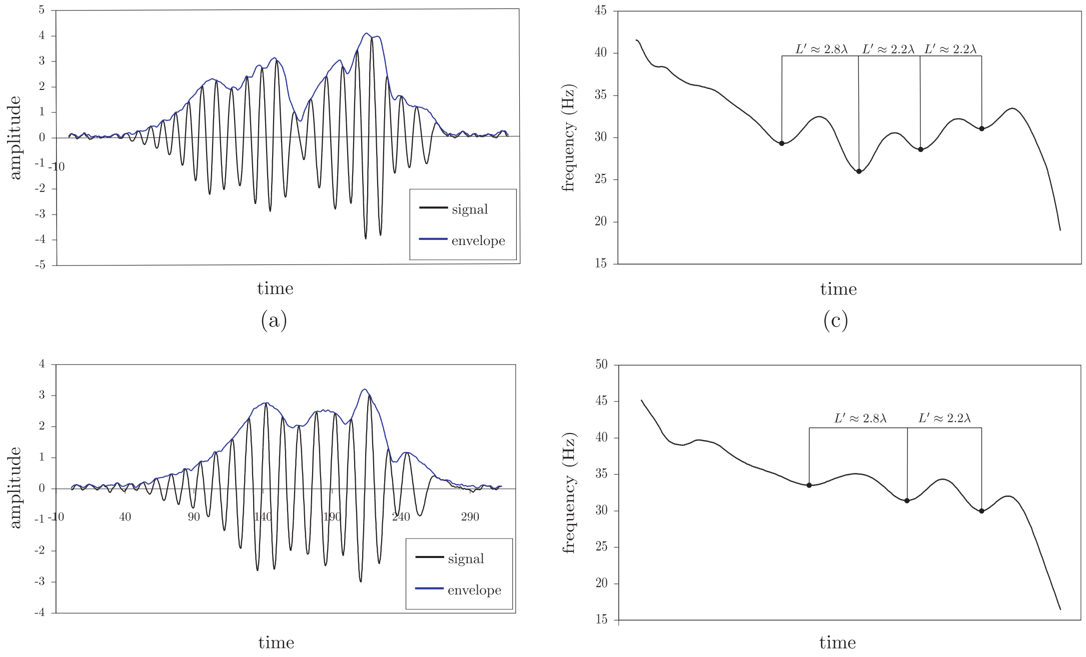
(a) (e)

(b) (f)

(c) (g)

(d) (h)

Figure 5.1: Radar resolution experiment: (a), (e) Doppler amplitude and frequency, one defect; (b), (f) Doppler amplitude and frequency, two defects, $L = 1.5\lambda$; (c), (g) Doppler amplitude and frequency, two defects, $L = 2.5\lambda$; (d), (h) Doppler amplitude and frequency, two defects, $L = 6.5\lambda$



(a) (c)

(b) (d)

Figure 5.2: Multi-defect experiment: (a), (c) Doppler amplitude and frequency, four equally spaced defects at $L = 2.5\lambda$; (b), (d) Doppler amplitude and frequency, three defects at $L = 1.25\lambda$ and $L = 1.6\lambda$

5.2 Signal Processing System

In the current section we present the approach which utilizes both the Doppler amplitude and the Doppler frequency to improve spatial resolution of the radar. Here the idea is to consider the Doppler measurement system in terms of a so-called *Finite Impulse Response* system which is characterized by its input, output, and internal functions. The input function (which describes defects) is computed from the output function (i.e. measured signal) and the internal function (also called *transfer function* or *impulse response*) summarizes internal properties of the Doppler measurement system.

Definition 5.2.1 *A **signal processing system** is any device that takes zero or more signals as input and returns zero or more signals as output.*

In the following we refer to a signal processing system as just a system. In this thesis we do not get familiar with all possible system types. Their detailed description can be found in [78,79]. We only examine a class of systems called Finite Impulse Response systems (or simply FIR). In order to illustrate the concept of FIR system we use *block diagrams*. Each diagram is mathematically defined and has a unique graphical notation. The most important block diagrams for basic operations as *unit delay*, *summation*, and *multiplication* are given below.

Definition 5.2.2 *Let* $\mathbf{b} \in \mathbb{R}^n$ *be the input of the **unit delay** operation. We define*

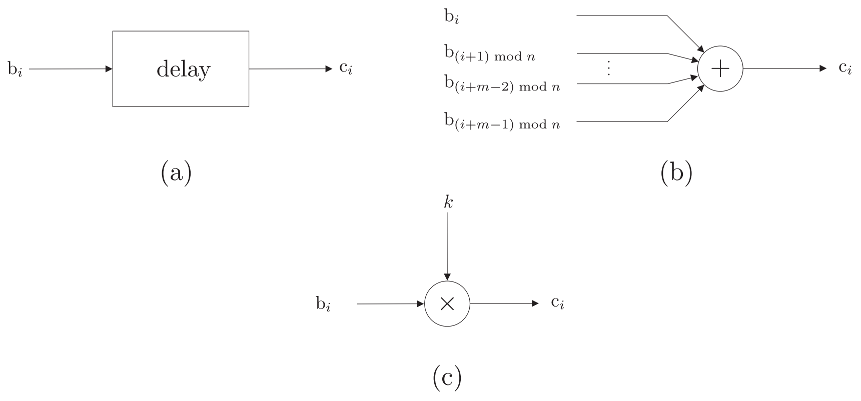


Figure 5.3: Block diagrams of fundamental operations: (a) unit delay; (b) m-inputs summation; (c) multiplication

its output $\mathbf{c} \in \mathbb{R}^n$ *to be a delayed version of* $\mathbf{b}$ *such that* for all $i \in [0 : n-1]$*:*

$$c_i = b_{(i-1 \bmod n)}$$

The block diagram of the unit delay is represented in Figure 5.3(a).

Definition 5.2.3 *Similarly to* Definition 5.2.2 *we define* ***summation*** *of* $m \in \mathbb{N}^+$ *inputs. Thus, for an input signal* $\mathbf{b} \in \mathbb{R}^n$*, an output signal* $\mathbf{c} \in \mathbb{R}^n$*, and some* $i \in [0 : n-1]$ *we have:*

$$c_i = \sum_{j=i}^{i+m-1} b_{(j \bmod n)}$$

The block diagram of m-input summation is represented in Figure 5.3(b).

Definition 5.2.4 *The* ***multiplication*** *operation of input signal* $\mathbf{b} \in \mathbb{R}^n$ *and real-valued constant* $k \in \mathbb{R}$ *is written for some* $i \in [0 : n-1]$ *as:*

$$c_i = k \cdot b_i,$$

where the corresponding block diagram is represented in Figure 5.3(c).

By using unit-delay, summation, and multiplication diagrams we construct a FIR system. Its example implementation is presented in Figure 5.4. This FIR system has one input and one output. Let the vector $\mathbf{x} \in \mathbb{R}^n$ be the input of the system. The unit delays $D_1, D_2, \ldots, D_{n-1}$ receive samples $x_\tau, x_{(\tau-1 \bmod n)}, \ldots, x_{(\tau-n+2 \bmod n)}$ for the particular index $\tau \in [0 : n-1]$. At the same time the summation operation takes for all $i \in [0 : n-1]$ products $h_i\, x_{(\tau-i \bmod n)}$, where h_i is an element of some real valued vector $\mathbf{h} \in \mathbb{R}^n$. By using definitions of unit-delay, summation and multiplication (see Definitions 5.2.2, 5.2.3, and 5.2.4) we conclude that the output of the system $\mathbf{y} \in \mathbb{R}^n$ is computed for all $\tau \in [0 : n-1]$ as:

$$y_\tau = \sum_{i=0}^{n-1} h_i \cdot x_{(\tau-j \bmod n)} \tag{5.2}$$

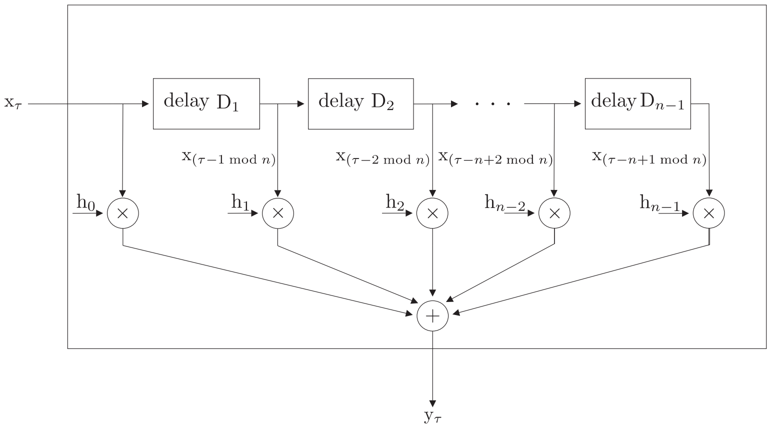


Figure 5.4: FIR system block diagram

From (5.2) it is clear that a FIR system uses the current and $n-1$ previous samples of the input signal to compute a new instance of the output signal. The vector $\mathbf{h} = (h_0, h_1, \ldots, h_{n-1})$ entirely characterizes a FIR system, i.e. it determines the rule of transforming the input into the output. In the literature $\mathbf{h}$ is referred to as the *impulse response* or *transfer function* of the FIR system.

In the literature the equation (5.2) is known as positive wrapped convolution, for reference see [80, page 72], and is written for $\tau \in [0 : n-1]$ as:

$$y_\tau = \sum_{i=0}^{\tau} h_i x_{\tau-i} + \sum_{i=\tau+1}^{n-1} h_i x_{n+\tau-i} \equiv (\mathbf{h} * \mathbf{x})_\tau \tag{5.3}$$

In the latter equation we use symbol $(*)$ to denote positive wrapped convolution.

Corollary 5.2.5 *The* ***impulse response*** *of a* FIR *system is observed at its output if the input is an unit sample* [81].

Let us apply the unit sample $\boldsymbol{\delta} \in \mathbb{R}^n$ defined for all $\tau \in [0 : n-1]$ as

$$\delta_\tau = \begin{cases} 1 & \text{if} \quad \tau = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

to the input of the FIR system (i.e. $\mathbf{x} = \boldsymbol{\delta}$), see Figure 5.4. By substitution of $\mathbf{x}$ into equation (5.3) we observe the impulse response at the output $\mathbf{y}$ of the FIR system for all $\tau \in [0 : n-1]$ as:

$$y_\tau = h_\tau$$

5.3 Deconvolution Approach

In the following sections we will study the approach (which we later call deconvolution approach) to compute the input of the FIR system from its output and impulse response. We remind that the input of the FIR system is a signal which clearly determines positions of defects. For deconvolution approach we assume

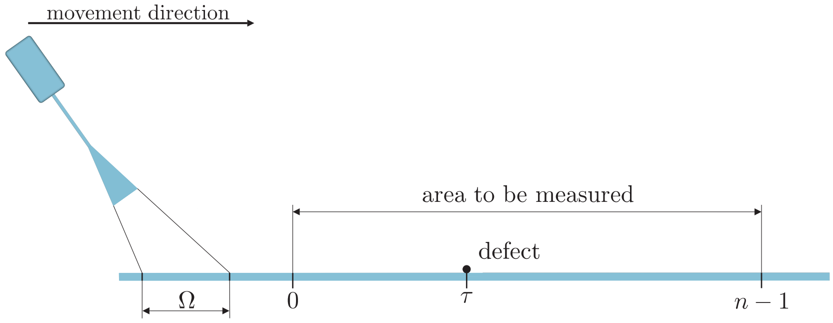


Figure 5.5: Measurement of the impulse response

that a sample of the input signal associated with position of defect has a value "1". If there is no defect, then the sample has a value "0".

In practice, the FIR system output (i.e. Doppler signal) is known, since it is measured. In the we consider how to determine the FIR system impulse response.

5.3.1 Impulse Response Evaluation

An impulse response describes such physical aspects of microwaves as propagation, reflection, material properties, influence of defects on measured signal etc. It also contains information about measurement conditions such as speed, slope angle, antenna radiation pattern etc. The high information capacity of the impulse response makes its modeling extremely difficult. We decide to derive the impulse response from a so-called reference measurement. This approach is very common in NDT. It includes collection of information about some physical quantity from artificially-made defects. Then, this information is used to classify and characterize real specimens.

Let us consider an approach to derive the FIR impulse response $\mathbf{h} \in \mathbb{R}^n$. We perform a line-scan of the specimen with one artificial defect[1], where $\mathbf{y} \in \mathbb{R}^n$ is the measured signal. Placing the defect at the position which corresponds to the measured sample y_0 is equivalent to feeding the unit sample $\mathbf{x} = \boldsymbol{\delta} \in \mathbb{R}^n$ to the FIR system, for reference see equation (5.4). These force the impulse response to appear at the output of the FIR system i.e. $\mathbf{y} = \mathbf{h}$. However, in practice it is difficult to ensure to coincide the measured sample y_0 and the position of the defect. This is because the size Ω of an area irradiated by the radar is only approximately known, see Figure 5.5. In order to overcome the above mentioned problem we suggest the following. We place the defect somewhere inside the area to be measured such that the defect position τ is in the interval $(0 : n-1)$. It is also necessary to ensure that the defect is not seen by the radar at 0-th and $(n-1)$-th measured samples. The location of the defect at τ implies shift of the unit sample. Hence, the new input of the FIR system $\mathbf{x}' \in \mathbb{R}^n$ is given as:

$$\mathrm{x}'_i = \begin{cases} \mathrm{x}_{\tau+i} & \text{if} \quad 0 \leq i \leq (n-1-\tau) \\ \mathrm{x}_{i-(n-\tau)} & \text{if} \quad (n-\tau) \leq i \leq (n-1) \end{cases} \tag{5.5}$$

[1] the defect is a point scatterer, for reference see Section 5.1

By using equation (5.5) and the structure of the FIR system (see Figure 5.4) the impulse response can be computed from the measured signal as:

$$\mathrm{h}_i = \begin{cases} \mathrm{y}_{\tau+i} & \text{if} \quad 0 \leq i \leq (n-1-\tau) \\ \mathrm{y}_{i-(n-\tau)} & \text{if} \quad (n-\tau) \leq i \leq (n-1) \end{cases} \tag{5.6}$$

From the considerations given above we conclude that in order to define the impulse response of the Doppler measurement system we need to make a reference measurement. From that measurement (signal $\mathbf{y}$) we compute the impulse response $\mathbf{h}$ according to equation (5.6). This impulse response is kept in data bank and is used by deconvolution approach (also called convolution inverse) for defects evaluation.

5.3.2 Deconvolution (Convolution Inverse)

Let us denote $\mathbf{x} \in \mathbb{R}^n$ be the input signal of the FIR system. The input signal characterizes defect. Some entry of the input signa is "1" if the defect is located at the position corresponding to that entry. The entry is 0 otherwise. In the following we call the input signal $\mathbf{x}$ as characteristic signal. Let the output of the FIR system is $\mathbf{y} \in \mathbb{R}^n$. The impulse response which is already pre-saved is $\mathbf{h} \in \mathbb{R}^n$. By substitution of these signals into equation (5.3) we derive for all $\tau \in [0 : n-1]$

$$\mathrm{y}_\tau = (\mathbf{h} * \mathbf{x})_\tau. \tag{5.7}$$

We use the *convolution theorem* for positive wrapped convolution (for reference see [80, page 72]), to represent equation (5.7) in a more convenient form. This theorem says that the Fourier transform of the convolution of two signals is equal to componentwise multiplication of their Fourier transformations:

$$\hat{\mathcal{F}}_{\mathbf{y}}(f) = \hat{\mathcal{F}}_{\mathbf{x}}(f) \cdot \hat{\mathcal{F}}_{\mathbf{h}}(f), \tag{5.8}$$

for all $f \in [0 : n-1]$.

From (5.8) we evaluate the characteristic signal $\mathbf{x}$ for all $t \in [0 : n-1]$ as

$$\mathrm{x}'_t = \hat{\mathcal{F}}^{-1}\left(\frac{\hat{\mathcal{F}}_{\mathbf{y}}}{\hat{\mathcal{F}}_{\mathbf{h}}}\right)(t). \tag{5.9}$$

Equation (5.9) represents the deconvolution procedure which solves the goal of finding the FIR system input i.e. characteristic signal.

In practical applications equation (5.9) may fail because of the noise in a measured signal even at high SNR. We will study this problem in the following section.

5.3.3 Problems of Deconvolution Approach

It is well known that in practice the measured signal is always corrupted by noise. It happens because of many reasons. These are internal noise of the radar, instability (shaking) of the Doppler measurement system, environment temperature, any external microwave source, roughness of the measured specimen etc. In signal

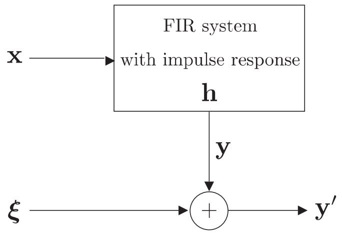


Figure 5.6: FIR system with additive noise

processing it is convenient to represent noise in form of the signal which is added to the noise-free signal [82, 83]. In Figure 5.6 we show a flowchart which introduces a FIR system with noise. Here, a noise corrupted signal $\mathbf{y}'$ is the result of summation of the ideal FIR output $\mathbf{y}$ and noise $\boldsymbol{\xi}$. In Section 5.4.2 we will discuss the nature of the noise in details.

Let us examine the influence of noise on the deconvolution approach. We introduce $\mathbf{x} \in \mathbb{R}^n$ to be the FIR system input which is defined as the unit-sample shifted by $\lfloor n/2 \rfloor + 1$, see Figure 5.7(a). This models the presence of the defect placed in the middle of the imaginary line of scan. A modeled noise-free FIR system output $\mathbf{y} \in \mathbb{R}^n$ is represented in Figure 5.7(b). In this experiment $\mathbf{y}$ is an arbitrary noise-free Doppler signal since the only purpose of the experiment is to test the noise-stability of the deconvolution approach. A noise-free impulse response $\mathbf{h} \in \mathbb{R}^n$ is computed from $\mathbf{y}$ according to (5.6).

We perform deconvolution by using equation (5.9) assuming that $\mathbf{x}$ is unknown and $\mathbf{y}$ and $\mathbf{h}$ are given. The result of this is represented in Figure 5.7(c). By comparing signal from Figure 5.7(c) with the original FIR system input (given in Figure 5.7(a)) we conclude that the deconvolution procedure retrieves FIR system input successfully. This certainly let us expect a stability of the deconvolution procedure (5.9) at noise-free environment.

We worsen the situation by adding noise $\boldsymbol{\xi}$, with that SNR$=$ 35db, to the signal $\mathbf{y}$. After this the deconvolution procedure is computed again, see Figure 5.7(d). The experiment shows that even if noise is very small the deconvolution approach fails.

We also examine the deconvolution approach with real data, namely applying it to the Doppler signals represented in Figures 5.1 and 5.2. In the situation of two defects placed at distance $L = 1.5\lambda$ it is extremely difficult to differentiate true defects and spurious peaks, see Figure 5.7(e). Here real locations of defects are labeled by dots. The situation is similar in the case of two defects placed at $L = 2.5\lambda$ and $L = 6.5\lambda$, see Figure 5.7(f) and (g), respectively. Here we can recognize defects (also labeled by dots), but it is still not possible to differentiate real and spurious defects. In case of four defects we are only able to recognize the first defect while the last three defects are lost, see Figure 5.7(h).

The problem of defect detection rises because of the division in equation (5.9). It magnifies effect of noise so that the inverse Fourier transform heavily suffers. Mathematically this problem can also be written as follows. Let $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ are ideal and actual solutions. Then

$$(\mathbf{x} * \mathbf{h})_\tau - (\mathbf{x}' * \mathbf{h})_\tau \approx 0$$

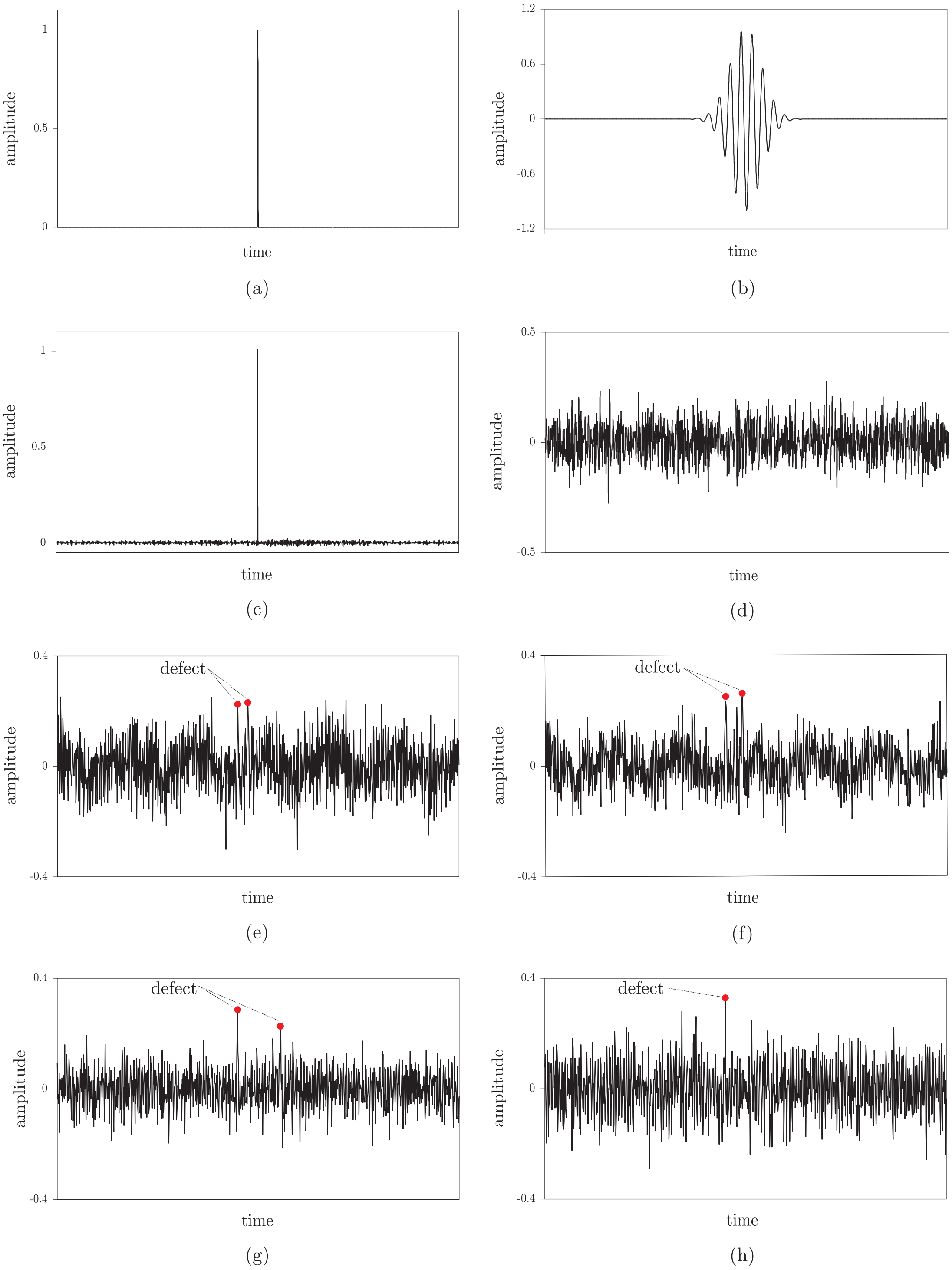


Figure 5.7: Deconvolution approach: (a) ideal FIR system input; (b) ideal FIR system output; (c) deconvolution inverse (no noise); (d) deconvolution inverse (with noise); (e) defect signal, two defects, $L = 1.5\lambda$; (f) defect signal, two defects, $L = 2.5\lambda$; (g) defects signal, two defects, $L = 6.5\lambda$; (h) defects signal, four equally spaced defects, $L = 2.5\lambda$

does not imply that $\mathbf{x} - \mathbf{x}' \approx 0$. Thus, in general, $\mathbf{x}'$ may be far from the correct value even if the noise is small.

In order to overcome the above problem the *Maximum Entropy Deconvolution* (MED) algorithm is applied. We discuss this algorithm in the following section.

5.4 Maximum Entropy Deconvolution (MED)

Maximum entropy (ME) processing includes extremely robust algorithms to analyze real observed data which are corrupted by noise. ME processing based on certain suppositions about the nature of noise and measured data in terms of probability theory. Operations which are defined in probability theory allow to extract noise-free signal from the noised one under suppositions which were made.

ME signal processing was originally developed in the context of optical image processing. It is applied, in particular, to reconstruct sharp objects from noisy images which are blurred by out-of-focus or motion effects. Examples of successive image restoration with ME can be found in [84]. Nowadays ME has become very popular in processing of medicine images. Thus, in [85] a restoration technique of nuclear images is developed. It was shown that ME processing provides superior results in comparison to the standard image processing methods. In [86] it is reported about efficient multi-thresholding of slice images. Here ME approach is referred to as the most important threshold selection method. Recently it was discovered that ME is very efficient in speech recognition applications [87,88]. This technique allows to differentiate asynchronous and overlapping speech features and their combination.

5.4.1 Bayes' Estimation

In Section 5.3.3 we have discussed disadvantages of the deconvolution approach. A problem we examined was to compute the input of the FIR system by using its noisy measured output and the impulse response. In order to overcome this problem the MED algorithm, for Doppler signal processing, was developed.

Let $(\Omega, \mathcal{F}, P)$ be a probability space on the domain Ω, where $(\Omega, \mathcal{F})$ is a measurable space, $\mathcal{F}$ are the measurable subsets of Ω, and P is a measure on Ω with $P(\Omega) = 1$, for reference see [89, p. 38-39, 78-81, and 112-114], [90]. We define the domain Ω as a set of vector pairs

$$\Omega = \{\ (\mathbf{x}, \mathbf{y}') \mid \mathbf{x} \in [0 : M-1]^n,\ \mathbf{y}' \in \mathbb{R}^n\ \} = \mathrm{I} \times \widetilde{\mathrm{M}},$$

where $\mathbf{x}$ is an ideal FIR input, $\mathbf{y}'$ is a real measured signal with noise and a constant M is, intuitively, a sum of height of all defects.

Let us define the special events $\mathrm{H}_{\mathbf{x}},\ \mathrm{E}_{\mathbf{y}'} \in \mathcal{F}$. Let $\mathrm{H}_{\mathbf{x}} = \{\,\mathbf{x}\,\} \times \widetilde{\mathrm{M}}$ be the set of all the experiments for some ideal input $\mathbf{x}$. Later we refer $\mathrm{H}_{\mathbf{x}}$ to as hypothesis. Let $\mathrm{E}_{\mathbf{y}'} = \mathrm{I} \times \{\,\mathbf{y}'\,\}$ is the event that the measured noisy signal is $\mathbf{y}'$ (the noise-free ideal signal is $\mathbf{y}$), for reference see Section 5.3.3.

The task to solve is to decide which hypothesis $\mathrm{H}_{\mathbf{x}}$ for $\mathbf{x} \in \mathrm{I}$ is more probable if $\mathrm{E}_{\mathbf{y}'}$ is given. The solution of the task can be found by using of the Bayes'

Theorem [89]. Let

$$\Omega = \bigcup_{\mathbf{x} \in \mathbf{I}} \mathrm{H}_{\mathbf{x}}, \tag{5.10}$$

where $\mathbf{x}_i$ means some i-th ideal input signal. Then, for every $i \in [0 : n-1]$ holds

$$P(\mathrm{H}_{\mathbf{x}} \,|\, \mathrm{E}_{\mathbf{y}'}) = \frac{P(\mathrm{H}_{\mathbf{x}}) \cdot P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})}{P(\mathrm{E}_{\mathbf{y}'})} \tag{5.11}$$

The latter equation expresses the Bayers' theorem for computation of co-called posterior probability $P(\mathrm{H}_{\mathbf{x}} \,|\, \mathrm{E}_{\mathbf{y}'})$, i.e. the probability of the event $\mathrm{H}_{\mathbf{x}}$ under condition $\mathrm{E}_{\mathbf{y}'}$ is measured.

The idea of the Bayes' estimator in (5.11) is to choose a $\mathrm{H}_{\mathbf{x}_i}$, which maximizes $P(\mathrm{H}_{\mathbf{x}_i} \,|\, \mathrm{E}_{\mathbf{y}'})$. In equaiton (5.11) the denominator can be omitted sinse it is constant for any i. Hence, the maximization task is written:

$$\max_{\mathrm{H}_{\mathbf{x}} \in \Omega} \quad P(\mathrm{H}_{\mathbf{x}}) \cdot P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}}) \tag{5.12}$$

Later we discuss the computation of probabilities $P(\mathrm{H}_{\mathbf{x}})$ and $P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})$.

5.4.2 Computation of the probability $P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})$

The goal of this section is to express $P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})$ in terms of the measured Doppler signal. Let us denote the ideal measured signal (without noise) and the real measured signal (with noise) as $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^n$, respectively. The ideal FIR system output $\mathbf{y}$ is the positive wrapped convolution for $\tau \in [0 : n-1]$

$$\mathrm{y}_\tau = (\mathbf{x} * \mathbf{h})_\tau, \tag{5.13}$$

where $\mathbf{x}, \mathbf{h} \in \mathbb{R}^n$ are the ideal input and the impulse response of the FIR system, for reference see equation (5.7).

Let us define the set of real number $\mathbb{R}_\varepsilon$ as

$$\mathbb{R}_\varepsilon = \{\, \varepsilon \cdot z \,|\, z \in \mathbb{Z}\},$$

where ε is called accuracy. By using the considerations in Section 5.3.2 concerning the FIR system with noise $\mathbf{n}$, we define

$$\mathbf{y}' = \mathbf{round}\,(\mathbf{y} + \mathbf{n})\,, \tag{5.14}$$

where the function **round** rounds values of $\mathbf{y}'$ to the nearest in $\mathbb{R}$.

We assume entries n_i for $i \in [0 : n-1]$ to be mutually independent normally distributed random variable with standard deviation ε and mean 0. Then, the probability $P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})$ in equation (5.12) can be defined as:

$$\begin{aligned} P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}}) &= P\left(\mathbf{y}' = \mathbf{round}(\mathbf{y}+\mathbf{n})\right) \\ &= P\left(\bigcap_{i=1}^{n} \mathrm{y}_i = \mathbf{round}(\mathrm{x}_i + \mathrm{n}_i)\right) \\ &= P\left(\bigcap_{i=1}^{n} \mathrm{n}_i \in \mathrm{y}'_i - \mathrm{y}_i + \left[-\frac{\varepsilon}{2} : \frac{\varepsilon}{2}\right]\right) \\ &= \prod_{i=0}^{n-1} P\left(\mathrm{n}_i \in \mathrm{y}'_i - \mathrm{y}_i + \left[-\frac{\varepsilon}{2} : \frac{\varepsilon}{2}\right]\right) \\ &\approx \prod_{i=0}^{n-1} \frac{\varepsilon}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\mathrm{y}'_i - \mathrm{y}_i)}{2\sigma^2}\right) \end{aligned} \tag{5.15}$$

Since $\mathbf{y}$ is computed from FIR input $\mathbf{x}$ (see equation 5.13), then the probability $P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_{\mathbf{x}})$ shows how close $\mathbf{x}$ is to the input signal of the FIR system, which minimize the difference between noisy real signal $\mathbf{y}'$ and ideal signal $\mathbf{y}$. Thus, a small value of probability means that $\mathbf{x}$ is "unlikely" to be defined correctly i.e. $\mathbf{y}'$ and $\mathbf{y}$ differ strongly.

5.4.3 Computations of probability $P(\mathrm{H}_{\mathbf{x}})$

In the current section we give the definition of the probability $P(\mathrm{H}_{\mathbf{x}})$, where H is the event that the FIR system receives some ideal input signal $\mathbf{x}$.

In order to compute the probability that a particular $\mathbf{x}$ appears at the input of FIR system we make the following abstraction. We know that we perform measurements by moving the Doppler radar along the line of scan. We imagine that the line of scan is separated into small equal pieces. We call these bins and assume that bins are numbered. If a defect lies on the line of scan in the particular position, then we say that the bin associated with this position is not empty. We also assume the bins are filled by grains of sand. We number grains $1, 2, \ldots, M$ such that the total number of grains is M.

We describe the outcome of throwing M grains by a function

$$b : [0 : M-1] \to [0 : n-1]$$

where $b(j)$ is the bin hit by the j-th grain we throw. For each bin i let P_i to be the probability that bin i is hit. We assume always one bin is hit:

$$\sum_{i=0}^{n-1} P_i = 1$$

For function b we define vector

$$\mathbf{x} = (\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_{n-1})$$

by

$$\mathrm{x}_i = \#\,[\,j \,|\, b(j) \;=\; i],$$

where x_i is the number of grains thrown into bin i by function b. Thus, we have

$$\sum_{i=0}^{n-1} \mathrm{x}_i = M. \tag{5.16}$$

If we throw M grains independently into bins with probabilities P_i then the probability to observe the particular outcome b is

$$Pr(b) = P_0^{\mathrm{x}_0} \cdot P_1^{\mathrm{x}_1} \cdot \ldots \cdot P_{n-1}^{\mathrm{x}_{n-1}}. \tag{5.17}$$

For each vector $\mathbf{x}$ satisfying (5.16) there are

$$\frac{M!}{\mathrm{x}_0!, \mathrm{x}_1!, \ldots, \mathrm{x}_{n-1}!}$$

functions b providing vector $\mathbf{x}$, see [91, p. 44]. Hence the probability to observe vector $\mathbf{x}$ is

$$Pr(\mathbf{x}) = \frac{M!}{\mathrm{x}_0!, \mathrm{x}_1!, \ldots, \mathrm{x}_{n-1}!} \cdot P_0^{\mathrm{x}_0} \cdot P_1^{\mathrm{x}_1} \cdot \ldots \cdot P_{n-1}^{\mathrm{x}_{n-1}} \tag{5.18}$$

In application we later assume that all P_i equal to $1/(m'n')$, where m', n' are dimensions of Doppler image.

5.4.4 Computation of probability $P(\mathrm{H}_\mathbf{x} \,|\, \mathrm{E}'_\mathbf{y})$

It is possible to rewrite the maximization task (5.12) by taking the natural logarithm from its both parts without changing the problem:

$$\max_{\mathrm{H}_\mathbf{x} \in \Omega} \quad \ln[\, P(\mathrm{H}_\mathbf{x} \,|\, \mathrm{E}_{\mathbf{y}'})\,] \propto \ln[\, P(\mathrm{H}_\mathbf{x}) P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_\mathbf{x})\,]. \tag{5.19}$$

According to the mathematical transformations proposed in [92] we rewrite equations (5.15) and (5.18) in the form which is more suitable for practical implementation:

$$\ln[\, P(\mathrm{E}_{\mathbf{y}'} \,|\, \mathrm{H}_\mathbf{x})\,] \;\propto\; \alpha\, \chi(\mathbf{x}, \mathbf{y}', \mathbf{h}, \sigma) \tag{5.20}$$

and

$$\ln[\, P(\mathrm{H}_\mathbf{x})\,] \;\propto\; S(\mathbf{x}, \mathbf{b}), \tag{5.21}$$

where complete definitions of functions χ, S and parameter α we be discuss in the following.

Let Φ be the set of all possible inputs of the FIR system. Since each hypothesis $\mathrm{H}_\mathbf{x} \in \Omega$ is associated with some unique system input $\mathbf{x} \in \Phi$, where $\Phi \subset \mathbb{R}^n$, we can change the argument of the maximization task to $\mathbf{x}$. Thus, by using equations (5.20) and (5.21) we rewrite (5.19) in terms of χ and S functions as follows:

$$\max_{\mathbf{x} \in \Phi} \quad \Psi(\mathbf{x}) = -\left(S(\mathbf{x}, \mathbf{b}) + \alpha \cdot \chi(\mathbf{x}, \mathbf{y}', \mathbf{h}, \sigma) \right), \tag{5.22}$$

where $\alpha \in \mathbb{R}$, $\alpha \geq 0$ is the so-called regularization parameter, which we discuss below. In the literature the function Ψ is referred to as the *entropy function*.

Let us consider all the components of equation (5.22) in detail. The function S can be derived from equation (5.18). By using the Stirling's formula [93, pages 50-53]:

$$\ln n! \approx n \cdot \ln n - n + \frac{1}{2}\ln(2\pi n)$$

and letting $P(w_i) = b_i$, we derive

$$S(\mathbf{x}, \mathbf{b}) = \sum_{i=0}^{n-1} \mathrm{x}_i \left(\ln\left(\frac{\mathrm{x}_i}{\mathrm{b}_i}\right) - 1 \right). \tag{5.23}$$

In the latter equation the signal $\mathbf{b} \in \mathbb{R}^n$ is positive for all $i \in [0 : n-1]$, i.e. $\mathrm{b}_i > 0$. In practice all b_i-th are assigned by some small values (will be discussed in detail in Section 5.6.4).

We notice that S returns a real-valued number if for all $i \in [0 : n-1]$ the value x_i is greater then 0.

The next function χ is defined as:

$$\chi(\mathbf{x}, \mathbf{h}, \mathbf{y}', \sigma) = \sum_{i=0}^{n-1} \frac{(\mathrm{y}'_i - (\mathbf{x} * \mathbf{h})_i)^2}{2\sigma^2}. \tag{5.24}$$

We notice that in practice the value of the standard deviation σ is usually given.

The maximization task (5.22) has two parts, namely functions S and χ. If we only maximize χ, we will find $\mathbf{x}$ such that the noise-free signal[2] $\mathbf{y}$ and the measured signal $\mathbf{y}'$ will be equal. In this case $\mathbf{x}$ will entirely approximate noise which is present the in measured $\mathbf{y}'$. It may cause spurious oscillations (peaks) in $\mathbf{x}$ so that it will be impossible to detect positions of defects. From other side, if we only maximize S, the measured signal $\mathbf{y}'$ will not depend on characteristic signal $\mathbf{x}$, see equation (5.23). To set the desired trade-off between χ and S we use the regularization parameter α. If α is large, then too much weight will be given to the measured data (function χ), i.e. the effect of noise will be magnified. If α is too small, then too much weight will be given to the entropy function S so that all the instances of $\mathbf{x}$ will fall down to the default value $\mathbf{b}$. Generally, α can be found iteratively as it discussed in [92]. This, of course, requires additional computational resources. In practice α is estimated once from the reference measurements so that it corresponds to the particular measurement conditions. If the conditions change, then α must be estimated again.

To find a solution of the problem (5.22) is a relatively complex task. It requires maximization of the function in n-dimensional space. This problem will be studied in the following sections.

5.5 Maximization of Posterior Probability

As we have already discussed, maximization of posterior probability $P(E \,|\, H)$ is done by looking for a vector $\mathbf{x} \in \mathbb{R}^n$ (FIR input) which maximizes function Ψ, see

[2] we remind that the ideal measured noise-free signal $\mathbf{y}$ is defined for all $i \in [0 : n-1]$ as $\mathrm{y}_i = (\mathbf{x} * \mathbf{h})_\tau$

equations (5.19) and (5.22). The problem (5.22) is called optimization problem and belongs to the *optimization theory.* In the next section we present some background of optimization theory needed in the following reading.

5.5.1 Optimization Problem, Basic Definitions and Notations

Optimization theory deals with either constrained or unconstrained problems.

The *unconstrained* optimization problem [94, page 3] is to maximize a real-valued function f (called *objective function*) of n variables. To maximize means to find a *local maximizer* $\mathbf{x}^* \in \mathbb{R}^n$ such that

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \text{ near } \mathbf{x}^*. \tag{5.25}$$

It is standard to denote this problem as

$$\max_{\mathbf{x}} f(\mathbf{x}). \tag{5.26}$$

In general, a maximization problem (5.26) can be referred to as a minimization problem by substituting $-f$ instead of f. We call the *optimizer* the minimizer of minimization problem or the maximizer of maximization problem.

The global maximization problem is defined as

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) \quad \text{for all } \mathbf{x}, \tag{5.27}$$

where $\mathbf{x}^*$ is a *global optimizer.* In practice seeking for a global maximum $\mathbf{x}^*$ is a difficult task. Usually this type of optimization problems is applied to functions which have properties of *convexity* and *smoothness.*

The *constrained* maximization problem is to maximize a function f over set $\mathbb{U} \subset \mathbb{R}^n$, i.e. to find a local maximizer $\mathbf{x}^* \in \mathbb{U}$ such that

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{U} \text{ near } \mathbf{x}^*. \tag{5.28}$$

Similar to (5.26) we express this as

$$\max_{\mathbf{x} \in \mathbb{U}} f(\mathbf{x}). \tag{5.29}$$

In practice constrained problems are usually transformed into easier subproblems that can then be solved in terms of unconstrained optimization. A standard solution is to design a so-called penalty function which is used to define set $\mathbb{U}$ (i.e. to set desired constrains), [94, 95].

Definition 5.5.1 *For $\mathbf{x} \in \mathbb{R}^n$ we define the **gradient** $\nabla \mathbf{f} \in \mathbb{R}^n$ of function $f : \mathbb{R}^n \to \mathbb{R}$ as*

$$\nabla \mathbf{f} = \left(\frac{\partial f}{\partial \mathrm{x}_1}, \frac{\partial f}{\partial \mathrm{x}_2}, \ldots, \frac{\partial f}{\partial \mathrm{x}_{n-1}} \right), \tag{5.30}$$

*where i-th component of vector $\nabla \mathbf{f}$ such that $i \in [0 : n-1]$ is called **partial derivative** of f with respect to x_i.*

Definition 5.5.2 *We define the **Hessian matrix** (or second derivative) $\nabla^2\mathbf{f} \in \mathbb{R}^{n\times n}$ of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ as:*

$$\nabla^2 \mathrm{f}_{ij} = \frac{\partial^2 f}{\partial \mathrm{x}_i \partial \mathrm{x}_j}. \tag{5.31}$$

If the first and second derivatives exist and are continuous then it is said that the function f is twice continuously differentiable.

Definition 5.5.3 *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a matrix. We define the **conditional number** of $\mathbf{A}$ as*

$$k(\mathbf{A}) = \frac{\lambda_0}{\lambda_{n-1}}$$

where λ_1 and λ_{n-1} are the largest and lowest eigenvalues of the matrix $\mathbf{A}$, respectively.

5.5.2 Methods for Unconstrained Optimization

In the last thirty years a powerful collection of algorithms for unconstrained optimization has been developed. A broad description of these algorithms and their properties can be found in more detail in any optimization book, e.g. see for reference [94–101]. The choice of an algorithm to solve a specific task depends on several problem characteristics. These include *problem size* (i.e. the number of variables), *speed of convergence*, *computational demands*, *objective function properties* etc.

Methods for unconstrained optimization can be divided into classes according to their operating principles:

1. Basic Methods
2. Newtonian Methods
3. Quasi-Newtonian Methods

Steepest descent methods (which belong to the first group) are probably most important methods for unconstrained optimization. These algorithms produce a minimizer $\mathbf{x}^k$ iteratively as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \triangle \mathbf{x}^k, \tag{5.32}$$

where $\triangle\mathbf{x} \in \mathbb{R}^n$ is a vector called *search direction* and $k = 0, 1, \dots$ stands for the iteration number.

The search direction $\triangle\mathbf{x}$ in a descent method have to satisfy the following condition:

$$\nabla \mathbf{f}(\mathbf{x}^k)^T \triangle \mathbf{x}^k < 0, \tag{5.33}$$

i.e. it have to make an acute angle with the gradient and it is called *descent direction*. Condition (5.33) guarantees that function f can be reduced along direction $\triangle\mathbf{x}^k$. It is standard to take the descent direction equal to the negative gradient of the objective function:

$$\triangle \mathbf{x}^k = -\nabla \mathbf{f}^k.$$

In equation (5.32) a constant λ^k is called the *step length.* Generally, λ^k is chosen to ensure that the value of function f decreases at every iteration step. The procedure which allows to find such a λ^k is called *exact line search* and is defined as

$$\min_{\lambda^k > 0} f\left(\mathbf{x}^k + \lambda^k \triangle \mathbf{x}^k\right) \tag{5.34}$$

In practice it is very expensive to compute the exact solution of the line search problem given in (5.34). It may require many evaluations of objective function f. A more practical strategy is to evaluate λ^k by using *inexact line search.* For this a *backtracking line search* algorithm is proposed. In unconstrained optimization it is considered to be very simple and quite efficient. The main idea of the backtracking line search algorithm is to try out a candidate value of λ^k which satisfies a certain condition formulated below.

Definition 5.5.4 *Let f be an objective function. Let $\mathbf{x}^k$, $\triangle\mathbf{x}^k$, and λ^k be its minimizer, search direction, and step length at the k-th iteration, respectively. We define the* ***backtracking line search*** *function*

$$\text{btlSearch} : \mathbb{R} \to \mathbb{R},$$

by

$$\text{btlSearch}\left(\lambda^k\right) = \\ = \begin{cases} \lambda^k & \text{if} \quad f\left(\mathbf{x}^k + \lambda^k \triangle \mathbf{x}^k\right) \leq f\left(\mathbf{x}^k\right) + c_1 \lambda^k \nabla \mathbf{f}\left(\mathbf{x}^k\right)^T \triangle \mathbf{x}^k \\ \text{btlSearch}\left(c_2\, \lambda^k\right) & \text{otherwise} \end{cases},$$

where c_1, $c_2 \in \mathbb{R}$ are constants such that $0 < c_1 < 0.5$, $0 < c_2 < 1$.

Definition 5.5.4 is based on so-called **Wolfs conditions** [97, page 36]. These guarantee a length λ^k such that the objective function will be sufficiently decreased in descent direction $\triangle\mathbf{x}^k$. Function **btlSearch** starts from $\lambda^k = 1$ (Newton step) with following decreasing. In the literature it was shown that **btlSearch** eventually terminates [97, page 41].

Advantage of steepest descent methods is their computational simplicity. Usually it takes $\mathcal{O}(n)$ arithmetical operations to evaluate the objective function $f(\mathbf{x}^k)$ and the gradient $\nabla\mathbf{f}(\mathbf{x}^k)$. In practice solving the optimization problem with descent methods may be not efficient. Main disadvantages of descent methods are:

- Descent methods often exhibit linear convergence
- The choice of backtracking line search parameters c_1 and c_2 has a noticeable effect on convergence. It was shown that the exact line search may improve the convergence of steepest methods.
- The convergence rate critically depends on the conditional number of the Hessian of the objective function, see Definition 5.5.3.

This makes steepest descent methods often unpractical even if the problem is moderately small.

The algorithms for unconstrained optimization which belong to the second group are entirely based on the Newton algorithm. This algorithm operates according to the iterative scheme given in equation (5.32). Its descent direction is determined from the gradient and Hessian of the objective function f as

$$\triangle \mathbf{x}^k = -\left(\nabla^2 \mathbf{f}(\mathbf{x}^k)\right)^{-1} \nabla \mathbf{f}(\mathbf{x}^k). \tag{5.35}$$

The Newton algorithm also uses the backtracking line search algorithm (see Definition 5.5.4) in order to optimize constant λ^k.

In the following we summarize the most important advantages of the Newton algorithm:

- Convergence of the Newton algorithm is very fast. In particular, it has a quadratic character if a current minimizer $\mathbf{x}^k$ is close to the local minimizer $\mathbf{x}^*$. As soon as the algorithm reaches a quadratic convergence it only needs a few iterations to produce a solution of very high accuracy.

- The Newton algorithm scales very well with problem size, i.e. these methods are independent from the conditional number. Convergence speed of these methods on problem of size n is similar to its performance on problem of size m even if $n \gg m$.

- The Newton algorithm reaches global convergence if the objective function is convex (concave)[3]. In [92] the concavity of the entropy function Ψ was not presented. In Appendix A.0.4 we introduce the entropy function concavity proof.

Even if the Newton algorithm has high convergence speed and scalability it also has several disadvantages:

- Computation and saving of the Hessian of the objective function at every iteration step may strongly slow down the algorithm.

- Computations of Newton step requires the matrix inverse which is computationally expensive. Its complexity is $\mathcal{O}(n^3)$.

The third group of methods of unconstrained optimization which are referred to as *quasi-Newton methods* require less computational efforts to form the search direction. These methods utilize different computational schemes for the approximation of Hessian $\nabla^2 \mathbf{f}(\mathbf{x}^{k+1})^{-1}$ from the k-th iteration. One of the most successful algorithms is the BFGS one, named after its discovers Broyden, Fletcher, Goldfarb, and Shanno. According to [96, page 67] the inverse of the Hessian can be approximated by:

$$\nabla^2 \mathbf{f}(\mathbf{x}^{k+1})^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \nabla^2 \mathbf{f}(\mathbf{x}^k)^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \tag{5.36}$$

[3] depends whether the minimization or maximization of the objective function is performed

In (5.36) vectors $\mathbf{s}_k, \mathbf{y}_k \in \mathbb{R}^n$ are defined as follows:

$$\mathbf{s}_k = \mathbf{x}^{k+1} - \mathbf{x}^k$$
$$\mathbf{y}_k = \nabla \mathbf{f}(\mathbf{x}^{k+1}) - \nabla \mathbf{f}(\mathbf{x}^k).$$

Only at the first iteration of the BFGS algorithm the computation of $\nabla^2 \mathbf{f}(\mathbf{x}^0)^{-1}$ is required. After that the Hessian is computed by the iterative scheme (5.36).

Let us summarize properties of the BFGS algorithm. Its advantages are:

- the BFGS algorithm requires computation of the Hessian only at the very first iteration. In practice it can be computed before the main iteration cycle.

- the BFGS algorithm is computationally efficient. It requires $\mathcal{O}(n^2)$ arithmetical operations per iteration.

The following statements are referred to as disadvantages of BFGS algorithm:

- Unknown initial approximation of $\nabla^2 \mathbf{f}(\mathbf{x}^0)^{-1}$. Since the computation of the initial Hessian can be very expensive it is standard to use an identity matrix instead. This may slow down the convergence of the algorithm.

- In practice the equation (5.36) may produce a very poor approximation of the Hessian after several iterations. This problem has been studied analytically and experimentally in the literature. It has been shown that the BFGS algorithm corrects itself only if an adequate line search (see Definition 5.5.4) is performed. This also may reduce the convergence speed of BFGS.

- Being applied to a practical problem the rate of convergence of the BFGS algorithm is lower than the speed of convergence of the Newton algorithm. It is superlinear rather than quadratical.

In the following sections we will check, which method from unconstrained optimization is the most suitable for MED algorithm applied on Doppler signal.

5.5.3 Comparison of methods of unconstrained optimization

In this section we compare the efficiency of algorithms of unconstrained optimization (being applied in the context of the MED algorithm) which we discussed in Section 5.5.2. These are steepest descent, BFGS, and Newton algorithms. We perform the comparison of algorithms in two steps. At the first step we check which of the algorithms is able to compute the correct solution. Here we also discuss the influence of backtracking line search on the computed optimizer. At the second step we check which of the algorithms has the highest convergence speed.

Let us consider the first comparison step. We perform a number of measurements of specimens with artificial defects. The locations of defects are known. Then, the measured data are processed with the MED algorithm for different unconstrained optimization solvers (namely, steepest descent, BFGS and Newton algorithms). The position of defects are estimated from the processed data.

We compare the quality of the estimation among the algorithms. We also verify whether estimated location of defects coincide with real ones.

In all the experiments (for steepest descent, BFGS and Newton algorithms) estimated positions of defects coincide with real ones. However, the computed characteristics signals are corrupted by a number of spurious peaks which can heavily degrade the resolution. In order to estimate which of algorithms introduces the lesser number of spurious peaks we suggest the following. We assume that in a very worse case each entry of the characteristic signal has a peak which is interpreted as a defect. We count all the spurious peaks in the computed signal and introduce it as a percentage ratio of the very worst case. If the percentage ratio is small, then we conclude that the solution is close to the optimal i.e., real defects can be found easily. At higher percentage ratio it is difficult to differentiate between real and spurious peaks, hence we refer to the solution as non-optimal.

In Figure 5.8(a) we compare steepest descent, BFGS and Newton algorithms without backtracking line search procedure. Both steepest descent and BFGS introduce inferior performance in comparison with Newton algorithm. Neither steepest descent nor BFGS do not reach the optimal solution what leads to about 50% of spurious peaks in the characteristic signal. In a contrary, the Newton algorithm introduces only about 5% of spurious peaks what makes estimation of location of the real defects more correct.

In Figure 5.8(b) we compare steepest descent, BFGS and Newton algorithms with backtracking line search procedure. It improves the performance of the steepest descent algorithm. As we expected the line search ensures the better optimizer what reduces the number of spurious peaks.

In Figure 5.8(b) we can see that the BGFS fails in seeking for the optimal solution even if the line search is applied. A poor performance of the BFGS algorithm is caused by the update of Hessian inverse of the entropy function in equation (5.36). Such an update leads to very low or even zero convergence of the algorithm.

Analyzing Figures 5.8(a), (b) we conclude that the Newton algorithm allows to extract the best optimizer among the others algorithms. We also conclude that the performance of the Newton algorithm does not depend on the line search procedure.

Data represented in 5.8(a), (b) were computed with the impulse response (we refer to as proper impulse response) measured for the same type of artificial defect which was used in the measurement. This, certainly, makes search of solution of MED algorithm easier. Often in practice, the form of the defects of the analyzed specimen differs from the form of the defect which is used to measure the impulse response (we refer to as improper impulse response). In the following we examine behavior of the Newton algorithm with proper and improper impulse responses, see Figures 5.8(c) and (d).

In Figure 5.8(c) the optimizer computed by Newton algorithm with the proper impulse response is presented. Here we can clearly see the edges of the specimen and a number of defects. The optimizer computed by the Newton algorithm with improper impulse response is presented in Figure 5.8(d). Comparing Figures 5.8(c) and (d) we conclude that improper impulse response worsens the optimizer making impossible separation of closely spaced defects. However, even if the optimizer in

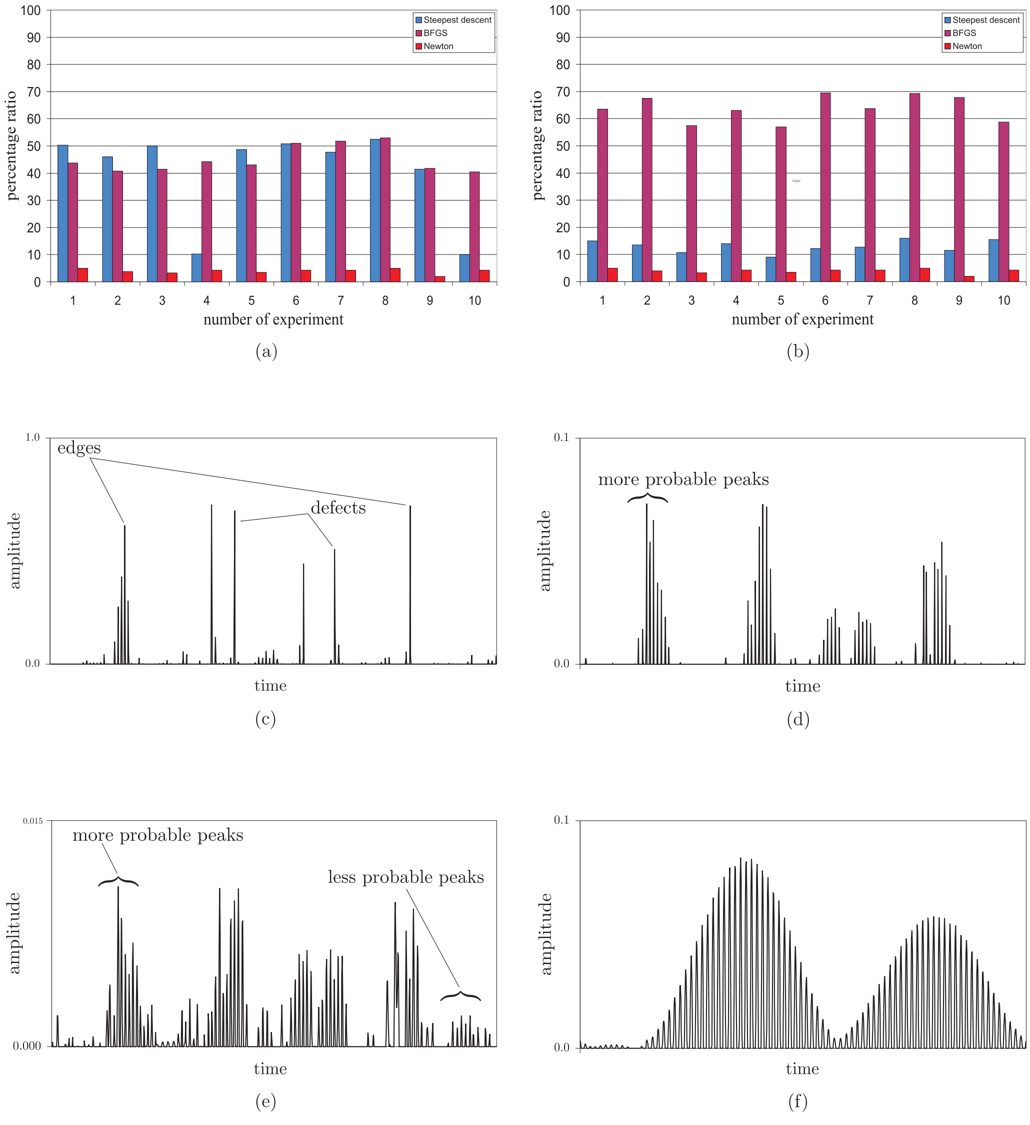


Figure 5.8: Applicability of methods of unconstrained optimization: (a) steepest descent, BGFS and Newton algorithms, no *bltSearch*; (b) steepest descent, BGFS and Newton algorithms, with *bltSearch*; (c) Newton algorithm, proper impulse response, with *bltSearch*; (d) Newton algorithm, improper impulse response, with *bltSearch*; (e) steepest descent, improper impulse response, with *bltSearch*; (f) steepest descent, improper impulse response, no *bltSearch*

Figure 5.8(d) is not optimal, it still gives information about defects locations. This optimizer does not contain much spurious peaks so that defects of the specimen can be easily detected. Such a good performance of the Newton algorithm can be explained through its convergence property. Thus, the Newton algorithm seeks for the best optimizer for the given input data and splits more probable information (real peaks) from less probable one (spurious peaks).

The optimizers computed by steepest descent algorithm for improper impulse response with and without backtracking line search are presented in Figures 5.8(e) and (f), respectively. As we can see even if the backtracking line search is used, the steepest descent algorithm does not provide the optimizer that is as good as the optimizer computed by Newton algorithm (compare Figures 5.8(d) and (e)). Later we will also show that the backtracking line search seriously slows down the steepest descent algorithm. The optimizer computed by the steepest descent algorithm without backtracking line search is represented in Figure 5.8(f). As we can see the search of the optimizer fails.

Let us consider the second step of the algorithms comparison. Here we compare the speed of convergence of the steepest descent, BFGS, and the Newton algorithms with applied backtracking line search.

In Figure 5.9(a) we represent how the entropy function increases with every iteration. The Newton algorithm shows the best convergence speed. It reaches the optimum in a few iterations. As we have already discussed in Section 5.5.2 is has quadratical speed of convergence. The steepest descent algorithm has a significantly lower convergence than the Newton algorithm, so it approaches the optimal solution very slowly. The BFGS algorithm gets stuck after some iterations far away from the correct solution. In the experiments both the steepest descent and BFGS algorithms do not terminate so that we needed to interrupt them manually.

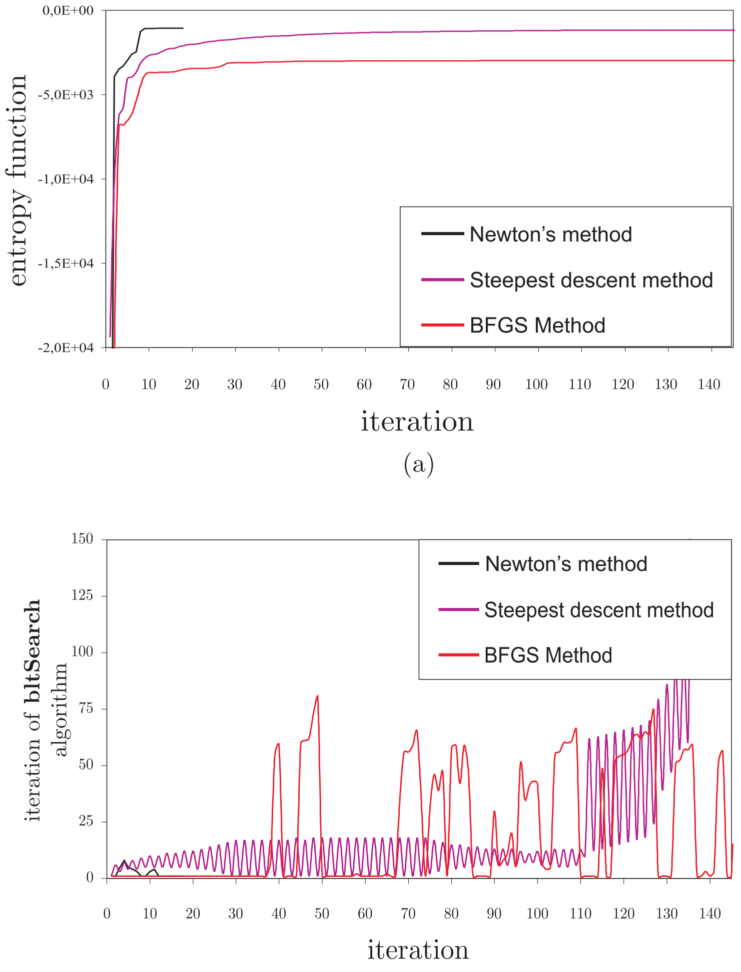
Another important criterion which can be used in the algorithms' comparison is the number of iterations required by the backtracking line search algorithm. If it is too large, the speed of computations may enormously decrease. In such a case even a fast algorithm like the steepest descent algorithm is not useful in practice. In Figure 5.9(b) we represent the number of iterations of the backtracking line search algorithm for every iteration of the main algorithm (i.e. steepest descent, BFGS, and Newton algorithm being applied for MED).

The Newton algorithm shows the best performance. It requires only a few iterations of backtracking line search.

The cost of the steepest descent technique increases, in particular, when the current solution approaches to the optimum. Here, we can see that the number of backtracking iterations rises.

The number of backtracking iterations in BFGS spontaneously jumps from some high to some low values.

In this subsection we have compared different algorithms of unconstrained optimization. We conclude that the Newton algorithm is the most efficient one for our task. In all the experiments it shows very high convergence speed and stability.



(a)

(b)

Figure 5.9: Convergence speed of methods of unconstrained optimization: (a) speed of entropy function descent; (b) backtracking line search iterations

5.5.4 Gradient and Hessian of the Entropy Function Ψ

The Newton algorithm requires computation of gradient and Hessian of the objective function at every iteration. In the current section we derive the gradient and the Hessian of the entropy function Ψ given in equation (5.22).

The gradient $\nabla\boldsymbol{\Psi} \in \mathbb{R}^n$ is defined for any $\mathbf{x} \in \mathbb{R}^n$ as

$$\nabla\boldsymbol{\Psi}_i = \frac{\partial\Psi}{\partial \mathrm{x}_i} = -\left(\frac{\partial S}{\partial \mathrm{x}_i} + \alpha\frac{\partial\chi}{\partial \mathrm{x}_i}\right) = -\left(\nabla\mathbf{S} + \alpha\nabla\boldsymbol{\chi}\right), \tag{5.37}$$

where $i \in [0 : n-1]$. The Hessian $\nabla^2\boldsymbol{\Psi} \in \mathbb{R}^{n\times n}$ is defined as gradient of $\nabla\boldsymbol{\Psi}$ for $i, j \in [0 : n-1]$, i.e:

$$\nabla^2\boldsymbol{\Psi}_{i,j} = \frac{\partial}{\partial \mathrm{x}_j}\nabla\boldsymbol{\Psi} = -\left(\frac{\partial^2 S}{\partial \mathrm{x}_i\partial \mathrm{x}_j} + \alpha\frac{\partial^2\chi}{\partial \mathrm{x}_i\partial \mathrm{x}_j}\right) = -\left(\nabla^2\mathbf{S} + \alpha\nabla^2\boldsymbol{\chi}\right). \tag{5.38}$$

The first terms of equations (5.37) and (5.38) are the first and second partial derivatives of S, see equation (5.23), so that

$$\nabla S_i = \frac{\partial S}{\partial \mathrm{x}_i} = \ln \mathrm{x}_i - \ln \mathrm{b}_i \tag{5.39}$$

and

$$\nabla^2 S_{i,j} = \frac{\partial S}{\partial \mathrm{x}_i\partial \mathrm{x}_j} = \frac{\delta(i-j)}{\mathrm{x}_i}, \tag{5.40}$$

where $\delta : \mathbb{Z} \to \mathbb{R}$ is the unit-sample. From equation (5.40) we conclude that $\nabla^2\mathbf{S}$ is a matrix which has non-zero values only on its main diagonal. We also say that $\nabla^2\mathbf{S}$ is not defined if there is at least one zero sample in $\mathbf{x}$.

By using the definition of the positive wrapped convolution given in equation (5.3) we define the signal $\mathbf{y} \in \mathbb{R}^n$ for all $\tau \in [0 : n-1]$ as

$$\mathrm{y}_\tau = (\mathbf{x} * \mathbf{h})_\tau = \sum_{k=0}^{\tau} \mathrm{x}_{\tau-k}\mathrm{h}_k + \sum_{k=\tau+1}^{n-1} \mathrm{x}_{n+\tau-k}\mathrm{h}_k, \tag{5.41}$$

where signals $\mathbf{x}$ and $\mathbf{h}$ are given in equation (5.24). A partial derivative of y_τ is

$$\frac{\partial \mathrm{y}_\tau}{\partial \mathrm{x}_i} = \mathrm{h}_{k(\tau,i)}, \quad \text{where} \quad k(\tau, i) = \begin{cases} \tau - i & \text{if} \quad \tau \geq i \\ \tau - i + n & \text{otherwise.} \end{cases} \tag{5.42}$$

In order to derive the gradient of χ we substitute equations (5.41) and (5.42) in to (5.24) so that for $i \in [0 : n-1]$ we have:

$$\begin{aligned}
\nabla\chi_i = \frac{\partial\chi}{\partial \mathrm{x}_i} &= \frac{1}{2\sigma^2}\sum_{\tau=0}^{n-1}\frac{\partial\left(\mathrm{y}'_\tau - (\mathbf{x} * \mathbf{h})_\tau\right)^2}{\partial \mathrm{x}_i} \\
&= \frac{1}{2\sigma^2}\sum_{\tau=0}^{n-1}\frac{\partial(\mathrm{y}'^2_\tau - 2\mathrm{y}'_\tau\mathrm{y}_\tau + \mathrm{y}^2_\tau)}{\partial \mathrm{x}_i} \\
&= \frac{1}{2\sigma^2}\sum_{\tau=0}^{n-1}\left(-2\mathrm{y}'_\tau\frac{\partial \mathrm{y}_\tau}{\partial \mathrm{x}_i} + 2\mathrm{y}_\tau\frac{\partial \mathrm{y}_\tau}{\partial \mathrm{x}_i}\right) \\
&= \frac{1}{\sigma^2}\sum_{\tau=0}^{n-1}\mathrm{h}_{k(\tau,i)}(\mathrm{y}_\tau - \mathrm{y}'_\tau).
\end{aligned} \tag{5.43}$$

The Hessian of χ is defined from equation (5.43) as follows:

$$\nabla^2 \chi_{i,j} = \frac{\partial}{\partial \mathrm{x}_j} \nabla \chi_i = \frac{1}{\sigma^2} \sum_{\tau=0}^{n-1} \mathrm{h}_{k(\tau,i)} \left(\frac{\partial \mathrm{y}_\tau}{\partial \mathrm{x}_j} - \frac{\partial \mathrm{y}'_\tau}{\partial \mathrm{x}_j} \right) = \frac{1}{\sigma^2} \sum_{\tau=0}^{n-1} \mathrm{h}_{k(\tau,i)} \, \mathrm{h}_{p(\tau,j)} \tag{5.44}$$

In general, a computation of the Hessian $\nabla^2 \mathbf{\Psi}$ at every iteration step of the Newton algorithm is a challenging task. In the following section we develop an algorithm for its fast computation.

5.5.5 Fast Hessian Computation

In the current section we prove some properties of Hessian $\nabla^2 \mathbf{\Psi}$. We also show how the fast computation of $\nabla^2 \mathbf{\Psi}$ can be done exploiting these properties.

The computation of the Hessian $\nabla^2 \mathbf{\Psi}$ requires computation of $\nabla^2 \mathbf{S}$ and $\nabla^2 \boldsymbol{\chi}$ given in equations (5.40) and (5.44), respectively. The matrix $\nabla^2 \boldsymbol{\chi}$ is constant for constant vector $\mathbf{h}$, therefore the only matrix which must be updated while the Newton algorithm iterates is $\nabla^2 \mathbf{S}$.

Since matrix $\nabla^2 \boldsymbol{\chi}$ only depends on the impulse response $\mathbf{h}$, it can be precomputed before the algorithm starts to iterate in order to save computational time. The number of impulse responses[4] is usually restricted hence the memory cost for presaving is not high.

In the following we prove that in order to derive $\nabla^2 \boldsymbol{\chi}$ it is only necessary to compute its first row, i.e. vector $\nabla^2 \boldsymbol{\chi}[0][0:n-1]$.

Theorem 5.5.5 *Let $\nabla^2 \boldsymbol{\chi} \in \mathbb{R}^{n \times n}$ be the Hessian of χ defined by equation* (5.44). *We prove that each entry $\nabla^2 \chi_{i,j}$ can be found from its first row $\nabla^2 \boldsymbol{\chi}[0][0:n-1]$ such that* for all $i \in [1:n-1]$ *and* $j \in [0:n-1]$*:*

$$\nabla^2 \chi_{i,j} = \nabla^2 \chi_{0,j'(i)}, \quad \text{where} \quad j'(i) = \begin{cases} j-i & \text{if} \quad j \geq i \\ j-i+n & \text{otherwise.} \end{cases} \tag{5.45}$$

Proof:

Let us rewrite equation (5.44) for $i,j \in [0:n-1]$ *as:*

$$\begin{aligned} \sigma^2 \; \nabla^2 \chi_{i,j} = & \sum_{\tau=0}^{\min(i,j)-1} \mathrm{h}_{n+\tau-i} \, \mathrm{h}_{n+\tau-j} \; + \\ & \sum_{\tau=\min(i,j)}^{\max(i,j)-1} \mathrm{h}_{n+\tau-\max(i,j)} \, \mathrm{h}_{\tau-\min(i,j)} \; + \\ & \sum_{\tau=\max(i,j)}^{n-1} \mathrm{h}_{\tau-i} \, \mathrm{h}_{\tau-j} \end{aligned} \tag{5.46}$$

Let us introduce three claims we need to show the goal of the theorem. The first claim represents the goal for the entries of the matrix $\nabla^2 \boldsymbol{\chi}$ which are placed above the main diagonal:

claim1 for all $i \in [1:n-2], j \in [i+1:n-1]$ holds $\nabla^2 \chi_{i,j} = \nabla^2 \chi_{0,j-i}$

[4] in practice we measure a number of impulse responses for defects of different forms

The second claim represents the goal for the entries of the matrix $\nabla^2\boldsymbol{\chi}$ which are places on the main diagonal:

$$\textbf{\textit{claim2}} \quad \text{for all} \quad i,j \in [1:n-1] \quad \text{s.t.} \quad i=j \quad \text{holds} \quad \nabla^2\chi_{i,j} = \nabla^2\chi_{0,j-i} = \nabla^2\chi_{0,0}$$

The third claim represents the goal for the entries of the matrix $\nabla^2\boldsymbol{\chi}$ which are placed below the main diagonal:

$$\textbf{\textit{claim3}} \quad \text{for all} \quad j \in [0:n-2], i \in [j+1:n-1] \quad \text{holds} \quad \nabla^2\chi_{i,j} = \nabla^2\chi_{0,n+j-i}$$

*Let us show the correctness of claim*1 *using the definition of $\nabla^2\boldsymbol{\chi}$ given in* (5.46) *and taking into account that $j > i$ (see conditions of claim1)):*

$$\sigma^2\ \nabla^2\chi_{0,j-i} = \sum_{\tau=0}^{j-i-1} \mathrm{h}_{n+\tau-j+i}\ \mathrm{h}_\tau + \sum_{\tau=j-i}^{n-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-j+i} \tag{5.47}$$

$$\sigma^2\ \nabla^2\chi_{i,j} = \underbrace{\sum_{\tau=0}^{i-1} \mathrm{h}_{n+\tau-i}\ \mathrm{h}_{n+\tau-j}}_{t_1} + \underbrace{\sum_{\tau=i}^{j-1} \mathrm{h}_{n+\tau-j}\ \mathrm{h}_{\tau-i}}_{t_2} + \underbrace{\sum_{\tau=j}^{n-1} \mathrm{h}_{\tau-i}\ \mathrm{h}_{\tau-j}}_{t_3} \tag{5.48}$$

In (5.48) *in terms t_1, t_2, t_3, without changing the equation sense, we shift the range of indexes of every sum by* $(n-i)$, $(-i)$ *and* $(-i)$, *respectively:*

$$\sigma^2\ \nabla^2\chi_{i,j} = \sum_{\tau=n-i}^{n-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-j+i} + \sum_{\tau=0}^{j-i-1} \mathrm{h}_{n+\tau-j+i}\ \mathrm{h}_\tau + \sum_{\tau=j-i}^{n-i-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-j+i} \tag{5.49}$$

Comparing (5.47) *and* (5.49) *we conclude that claim1 holds.*

*Let us show the correctness of claim*2 *using the definition of $\nabla^2\boldsymbol{\chi}$ given in* (5.46) *and taking into account that $i = j$ (see definition of claim2):*

$$\sigma^2\ \nabla^2\chi_{0,j-i} = \nabla^2\chi_{0,0} = \sum_{\tau=0}^{n-1} \mathrm{h}_\tau \mathrm{h}_\tau \tag{5.50}$$

$$\sigma^2\ \nabla^2\chi_{i,j} = \underbrace{\sum_{\tau=0}^{i-1} \mathrm{h}_{n+\tau-i}\ \mathrm{h}_{n+\tau-i}}_{t_1} + \underbrace{\sum_{\tau=i}^{n-1} \mathrm{h}_{\tau-i}\ \mathrm{h}_{\tau-i}}_{t_2} \tag{5.51}$$

In (5.51) *in terms t_1, t_2, without changing the equation sense, we shift the range of index of every sum at* $(n-i)$ *and* $(-i)$, *respectively:*

$$\sigma^2\ \nabla^2\chi_{i,j} = \sum_{\tau=n-i}^{n-1} \mathrm{h}_\tau\ \mathrm{h}_\tau + \sum_{\tau=0}^{n-i-1} \mathrm{h}_\tau\ \mathrm{h}_\tau \tag{5.52}$$

Comparing (5.50) *and* (5.52) *we conclude that claim2 holds.*

*Let us show the correctness of claim*3*, using the definition of* $\nabla^2\boldsymbol{\chi}$ *and taking into account that* $j < i$*:*

$$\sigma^2\ \nabla^2\chi_{0,n+j-i} = \sum_{\tau=0}^{n+j-i-1} \mathrm{h}_{\tau-j+i}\ \mathrm{h}_\tau + \sum_{\tau=n+j-i}^{n-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-n-j+i} \tag{5.53}$$

$$\sigma^2\ \nabla^2\chi_{i,j} = \underbrace{\sum_{\tau=0}^{j-1} \mathrm{h}_{n+\tau-i}\ \mathrm{h}_{n+\tau-j}}_{t_1} + \underbrace{\sum_{\tau=j}^{i-1} \mathrm{h}_{n+\tau-i}\mathrm{h}_{\tau-j}}_{t_2} + \underbrace{\sum_{\tau=i}^{n-1} \mathrm{h}_{\tau-i}\mathrm{h}_{\tau-j}}_{t_3} \tag{5.54}$$

In (5.54) *in terms* t_1, t_2, t_3*, without changing the equation sense, we shift the range of index of every sum at* $(n-i)$*,* $(n-i)$ *and* $(-i)$*, respectively:*

$$\sigma^2\ \nabla^2\chi_{i,j} = \sum_{\tau=n-i}^{n+j-i-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-j+i} + \sum_{\tau=n+j-i}^{n-1} \mathrm{h}_\tau\ \mathrm{h}_{\tau-n-j+i} + \sum_{\tau=0}^{n-i-1} \mathrm{h}_\tau\mathrm{h}_{\tau-j+i} \tag{5.55}$$

Comparing (5.53) *and* (5.55) *we conclude that claim3 holds.*

Combining correctness of claims 1, 2 *and* 3 *we reach the main goal of the theorem.* ■*.*

According to Theorem 5.5.5 we reduce the storage cost of $\nabla^2\boldsymbol{\chi}$ from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. We remind that typical value of n is 1024, 2048 and 4096 samples. We note that the computational cost of Hessian $\nabla^2\boldsymbol{\Psi}$ can be reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ as following. First, we presave $\nabla^2\boldsymbol{\chi}$. Then, at every new iteration $\nabla^2\boldsymbol{\Psi}$ is computed from the updated $\nabla^2\mathbf{S}$ at cost of $\mathcal{O}(n)$, because $\nabla^2\mathbf{S}$ is diagonal, and the presaved $\nabla^2\boldsymbol{\chi}$.

5.6 Computation of Hessian Inverse

At every iteration k the Newton algorithm requires computation of the descent vector $\triangle\mathbf{x}^k$ given in equation (5.35). The computation of $\triangle\mathbf{x}^k$ requires in its turn the computation of the inverse Hessian of the objective function. Since the computation of the matrix inverse is very expensive, the Newton algorithm performs very slowly. In the current chapter we develop a fast algorithm for the computation of the inverse of the Hessian matrix of entropy function Ψ.

5.6.1 Methods for Computation Hessian Inverse

In the literature many different algorithms, for the computation of the matrix inverse, were developed. Let us give a short review.

Definition 5.6.1 *A square matrix is said to be* ***singular*** *if its determinant is zero, i.e.*

$$|\mathbf{A}| \equiv \det(\mathbf{A}) = 0. \tag{5.56}$$

Definition 5.6.2 *The* ***inverse*** *of a square nonsingular matrix* $\mathbf{A}$ *is denoted by the symbol* $\mathbf{A}^{-1}$ *and is defined by the relation* [102, page 50]

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}, \tag{5.57}$$

where **I** *is a unit matrix.*

In the literature there is a number of standard algorithms to compute matrix inverses. The practical numerical calculation of inverses is based on a so-called *triangular factorization* [102, page 94]. For a nonsingular square matrix the **LU** factorization is under consideration. Here **L** and **U** are the *lower* and *upper* triangular matrixes of a matrix **A**, respectively. The inverse of $\mathbf{A} = \mathbf{LU}$ is computed in the following steps:

1. compute **Y** from $\mathbf{UY} = \mathbf{I}$
2. compute **X** from $\mathbf{LX} = \mathbf{Y}$.

The inverse of **A** is equal to **X**:

$$\mathbf{A}^{-1} = \mathbf{X}$$

If **A** is symmetric then another transformation can be applied [102, 135]. Here, the matrix **A** is represented as a product $\mathbf{LDL}^T$, where **D** is a diagonal matrix. In both approaches defined above the complexity of computation of $\mathbf{A}^{-1}$ is $\mathcal{O}(n^3)$ that is not efficient in practice.

A more efficient approach that can be applied for our task is to compute the matrix inverse as the solution of a system of linear equations [103]. Using equations (5.35), (5.37) and (5.38) we can write

$$\Delta\mathbf{x}^k = -\nabla^2\Psi(\mathbf{x}^k)^{-1} \cdot \nabla\Psi(\mathbf{x}^k),$$

what leads us to the following equation

$$-\nabla^2\Psi(\mathbf{x}^k) \cdot \Delta\mathbf{x}^k = \nabla\Psi(\mathbf{x}^k) \tag{5.58}$$

Let the coefficient matrix $\mathbf{A} = -\nabla^2\Psi(\mathbf{x}^k)$, the right-hand side vector $\mathbf{b} = \nabla\Psi(\mathbf{x}^k)$, and vector of unknowns $\mathbf{x} = \triangle\mathbf{x}^k$, then equation (5.58) can be written as

$$\mathbf{Ax} = \mathbf{b} \tag{5.59}$$

In mathematical analysis there are two general approaches for solving problems similar to equation (5.59). These include so-called direct (or classical) and indirect (or iterative) algorithms. Below we shortly discuss three main groups of the most important algorithms:

- basic iterative algorithms (Jacobi, Gauss-Seidel, **S**uccessive **O**ver **R**elaxation)
- projection or Krylov subspace algorithms (GMRES, CG, BCG, QMR etc.)
- classical algorithms (e.g. Gauss-Elimination algorithm)

The first group contains basic iterative algorithms which find solution of equation (5.59) in the following general form:

$$\mathbf{x}^{k+1} = \mathbf{G} \cdot \mathbf{x}^k,$$

where $\mathbf{G} \in \mathbb{R}^{n\times n}$ is a so-called iteration matrix. Matrix $\mathbf{G}$ depends on the concrete algorithm and is some function of $\mathbf{A}$ and $\mathbf{b}$. A stop-criterion is determined by a residual error $\parallel \mathbf{r} \parallel_2$, where the residual vector $\mathbf{r}$ is given by:

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^k. \tag{5.60}$$

The convergence of basic iterative algorithms depends on the conditional number of matrix $\mathbf{G}$, see [97, 104]. If the conditional number is very large[5], then the convergence of these methods dramatically decreases. Studies on the convergence rate of basic iterative algorithms discovered a difficulty of their use in many practical applications. In particular, the number of iterations enormously grows with the number of unknowns. In this work we have tested basic iterative algorithms mentioned above for finding the solution of equation (5.60) as a part of the MED algorithm. In all the tests the basic iterative algorithms failed. More information about this group of algorithms can be found in literature [95, 104].

The second group includes so-called projection (or Krylov subspace) algorithms. These algorithms offer great convergence properties on large linear systems of equations [104]. The projection algorithms extract the solution of a linear system from a *subspace* in an iterative fashion. In this thesis we use GMRES algorithm to solve equation (5.59). A short description of the GMRES algorithm is given in the following section.

The third group contains classical algorithms. In practice, these algorithms are not as popular as iterative ones because they are computationally very expensive. Despite that the classical algorithms extract the precise solution of a system of linear equation what may lead to the fast convergence of the Newton algorithm. In Section 5.7 we develop an optimized version of the Gauss-Elimination algorithm which utilize some properties of the impulse response $\mathbf{h}$.

5.6.2 Projection Algorithms

Definition 5.6.3 *The* ***vector space*** *is a set that is closed under finite vector addition and scalar multiplication. Euclidean n-space $\mathbb{R}^n$ is called a real vector space.*

Definition 5.6.4 *The vector space* ***spanned*** *by vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \in \mathbb{R}^n$ is the following set*

$$\text{span}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \equiv \{\, a_0\mathbf{v}_0 + a_1\mathbf{v}_1 + \ldots + a_{n-1}\mathbf{v}_{n-1} \mid a_0,\, a_1, \ldots,\, a_{n-1} \in \mathbb{R} \,\},$$

Definition 5.6.5 *A* ***basis*** *of a vector space $\mathbb{R}^n$ is defined to be a subset $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ of vectors in $\mathbb{R}^n$ that are linearly independent and span vector space $\mathbb{R}^n$.*

The idea of a *projection algorithm* is to extract an approximate solution of the problem (5.59) from a subspace of *vector space.* We refer to this subspace as a *search subspace* $\mathcal{K}_m$ such that $\mathcal{K}_m \subset \mathbb{R}^n$, where $m \in \mathbb{N}^+$ is called a *search subspace dimension.*

[5] it happens when the highest eigenvalue of $\mathbf{G}$ is much larger than the lowest eigenvalue of $\mathbf{G}$, i.e. $\lambda^{\mathbf{G}}_{max} \gg \lambda^{\mathbf{G}}_{min}$

In general, a search subspace dimension m is not large relative to n. This, certainly, makes projection algorithms very efficient for solving of large system of linear equations.

5.6.3 GMRES Algorithm

GMRES (Generalized Minimum RESidual) was proposed in 1986 as a Krylov subspace algorithm for systems $\mathbf{Ax} = \mathbf{b}$, [105] . This projection algorithm uses search subspace $\mathcal{K} = \mathcal{K}_m$ and some additional constraint subspace $\mathcal{L} = \mathcal{L}_m$. The m-th Krylov and constraint subspaces are defined as:

$$\mathcal{K}_m(\mathbf{A}, \mathbf{v}_0) = \text{span}(\mathbf{v}_0,\, \mathbf{A}\mathbf{v}_0,\, \mathbf{A}^2\mathbf{v}_0\,, \ldots,\, \mathbf{A}^{m-1}\mathbf{v}_0) \quad \text{and} \quad \mathcal{L}_m = \mathbf{A}\mathcal{K}_m,$$

with

$$\mathbf{v}_0 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|_2}$$

The vector $\mathbf{r}_0$ in the latter equation is computed as

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,$$

where $\mathbf{x}_0$ is the initial guess defined by the user.

In the literature the GMRES algorithm is represented as the solver for the least square problem where for iteration $m \geq 1$ holds:

$$\min_{\mathbf{x}_m \in \mathcal{K}_m} \|\mathbf{r}_m\|_2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}_m\|_2 \quad \text{such that} \quad \mathbf{x}_m \perp \mathcal{L}_m, \tag{5.61}$$

The solution of the problem (5.61) can be found by solving

$$\min_{\mathbf{y}_m \in \mathbb{R}^m} \|\beta \mathbf{e}_1 - \bar{\mathbf{H}}_m \mathbf{y}_m\|_2. \tag{5.62}$$

In equation (5.62) a constant β and vector $\mathbf{e}_1$ are defined in [106] as:

$$\beta = \|\mathbf{r}\|_2,$$
$$\mathbf{e}_1 = (1, 0, \ldots, 0) \in \mathbb{R}^{m+1},$$

and the matrix $\bar{\mathbf{H}}_m \in \mathbb{R}^{m+1 \times m}$ we discuss later.

The vector $\mathbf{x}_m$ which minimize equation (5.61) is computed from the solution of equation (5.62) as:

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbb{V}_m \mathbf{y}_m, \tag{5.63}$$

such that $\mathbf{x}_0$ is the initial guess. The matrix $\mathbb{V}_m \in \mathbb{R}^{n \times m}$ is computed through a so-called Arnoldi's process [104].

The Matrix $\bar{\mathbf{H}}_m$ is derived by the GramSchmidt orthogonalization procedure, see for reference [107]. In the literature this procedure is referred to as a source of significant numerical errors. In order to reduce numerical errors an alternative implementation was outlined in [108] and called Householder transformation. In the literature it was experimentally shown that Householder transformation is numerically more stable than the GramSchmidt orthogonalization procedure, specially as the limits of residual reduction are reached. The implementation of GMRES algorithm we applied in this work is given in Appendix B.0.5.

In [105] it was shown that using properties of matrix $\bar{\mathbf{H}}_m$ a problem (5.62) can be solved very efficiently. Since m is usually small (i.e. $m \ll n$) the solving of (5.62) does not slow down the total computational speed of the GMRES algorithm.

In general, Krylov subspace algorithms and GMRES in particular do not guarantee reaching an acceptable solution within efficient computational time and storage. For instance, the GMRES algorithm may lead to the exact arithmetic solution only after n iterations with $O(n^3)$ operations (where n is the size of the problem). The convergence of the GMRES algorithm depends on properties of matrix $\mathbf{A}$ and vector $\mathbf{b}$. In order to increase the convergence speed of the algorithm a special operator $\mathbf{K}$ is developed. $\mathbf{K}$ needs to be such that the products $\mathbf{K}^{-1}\mathbf{A}$ and $\mathbf{K}^{-1}\mathbf{b}$ have better properties than $\mathbf{A}$ and $\mathbf{b}$. In the ideal case holds:

$$\mathbf{K}^{-1}\mathbf{A}\,\mathbf{x} = \mathbf{I}\,\mathbf{x} = \mathbf{K}^{-1}\,\mathbf{b}, \tag{5.64}$$

so that the GMRES will deliver the correct solution in one step. A system of linear equations (5.64) is called *preconditioned system* and $\mathbf{K}$ is called *preconditioner*. Certainly, in practice equality (5.64) is not reachable (at least in a fast way). Instead, it is suggested to develop $\mathbf{K}$ in such a way that the matrix $\mathbf{K}^{-1}\mathbf{A}$ becomes sparse[6]. Generally, it is also required that operator $\mathbf{K}$ has the following properties:

- $\mathbf{K}$ is a good approximation of $\mathbf{A}$ i.e. matrix $\mathbf{K}$ is close to $\mathbf{A}$
- The cost of the construction of $\mathbf{K}$ is not prohibitive
- The preconditioned system becomes sparse.

In the literature there is no general theory how we can select a preconditioner $\mathbf{K}$ having the properties mentioned above. Selection and construction of a good preconditioner for a given class of problems is often based on an educated guess. In practice preconditioners are explicitly developed for specific problems in order to exploit their features. There is a great freedom in definition and construction of preconditioners for Krylov subspace algorithms. This makes them very popular and successful. In this thesis we do not give an overview of already known techniques to find preconditioner. We refer the reader to [104, 109–114] for more information.

In the following subsection we will construct a new preconditioner which makes the GMRES algorithm very efficient for solving the problem given in (5.59).

5.6.4 MED Preconditioner

As we have already discussed in Section 5.5.3 the Newton algorithm solves the entropy problem (5.22) in an iterative fashion. At iteration step k the Newton algorithm requires computation of both Hessian $\mathbf{A}^k$ and gradient $\mathbf{b}^k$. In Section 5.5.5 we have introduced the fast computation of the Hessian $\mathbf{A}^k$ at cost of $\mathcal{O}(n)$. The solution

$$\mathbf{A}^k\,\mathbf{x}^k = \mathbf{b}^k \tag{5.65}$$

of the Newton algorithm (i.e. vector $\mathbf{x}^k$) at iteration k is done with the GMRES algorithm, see Section 5.6.3. The flowchart of this process is represented in Figure

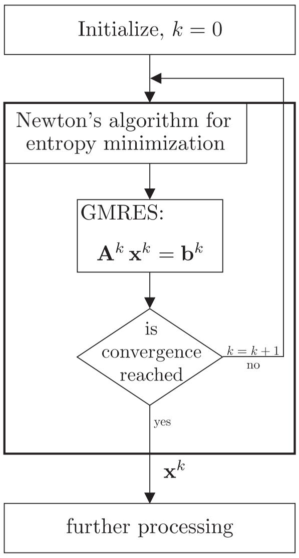


Figure 5.10: Flowchart: GMRES linear solver inside Newton algorithm

5.10(a). After the initialization phase the Newton algorithm iterates until the convergence is reached. Then, vector $\mathbf{x}^k$ is provided for the further processing. In the following we will experimentally show that the naive solution of (5.65) with GMRES may lead to a very low computational speed. In order to increase the speed of computations we apply the GMRES algorithm with a fast preconditioner, which we develop in the following.

The development of GMRES preconditioner $\mathbf{K}$ is based on the two-sided preconditioning scheme [115, page 176] formulated as:

$$\mathbf{K}_1^{k^{-1}}\,\mathbf{A}^k\ \ \mathbf{K}_2^{k^{-1}}\,\mathbf{z}^k = \mathbf{K}_1^{k^{-1}}\,\mathbf{b}^k \tag{5.66}$$

such that

$$\mathbf{x}^k = \mathbf{K}_2^{k^{-1}}\,\mathbf{z}^k \quad \text{and} \quad \mathbf{K}^k = \mathbf{K}_1^k\,\mathbf{K}_2^k.$$

The ideal preconditioner $\mathbf{K}^k$ in equation (5.66) can be computed from a so-called *LU decomposition* of Hessian $\mathbf{A}^k$ (see for reference [103]) as

$$\mathbf{A}^k = \mathbf{L}^k\,\mathbf{U}^k \tag{5.67}$$

where $\mathbf{K}_1^k = \mathbf{L}^k$ and $\mathbf{K}_2^k = \mathbf{U}^k$ are lower triangular and upper triangular matrices.

Let us introduce the standard way to compute a LU-decomposition of a matrix $\mathbf{A}$ such that $\mathbf{A} = \mathbf{LU}$ and $\mathbf{A}, \mathbf{L}, \mathbf{U} \in \mathbb{R}^{n\times n}$. The entries of matrices $\mathbf{L}$ and $\mathbf{U}$ are defined for all $i, k \in [0 : n-1]$ as:

$$\mathrm{l}_{i,k} = \begin{cases} (\,\mathrm{a}_{i,k} - \sum\limits_{j=0}^{k-1} \mathrm{l}_{i,j}\,\mathrm{u}_{j,k}\,)\,/\,\mathrm{u}_{k,k} & \text{if} \quad i > k \\ 1 & \text{if} \quad i = k \\ 0 & \text{if} \quad i < k \end{cases} \tag{5.68}$$

[6] sparse matrix is a matrix which has many zero entries

and

$$\mathrm{u}_{i,k} = \begin{cases} \mathrm{a}_{i,k} - \sum\limits_{j=0}^{i-1} \mathrm{l}_{i,j}\, \mathrm{u}_{j,k} & \text{if} \quad i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{5.69}$$

We notice that $\mathbf{L}$ is the lower triangular matrix and its upper entries are zeros. Matrix $\mathbf{U}$ is the upper triangular matrix. More information about LU decomposition can be found in the literature [102,103,116].

Complexity of LU decomposition $\mathcal{O}(n^3)$. It is very expensive to perform the decomposition at every iteration of the Newton algorithm k. Instead we suggest to use the standard algorithm only in order to decompose the Hessian $\mathbf{A}^0$ into $(\mathbf{L}^0, \mathbf{U}^0)$ before the Newton algorithm starts to iterate. At further iterations for $k \geq 1$ matrices $\mathbf{L}^k$ and $\mathbf{U}^k$ are not computed with expensive LU decomposition procedure rather they are updated in a very cheap way. Thus, at every iteration with $k \geq 1$ the updated preconditioner $\mathbf{K}^k$ is computed as:

$$\mathbf{K}^k \simeq \mathbf{A}^k \simeq \mathbf{L}^k\, \mathbf{U}^k, \quad \text{where} \quad \mathbf{L}^k = \mathbf{L}^0 \quad \text{and} \quad \mathbf{U}^k = (\mathbf{U}^0 + \mathbf{\Lambda}^k). \tag{5.70}$$

In equation (5.70) $\mathbf{\Lambda}^k$ stands for a diagonal matrix which contains the difference between $\mathbf{A}^0$ and $\mathbf{A}^k$ on the main diagonal. According to the definition of the Hessian of the entropy function (see Section 5.5.5), we introduce $\mathbf{\Lambda}^k$ in terms of the Hessian of S function as:

$$\mathbf{\Lambda}^k = \mathbf{A}^k - \mathbf{A}^0 = \nabla^2\mathbf{S}^k - \nabla^2\mathbf{S}^0.$$

The matrix $\mathbf{\Lambda}^k$ is diagonal because all $\nabla^2\mathbf{S}^k$ are diagonal.

IF the matrix $\mathbf{L}^0$ is approximately equal to the matrix $\mathbf{I}$ i.e.

$$\mathbf{L}^0 \approx \mathbf{I} \approx \mathbf{L^0}^{-1} \tag{5.71}$$

the following holds:

$$\mathbf{A}^k = \mathbf{A}^0 + \mathbf{\Lambda}^k = \mathbf{L}^0\, \mathbf{U}^0 + \mathbf{\Lambda}^k = \mathbf{L}^0\, (\, \mathbf{U}^0 + \mathbf{L}^{0^{-1}} \mathbf{\Lambda}^k) \approx \mathbf{L}^0\, (\mathbf{U}^0 + \mathbf{\Lambda}^k). \tag{5.72}$$

Theorem 5.6.6 *Let* $\mathbf{A} = -\left(\nabla^2\mathbf{S} + \nabla^2\chi\right)$ *be the Hessian of the entropy function at the* 0*-th iteration of the Newton algorithm. The matrixes* $\mathbf{L}$ *and* $\mathbf{U}$ *are computed from the LU decomposition of the matrix* $\mathbf{A}$ *such that* $\mathbf{LU} = \mathbf{A}$.

Let $m > n$ *and the vector* $\mathbf{x}^0 \in \mathbb{R}^n$ *be initialized by the following value* b*:*

$$b = \frac{1}{2^m\, \alpha\, \nabla^2\chi_{max}}, \quad \text{where} \quad \nabla^2\chi_{max} = \max_{i,j}\left(\left|\nabla^2\chi_{i,j}\right|\right), \nabla^2\chi_{max} > 1;$$

such that for all $k \in [0 : n-1]$ *holds* $\mathrm{x}_k = b$*, see for reference* (5.38), (5.40).

Let us define a set for the index $i \in [0 : n-2]$ *as*

$$\mathrm{S}_i^l = \{\, \mathrm{l}_{j',i'} \mid j' \in [i+1 : n-1] \quad \text{and} \quad i' \in [0 : i]\,\}$$

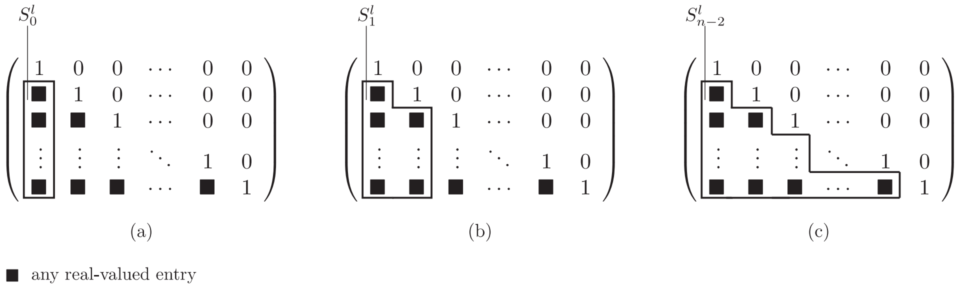


Figure 5.11: Elements of the set S_i^l: (a) entries of S_0^l; (b) entries of S_1^l; (c) entries of S_{n-2}^l

For every i *the set* S_i^l *contains entries of the matrix* $\mathbf{L}$ *which are located in columns from* 0 *to* i *below the main diagonal, see Figure 5.11.*

We claim that

$$\text{for all} \quad \mathrm{l} \in S_{n-2}^l \quad \text{holds} \quad |\,\mathrm{l}\,| < \varepsilon, \; 0 < \varepsilon < 1 \tag{5.73}$$

In (5.73) ε *stands for the limit value of* S_{n-2}^l *and is given as*

$$\varepsilon = \frac{1}{2^{m-n-1}}$$

Proof by induction on i that $\forall \mathrm{l} \in S_i \,:\, |\,\mathrm{l}\,| \leq \varepsilon$

1. Induction base*, i=0:*

By definition of the set S_i^l *we have:*

$$\mathrm{S}_0^l = \{\, \mathrm{l}_{j',0} \mid j' \in [1 : n-1] \,\}$$

In order to compute the elements of S_0^l *we utilize* LU *decomposition* **??***, definition of matrix* $\mathbf{A}$ *and definition of claim* (5.73)*:*

$$\mathrm{l}_{j',0} \overset{(5.68,5.69)}{=} \frac{\mathrm{a}_{j',0}}{\mathrm{a}_{0,0}} = \frac{\alpha\nabla^2\chi_{j',0}}{2^m\,\alpha\nabla^2\chi_{max} + \alpha\nabla^2\chi_{0,0}} \tag{5.74}$$

We estimate the absolute value of $\mathrm{l}_{j',0}$ *given in* (5.74) *as:*

$$|\,\mathrm{l}_{j',0}\,| \;\leq\; \left(\frac{\alpha\nabla^2\chi_{max}}{2^m\,\alpha\nabla^2\chi_{max} - \alpha\nabla^2\chi_{max}}\right)$$

We simplify the latter inequality as

$$|\,\mathrm{l}_{j',0}\,| \;\leq\; \left(\frac{1}{2^m - 1}\right) \tag{5.75}$$

From inequality (5.75) *and definition of* ε *we conclude that for all* $\mathrm{l}_{j',0}$ *from interval* (5.75) *holds*

$$\left|\,\mathrm{l}_{j',0}\,\right| < \varepsilon$$

2. Induction step $i \to i+1$*:*

By using the definition of the LU decomposition we have

$$\mathrm{u}_{i+1,i+1} = 2^m\,\alpha\nabla^2\chi_{max} + \underbrace{\alpha\nabla^2\chi_{i+1,i+1} - \sum_{j'=0}^{i} \mathrm{l}_{i+1,j'}\,\mathrm{u}_{j',i+1}}_{\mathrm{u}^*_{i+1,i+1}}\,. \tag{5.76}$$

We estimate the absolute value of the term $\mathrm{u}^*_{i+1,i+1}$ *in* (5.76)*. For that purpose we use the induction hypothesis, namely*

$$\forall\ \mathrm{l} \in S^l_i \quad \text{holds} \quad |\,\mathrm{l}\,| < \varepsilon.$$

Since all the elements $\mathrm{l}_{i+1,j'}$ *in* (5.76) *are from* S^l_i *we have* $\left|\,\mathrm{l}_{i+1,j'}\,\right| < \varepsilon$ *and* $\left|\,\mathrm{l}_{i+1,j'}\,\right| < 1$*, then*

$$\left|\,\mathrm{u}^*_{i+1,i+1}\,\right| \;\leq\; \left(\left|\,\alpha\nabla^2\chi_{i+1,i+1}\,\right| + \sum_{j'=0}^{i}\left|\,\mathrm{u}_{j',i+1}\,\right|\right) \tag{5.77}$$

The sum in (5.77) *can be represented in the following form:*

$$\sum_{j'=0}^{i}\left|\,\mathrm{u}_{j',i+1}\,\right| = |\,\mathrm{u}_{0,i+1}\,| + |\,\mathrm{u}_{1,i+1}\,| + |\,\mathrm{u}_{2,i+1}\,| + \ldots + |\,\mathrm{u}_{i,i+1}\,|\,, \tag{5.78}$$

where

$$\begin{aligned}
&|\,\mathrm{u}_{0,i+1}\,| = |\,\alpha\nabla^2\chi_{0,i+1}\,|\\
&|\,\mathrm{u}_{1,i+1}\,| \leq |\,\alpha\nabla^2\chi_{1,i+1}\,| + |\,\mathrm{u}_{0,i+1}\,|\\
&|\,\mathrm{u}_{2,i+1}\,| \leq |\,\alpha\nabla^2\chi_{2,i+1}\,| + |\,\mathrm{u}_{0,i+1}\,| + |\,\mathrm{u}_{1,i+1}\,|\\
&\vdots\\
&|\,\mathrm{u}_{i,i+1}\,| \leq |\,\alpha\nabla^2\chi_{i,i+1}\,| + |\,\mathrm{u}_{0,i+1}\,| + |\,\mathrm{u}_{1,i+1}\,| + \ldots + |\,\mathrm{u}_{i-1,i+1}\,|
\end{aligned}$$

Without changing the sense of (5.77) *we substitute instead of all* $\nabla^2\chi_{i,j}$ *in* (5.77) *and* (5.78) *the constant* $\nabla^2\chi_{max}$*. Under these considerations the sequence*

$$|\,\mathrm{u}_{0,i+1}\,|,\ |\,\mathrm{u}_{1,i+1}\,|,\ |\,\mathrm{u}_{2,i+1}\,|\,, \ldots\,, |\,\mathrm{u}_{i,i+1}\,|$$

turns to be a geometrical progression with a base $\triangle = 2$*. The prof of the latter claim is given in* Theorem 5.6.7*. Since that, the sum in equation* (5.78) *can be written as*

$$\alpha\nabla^2\chi_{max} + 2\alpha\nabla^2\chi_{max} + 4\alpha\nabla^2\chi_{max} + \ldots + 2^i\,\alpha\,\nabla^2\chi_{max} = (2^{i+1}-1)\alpha\nabla^2\chi_{max} \tag{5.79}$$

By substitution (5.79) *in* (5.77) *we have*

$$\left|\, \mathrm{u}^{*}_{i+1,i+1} \,\right| \;\leq\; 2^{i+1}\, \alpha \nabla^{2} \chi_{max} \tag{5.80}$$

Let us consider all the new elements $\mathrm{l}_{j,i+1}$ *which are computed at the* $(i+1)$*-th step of the induction for all* $j \in [i+2 : n-1]$*:*

$$\mathrm{l}_{j,i+1} = \underbrace{\left(\mathrm{a}_{j,i+1} - \sum_{j'=0}^{i} \mathrm{l}_{j,j'}\, \mathrm{u}_{j',i+1} \right)}_{\mathrm{l}^{*}_{j,i+1}} / \mathrm{u}_{i+1,i+1}, \tag{5.81}$$

We apply similar considerations to the term $\mathrm{l}^{*}_{j,i+1}$ *in* (5.81) *as we have done for* $\mathrm{u}^{*}_{i+1,i+1}$ *in* (5.76)*. Thus, we derive the interval for all possible values of* $\mathrm{l}^{*}_{j,i+1}$ *as*

$$\left|\, \mathrm{l}^{*}_{j,i+1} \,\right| \;\leq\; \left(2^{i+1}\, \alpha \nabla^{2} \chi_{max}\right) \tag{5.82}$$

Let us prove that $|\, \mathrm{l}_{j,i+1} \,| < \varepsilon$*. For that we substitute in* (5.81) *the extreme values of* $\mathrm{u}^{*}_{i+1,i+1}$ *and* $\mathrm{l}^{*}_{j,i+1}$ *given in equations* (5.80) *and* (5.82) *and definition of* $\mathrm{u}_{i+1,i+1}$ *in* (5.76)*. Then, we have to prove that*

$$\left| \frac{2^{i+1}\, \alpha \nabla^{2} \chi_{max}}{2^{m}\, \alpha \nabla^{2} \chi_{max} - 2^{i+1}\, \alpha \nabla^{2} \chi_{max}} \right| < \varepsilon$$

The latter equation can be simplified as

$$\frac{1}{2^{m-i-1}-1} < \frac{1}{2^{m-n-1}} = \varepsilon \tag{5.83}$$

The inequality (5.83)*, what introduces the correctness of the induction step. Combining the base and the induction step we prove the theorem goal* (5.73) ■.

Theorem 5.6.7 *According to the sum* (5.77) *and our considerations about equation* (5.78) *we write*

$$U_{j'} = \sum_{j=0}^{j'-1} U_{j} + K, \tag{5.84}$$

where $j' \in [1 : i]$*,* $U_{j'} = \left|\, \mathrm{u}_{j',i+1} \,\right|$*,* $K = \nabla^{2} \chi_{max}$ *and* $U_0{=}K$*.*

We claim that a sequence U_0*,* U_1*,* U_2*,* $\ldots$ *,*U_i *is a geometrical progression with a base* $\triangle = 2$*, i.e. any element of the progression for* $j' \in [1 : i]$ *can be computed from its previous element as*

$$U_{j'} = \triangle\, U_{j'-1} \tag{5.85}$$

Proof

By using equation (5.84) *we have for* $j' \in [1:i]$*:*

$$\begin{aligned} U_{j'} &= \sum_{j=0}^{j'-1} U_j + K \\ &= \sum_{j=0}^{j'-2} U_j + K + U_{j'-1} \\ &= \sum_{j=0}^{j'-2} U_j + K + \sum_{j=0}^{j'-2} U_j + K \\ &= \triangle\, U_{j'-1} \qquad \blacksquare \end{aligned}$$

From **Theorem 5.6.6** we conclude that by setting ε close to zero (i.e. big m) we force all the elements $\mathrm{l} \in S^l_{n-2}$ to be also close to zero. Hence, the matrix $\mathbf{L}^0$ turns to be close to the unit matrix (i.e. $\mathbf{L}^0 \approx \mathbf{I}$) so that the assumption (5.71) holds.

In the following we compare computational efficiency of the GMRES preconditioner given in (5.70) with a number of standard preconditioners. Let us denote $\mathbf{K}^k_1 = \mathbf{L}^k\mathbf{U}^k$ to be a preconditioner computed from equation (5.70); $\mathbf{K}^k_2 = \mathbf{L}^0\mathbf{U}^0$ to be a preconditioner computed at 0-th iteration of GMRES; $\mathbf{K}^k_3$ to be diagonal precontitioner[7]; and $\mathbf{K}^k_4 = \mathbf{I}$ to be the unit-matrix preconditioner; index k denotes iteration of the Newton algorithm.

In Figures 5.12(a),(c),(e) and (g) we represent the number of iterations of GMRES algorithm which are required for each iteration of the Newton algorithm. Figure 5.12(b),(d),(f) and (h) shows increasing of the entropy function at every iteration of the Newton algorithm.

We observed the fastest convergence of the GMRES algorithm with preconditioner $\mathbf{K}^k_1$, see Figures 5.12(a) and (b). In this case only a few iterations of the Newton algorithm are required to reach convergence. The number of GMRES iterations per one Newton iteration is also small.

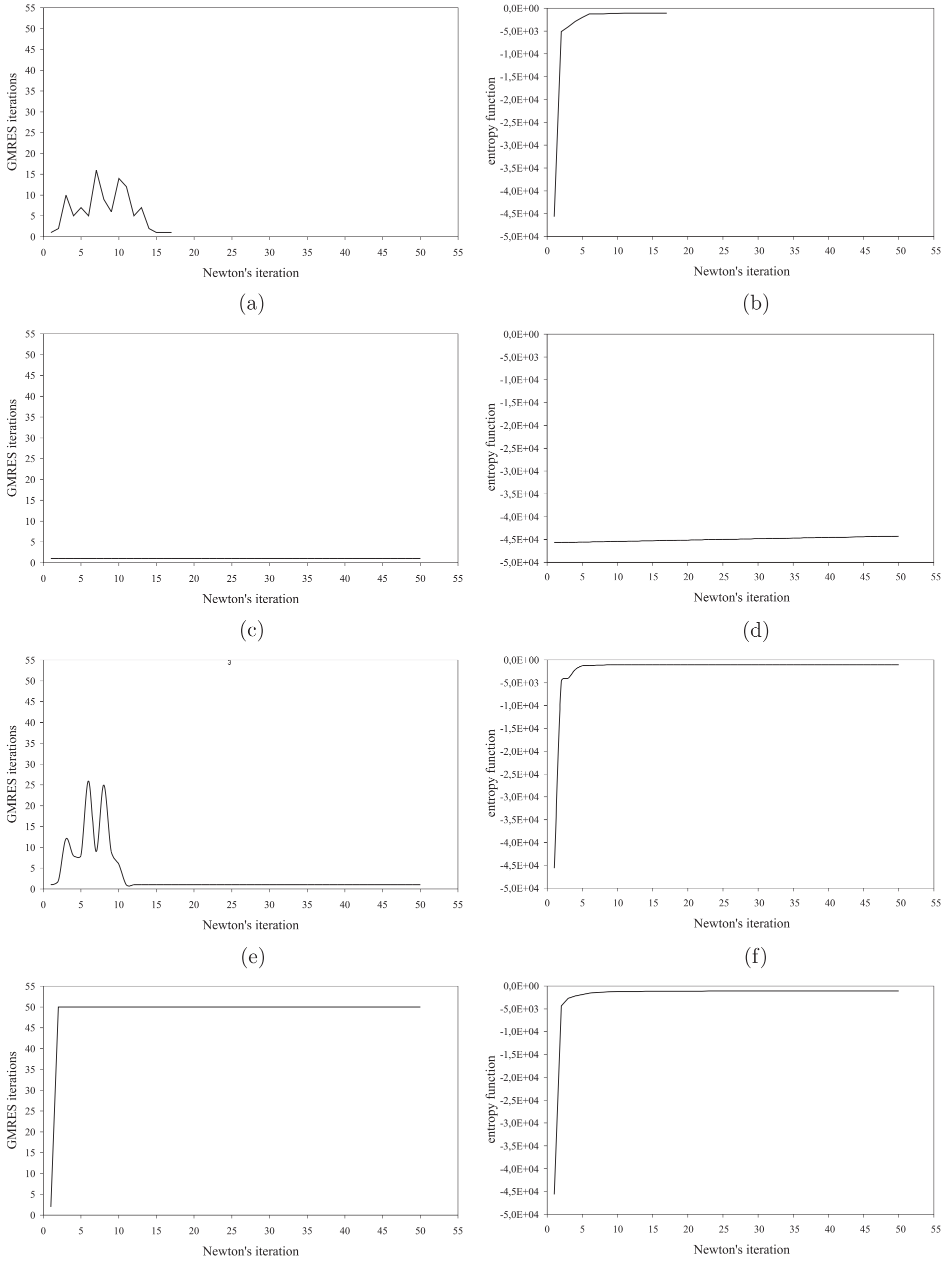
The GMRES with preconditioner $\mathbf{K}^k_3$ also requires few GMRES iterations for each Newton iteration. However, in the experiment this preconditioner does not allow the Newton algorithm to reach convergence, see Figure 5.12(e) and (f). In the experiment we had to interrupt the computation manually.

GMRES with $\mathbf{K}^k_2$ performs extremely badly. Even if it requires a small number of GMRES iterations (see Figure 5.12(c)) the entropy function increases very slowly (5.12(d)). In this case convergence is also not reached.

The last preconditioner we examine is $\mathbf{K}^k_4$. This preconditioner requires many GMRES iterations per Newton iteration. It affects the speed of computations, see slow increasing of entropy function in Figure 5.12(g) and (h).

In the following section we will compare the computational speed of the GMRES and Gauss elimination algorithm.

[7] i.e. the matrix $\mathbf{K}^k_3$ has diagonal entries of $\nabla^2\mathbf{\Psi}^k$ on its main diagonal



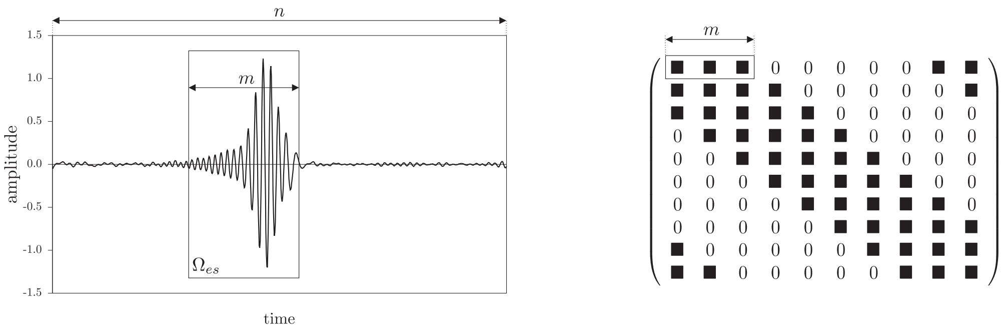
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 5.12: Comparison of GMRES preconditioners: (a) GMRES iterations for $\mathbf{K}_1^k$; (b) entropy function for $\mathbf{K}_1^k$; (c) GMRES iterations for $\mathbf{K}_2^k$; (d) entropy function for $\mathbf{K}_2^k$; (e) GMRES iterations for $\mathbf{K}_3^k$; (f) entropy function for $\mathbf{K}_3^k$; (g) GMRES iterations for $\mathbf{K}_4^k$; (h) entropy function for $\mathbf{K}_4^k$;



$$\begin{pmatrix}
\blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 & 0 & 0 & \blacksquare & \blacksquare \\
\blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 & 0 & 0 & \blacksquare \\
\blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 & 0 & 0 \\
0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 & 0 \\
0 & 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 \\
0 & 0 & 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 \\
0 & 0 & 0 & 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 \\
0 & 0 & 0 & 0 & 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\
\blacksquare & 0 & 0 & 0 & 0 & 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\
\blacksquare & \blacksquare & 0 & 0 & 0 & 0 & 0 & \blacksquare & \blacksquare & \blacksquare
\end{pmatrix}$$

(a)

(b)

Figure 5.13: Effective size of the impulse response: (a) impulse response **h** of length n with effective size m; (b) sparse structure of the Hessian

5.7 Gauss Elimination Algorithm

The GMRES algorithm which we applied for solving of the system of linear equations (5.65) is a good choice for scientific computations[8] but not for the industry. The mathematical complexity of the GMRES algorithms makes it hardly implementable in hardware[9]. For the hardware implementation a good substitution of the GMRES is the Gauss Elimination algorithm (or simply GE).

The complexity of the GE algorithm is $\mathcal{O}(n^3)$ and is equivalent to complexity of the matrix inverse. By utilizing some properties of the impulse inverse the cost of the GE can be reduced. In the following, at first, we introduce classical the GE algorithm. Then we define the *effective size* of the impulse response **h** and define the optimized Gauss elimination (OGE) algorithm.

Definition 5.7.1 *Let* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *be the Hessian of the entropy entropy function* Ψ *given in equation* (5.22)*. The standard way to compute matrix* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *which is the Gauss Elimination of the matrix* $\mathbf{A}$ *is defined in the following equation for all* $k \in [0 : n-1]$*,* $i \in [k : n-1]$ *and* $j \in [k-1 : n-1]$ *as:*

$$\mathrm{b}_{i,j}^{k} = \begin{cases} \mathrm{a}_{i,j} & \text{if } \ k = 0 \\ \mathrm{b}_{i,j}^{k-1} - \mathrm{b}_{k-1,j}^{k-1} \left(\dfrac{\mathrm{b}_{i,k-1}^{k-1}}{\mathrm{b}_{k-1,k-1}^{k-1}} \right), & \text{otherwise} \end{cases}$$

where $\mathrm{b}_{i,j}^{k}$ *denotes the entry of the matrix* $\mathbf{B}$ *computed at the iteration step* k*.*

The result of the Gauss Elimination algorithm is an upper triangular matrix.

Computational cost of the Gauss Elimination procedure can be reduced if the effective size of the impulse response **h** is taken into account. Under the term effective size we understand an amount of consecutive entries of **h** which participate in the oscillations. The effective size can be defined by the threshold. In Figure 5.13(a) we introduce the area Ω_{es} of length m which contains such entries. We

[8] High Performance Computing Clusters (HPCC), Distributed computing

[9] Application Specific Integrated Circuits (ASIC), Field-Programmable Gate Array (FPGA) etc.

note that $m < n$, where n is the length of $\mathbf{h}$. We assign all the entries of $\mathbf{h}$ outside Ω_{es} to be zeros.

A zero-filled impulse response $\mathbf{h}$ turns the Hessian of the entropy function to have sparse structure shown in Figure 5.13(b). By exploiting the sparse structure of the Hessian we can reduce the computational cost of the GE algorithm.

Let us develop the optimized version of the Gauss Elimination algorithm (OGE). The OGE differs from the classic algorithm only by redefining range of indexes i and j in Definition 5.7.1. In the OGE they depend on the effective size of the impulse response $\mathbf{h}$ for all $k > 0$ as:

$$i \in \begin{cases} i \in [k : k+m-2,\ n-m+1 : n-1] & \text{if} \quad k < n-2m+2 \\ i \in [k : n-1] & \text{otherwise} \end{cases}$$

and

$$j \in \begin{cases} j \in [k-1 : k+m-2,\ n-m+1 : n-1] & \text{if} \quad k < n-2m+2 \\ j \in [k-1 : n-1] & \text{otherwise} \end{cases}$$

The latter indexing scheme of i and j excludes zero-entries of the Hessian matrix being used by the Gauss Elimination algorithm. Such a re-indexing reduces the cost of GE algorithm from $O(n^3)$ to $\mathcal{O}(nm^2)$.

5.8 Speed Comparison Issue

In the current section we compare the computational speed of the MED algorithm for different solvers of the system of linear equations (5.65). We test the following solvers: the GMRES with updated LU preconditioner defined in (5.70), classical and optimized Gauss-elimination algorithms (i.e. GE and OGE, respectively). We examine measured data of different size, namely 512, 1024, 2048, and 4096 samples. In Figure 5.14 we present the time of computation of the MED algorithm expressed in seconds. Here, the time required for processing of a line-scan is presented.

The MED with GMRES shows the best performance for all problem sizes. The performance of the GE algorithm heavily decreases with problem size. Thus, in the case of 4096 samples it requires about 48 minutes. The optimized version of GE algorithm shows better performance than the classical version. In this test the effective size is 150, 300, 600, and 1200 for problem sizes of 512, 1024, 2048, and 4096 samples, respectively.

5.9 MED Algorithm Results

5.9.1 Detection of sharp defects with MED (1D case)

In the current section we test the ability of the MED algorithm to detect sharp defects and compare it with the Doppler amplitude, Doppler frequency, and deconvolution approaches discussed in Sections 5.1 and 5.3.3, respectively.

In Figures 5.1(b)-(h) we examined the resolution ability of the Doppler amplitude and the Doppler frequency approaches applied to the signals measured for

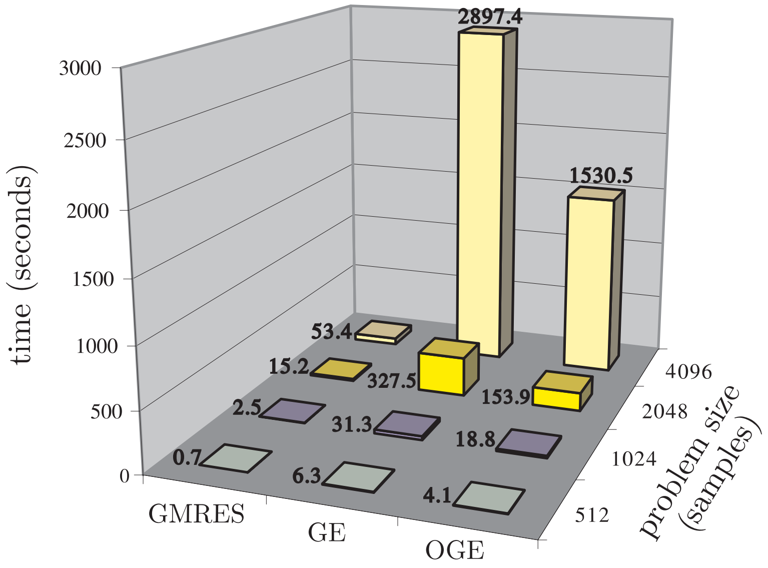


Figure 5.14: Speed comparison of GMRES, GE, and OGE

two defects placed at different distance between them. The result of the MED algorithm applied to the same measured signals is represented in Figures 5.15(a)-(c). By analyzing the signal given in Figure 5.15(a) we note that the estimated distance between the defects is equal to the real one[10], i.e. $L = L' = 1.5\lambda$ (compare it with Figure 5.1(b), (f)). In Figure 5.15(a) we can also see a number of spurious peaks. The amplitude of these peaks is much lower than the amplitude of peaks corresponding to defects. In practice, spurious peaks can be deleted by using the thresholding technique.

In two other cases (two defects at distanced $L = 2.5\lambda$ and $L = 6.5\lambda$), the MED algorithm also gives good results (compare Figures 5.15(b) with 5.1(c),(g) and Figures 5.15(c) with 5.1(d),(h)) Here the estimated distances between defects are also correct.

In the case of many defects the MED algorithm demonstrates great results. In Figure 5.15(d) all four equally spaced defects are perfectly separated. This result is better than than the result given in Figure 5.2(a),(c). In the case of three defects the MED is also able to separate defects (compare 5.2(b),(d) and 5.15(e)).

Let us compare the performance of the MED algorithm and the deconvolution approach. By comparing Figures 5.7(e),(f),(g),(h) and Figures 5.15(a),(b),(c) and (d), we conclude that the MED algorithm is not as sensitive to the noise as the deconvolution approach. Here we can see the MED algorithm reduces the effect of noise what significantly increases the resolution.

5.9.2 Detection of sharp defects with MED (2D case)

In the current section we introduce the result of the experiment which shows advantages of the Doppler imaging based on the MED algorithm over the amplitude Doppler imaging discussed in Chapter 3.

[10] we remind that the wavelength λ is about 12mm

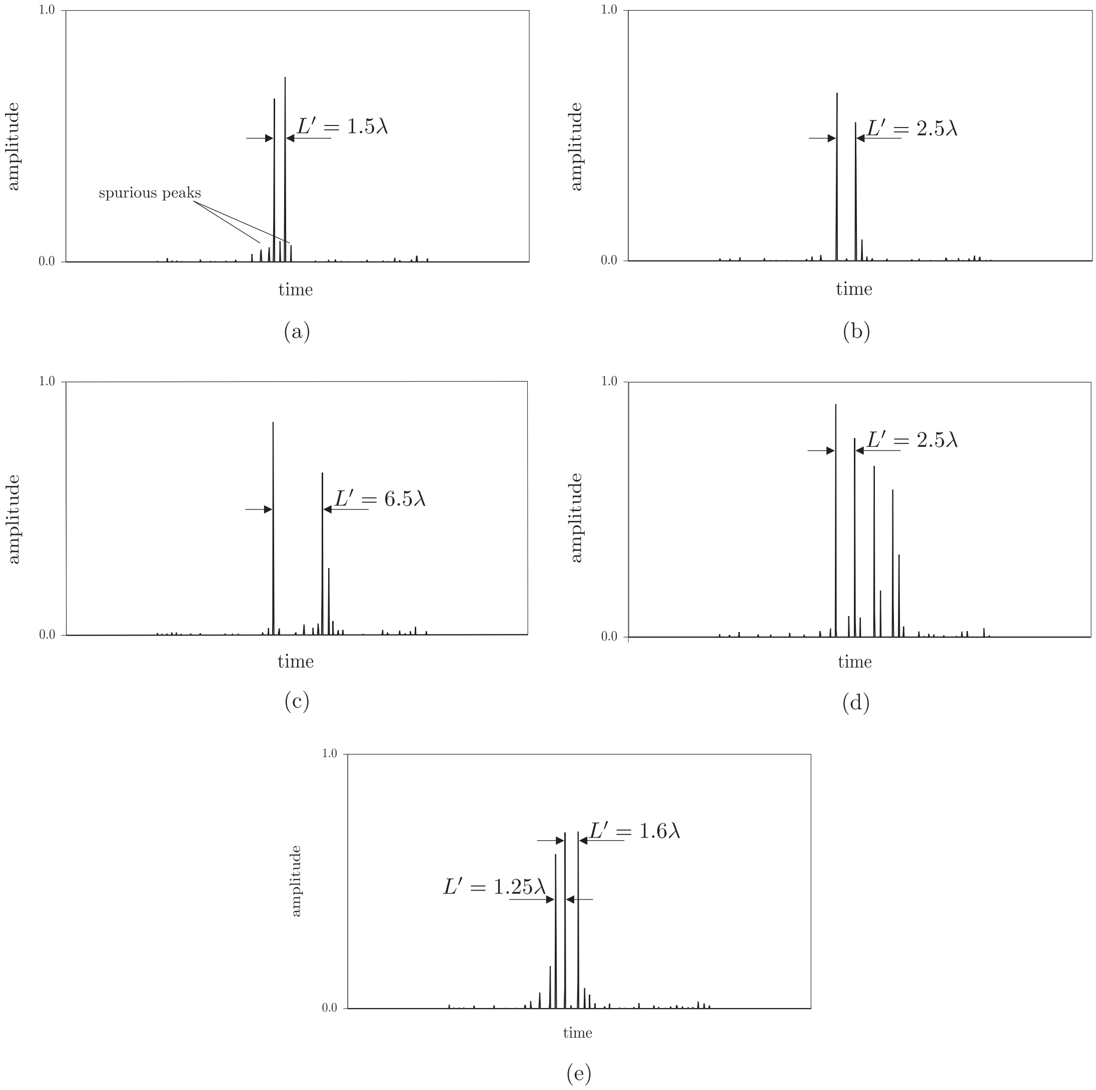


Figure 5.15: Maximum entropy deconvolution algorithm: (a) defect signal, two defects, $L = 1.5\lambda$; (b) defect signal, two defects, $L = 2.5\lambda$; (c) defect signal, two defects, $L = 6.5\lambda$ (d) defects signal, four equally spaced defects, $L = 2.5\lambda$ (e) defects signal, three defects spaced at $L = 1.25\lambda$ and $L = 1.6\lambda$

In order to perform the experiment we designed the specimen shown in Figure 5.16(a). This is a metal rectangular basement with artificial defects on its surface. Defects are made from a metal wire (thickness of 1mm) and have various geometrical shapes and sizes.

According to our considerations in Section 3.3 we perform multi-angle Doppler imaging from several angles of view. The measured surface scans are processed, rotated, and merged.

The amplitude Doppler image is represented in Figure 5.16(b). Here, we can see certain similarity with the specimen in Figure 5.16(a). Since the resolution ability of the Doppler amplitude approach is low, we can not precisely determine the form of defects. We only can determine the presence of defects and their locations. As we have already discussed before, for many practical applications this may be the only requirement.

The MED Doppler imaging introduces significant resolution improvement. In Figure 5.16(c) the defects are separated and perfectly recognizable. By comparing Figures 5.16(a) and 5.16(c) we also conclude that the locations of detected defects entirely coincide with the defects of the specimen. The squared shape of the detected circle in 5.16(c) can be explained through small number of angle scans. In practice, it is unlikely to perform large number of angle-scans and to proceed the acquired data in short periods of time.

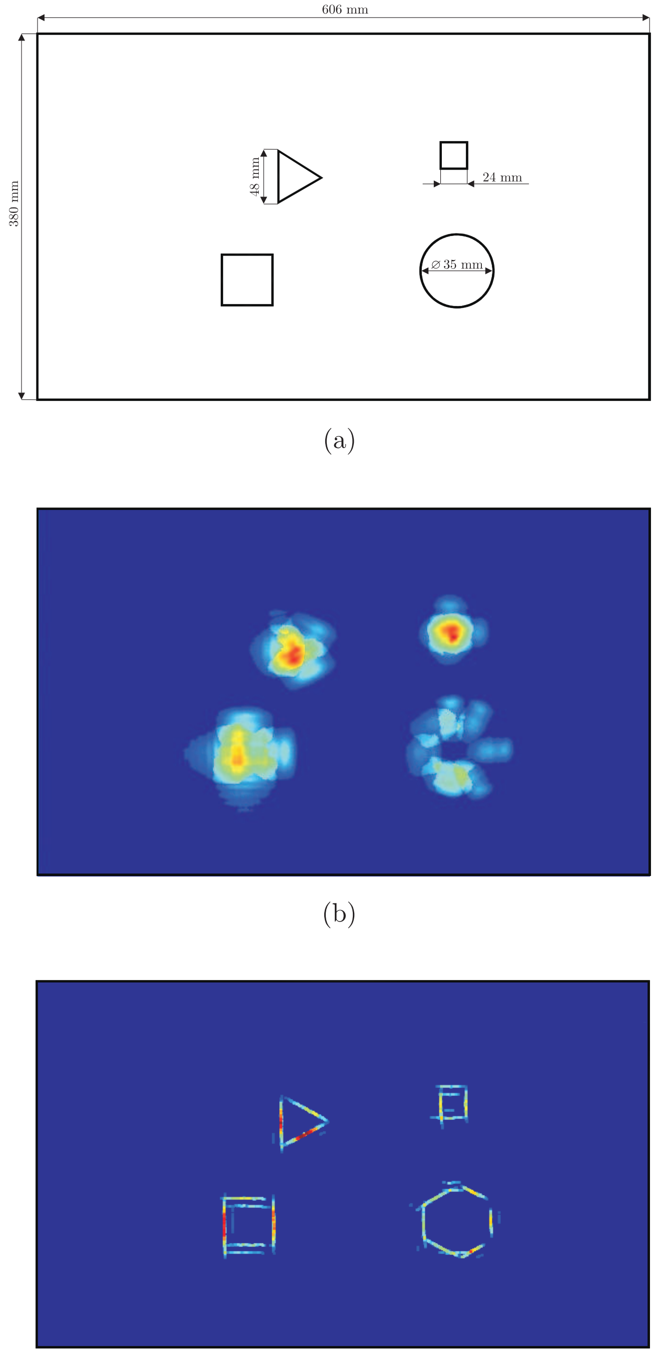
In both Figures 5.16(b) and 5.16(c) disturbances on the boundaries of the specimen are eliminated. In practice, the basement of the specimen can be chosen large enough to split Doppler effect on the boundaries from Doppler effect on defects.

From the experiment results represented in Figure 5.16 we conclude that the MED algorithm can be applied best on problems with sharp defects such as cracks, corrosion etc.

5.9.3 Detection of non-sharp defects with MED

Let us examine the ability of the MED algorithm to detect non-sharp defects. We perform 2D scan (from down to up) of a metal gun (in the following call specimen) which is shown in Figure 5.17(a). The amplitude Doppler image of the specimen is given in Figure 5.17(b). Analyzing this figure we conclude that by using of the amplitude image it is possible to detect presence of the specimen. However the shape of the specimen is not recognizable.

Since the specimen do not have pronounced sharp defects the measured signals are not suitable for the MED algorithm. As we have already discussed in Chapter 4 a typical behavior of the measured signal on the defect is the following. The Doppler amplitude changes from some low value to high, while the radar passes the defect, and then decreases again. The Doppler frequency steadily decreases. Such an amplitude and frequency variation creates an opportunity for MED algorithm to find entries in the analyzed signal which behave in a similar way (i.e. detect defects). The pattern which describes amplitude and frequency variation is the impulse response $\mathbf{h}$. Thus, non-sharp defects causes complex varying of the Doppler signal what leads the MED algorithm to fail. The MED Doppler image of the specimen is shown in Figure 5.17(c). As we can see the specimen is completely

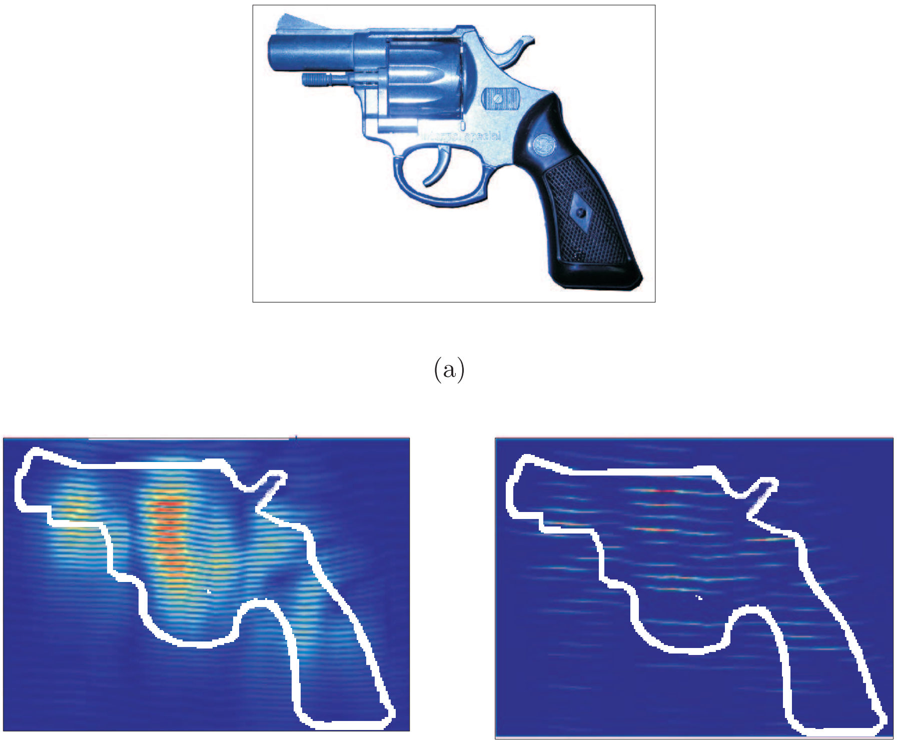


(a)

(b)

(c)

Figure 5.16: Sharp defects defection: (a) scheme of the specimen (b) amplitude Doppler image (c) MED Doppler image



(a)

(b)

(c)

Figure 5.17: Non-sharp defects detection: (a) picture of the specimen (gun) (b) amplitude Doppler image (c) MED Doppler image

unrecognizable. The multi-angle scan of the gun will not discover improvement of the MED algorithm.

5.10 Conclusion

In this chapter we discussed the Maximum Entropy Deconvolution (MED) algorithm as a means to increase spatial resolution of the CW radar. We have shown that the MED algorithm can be used for sharp defects detection and estimation of their exact spatial position. In order to increase speed of computation of MED we developed its fast version based on the GMRES iterative solver. As an alternative to the GMRES algorithm we have developed an optimized version of Gauss-Elimination algorithm which utilizes properties of the impulse response. That version of Gauss-Elimination algorithm can be implemented in hardware and used in industrial applications.

Chapter 6

Conclusion

The work described in this thesis introduces a prototype of the Doppler system for Non-Destructive Testing purposes.

In order to perform experiments a Doppler measurement System has been developed and built, see Chapter 2. It operated in different measurement modes such as raster and continuous measurements. The Doppler measurement system can also be easily adjusted for measurements with different designs of radars and antennas. The system ensures correct measurement triggering so that separately measured signals are spatially aligned. It provides an interface for programming of different motion patterns and for transporting the measured data to the PC. An advantage of the Doppler measurement system is its relatively low price because of the use of not-expensive CW-Radars.

An approach to find and estimate defects such as: material failure, irregularities, cracks, impact damages etc., is based on the Doppler amplitude and frequency analysis. In Chapter 3 the strength and weakness of the Doppler amplitude processing were discussed. In this chapter, for the first time, we introduce and formalize an 2D Multi-Angle Doppler imaging technique. This technique allows to reach better defect detection. The practical experiments, which confirm effectiveness of the 2D Multi-Angle Doppler imaging, are also presented.

Chapter 4 of the thesis is dedicated to the Doppler frequency analysis. We begin with mathematical model of the typical Doppler signal. This model helps us in double respect: to understand the nature of the Doppler signal and to choose the most suitable algorithm for Doppler frequency estimation. Here a number of modifications of standard frequency estimation techniques is proposed. Since the algorithm for Doppler frequency evaluation needs to be fast it is of high complexity. It is described in detail. At the end of the Chapter 4 the amplitude and frequency Doppler imaging techniques are compared.

In order to get higher spatial resolution of the Doppler radar a joint processing of the Doppler amplitude and frequency is proposed. We use the Maximum Entropy Deconvolution algorithm as a means to resolve closely spaced defects. Since the MED algorithm is computationally very complex we developed its fast version, which is based on an iterative solver. All the mathematical definitions and proofs are provided in Chapter 5. The speeding up of the MED algorithm is experimentally confirmed on real data. At the end of the Chapter we introduce some practical applications of the MED algorithm for processing of the Doppler signals.

The Doppler system is available in the Fraunhofer Institute for Non-Destructive Testing (IZFP), Saarbrücken. This laboratory system can be used in order to perform Non-Destructive Testing by means of the Doppler effect with microwaves.

Appendix A

Appendix

A.0.1 Phase Unwrapping

In the current section we introduce a phase unwrapping procedure (see Chapter 4). This procedure is applied to the instantaneous phase computed through equation (4.21) in order to remove discontinuities, see Figure A.1.

Definition A.0.1 *We define a **phase unwrapping** procedure*

$$\text{pUnwrap} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n.$$

Let $\phi' \in \mathbb{R}^n$ *be the an instantaneous phase computed through equation* (4.21). *Let us denote* $\phi \in \mathbb{R}^n$ *to be its unwrapped phase. For* $\phi = \text{pUnwrap}(\phi', pThd)$, $i \in [1 : n-1]$, $\phi_0 = \phi'_0$, $A(0) = 0$, *and a threshold value* $pThd \in \mathbb{R}$ *we define:*

$$\begin{aligned} \phi_i &= \phi'_i + A(i), \quad \text{where} \\ A(i) &= \begin{cases} \pi + A(i-1) & \text{if} \quad \mathbf{abs}(\phi'_i - \phi'_{i-1}) \geq pThd \text{ and} \\ A(i-1) & \text{otherwise} \end{cases} \end{aligned}$$

In the phase unwrapping procedure the value of $pThd$ is usually taken to be little bit smaller than π. It can be explained as follows. Because of digitalization errors an instantaneous phase ϕ' does not reach value $-\pi/2$ and $\pi/2$ at a discontinuity. A discontinuity is detected whenever a difference of values of two adjacent phase samples ϕ'_{i-1} and ϕ'_i exceeds threshold $pThd$, see Figure A.1. In such a case the value of ϕ_i is increased by π. If the next discontinuity appears, then ϕ_i is increased by π again. In this way an unwrapped phase ϕ becomes a signal without discontinuous (in our example it is a dashed line).

In the literature different algorithms for phase unwrapping were addressed. A widely used algorithm for phase unwrapping was suggested by Tribolet [117]. It was referred to as a very reliable algorithm in speed and seismic applications. Later, an improved version of this algorithm was developed [118]. A number of interesting algorithms for phase unwrapping can be also found in [119, 120].

A.0.2 Derivation of CW radar output, general case

In the current appendix we derive CW radar output signal, (see Chapter 1).

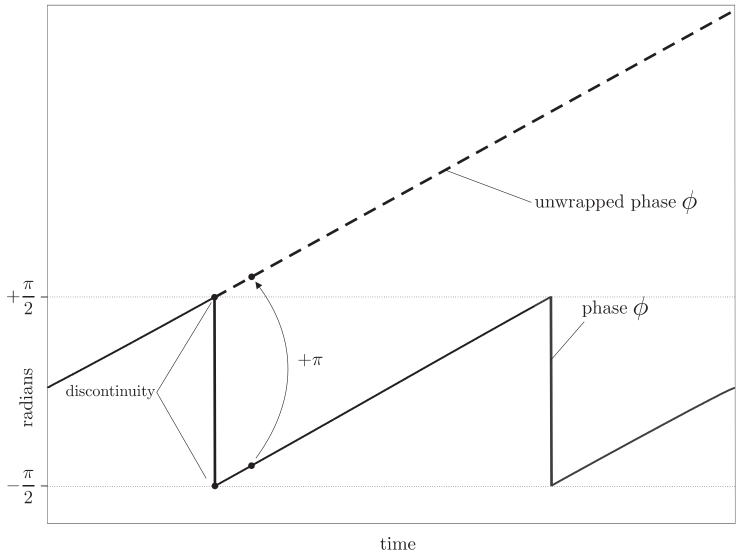


Figure A.1: Phase unwrapping procedure

Let us introduce some basic trigonometrical formulae that we need to derive CW radar output s^{out}.

$$\cos(\alpha)\,\cos(\beta) = \frac{1}{2}(\,\cos(\alpha+\beta) + \cos(\alpha-\beta)\,) \tag{A.1}$$

By substitution of incident signal s^{in} from equation (1.8) and scattered signal s^{sc} from equations (1.13) and (1.19) into (1.7) we have for all t:

$$\begin{aligned} s^{out}(t) &= \big(s^{in}(t) + s^{sc}(t)\big)^2 \\ &= \big(A^{in}\cos(2\pi f^{in}t + \varphi^{in}) + A^{sc}(t)\cos(2\pi f^{sc}(t) + \varphi^{sc})\big)^2 \\ &= \underbrace{\big(A^{in}\cos(2\pi f^{in}t + \varphi^{in})\big)^2}_{t_1} + \\ &\underbrace{\left(A^{sc}(t)\cos\left(2\pi f^{in}t \pm 2\pi\int\limits_0^t f^d(t)\,dt + \varphi^{sc}\right)\right)^2}_{t_2} + \\ &\underbrace{2\cdot A^{in}\cos\big(2\pi f^{in}t + \varphi^{in}\big)\,\cdot A^{sc}(t)\cos\left(2\pi f^{in}t \pm 2\pi\int\limits_0^t f^d(t)\,dt + \varphi^{sc}\right)}_{t_3}. \end{aligned} \tag{A.2}$$

In (A.2) terms t_1 and t_2 represent multiplication of cosines of the same argument. By using (A.1) we express t_1 and t_2 as follows:

$$t_1 = \frac{(A^{in})^2}{2}\,\big(\cos(4\pi f^{in}t + 2\varphi^{in}) + 1\big) \tag{A.3}$$

and

$$t_2 = \frac{(A^{sc}(t))^2}{2}\left(\cos\left(4\pi f^{in}t \pm 4\pi\int\limits_0^t f^d(t)\,dt + 2\varphi^{sc}\right) + 1\right) \tag{A.4}$$

A simplification of t_3 in (A.2) gives us

$$\begin{aligned} t_3 =& A(t)\cos\left(4\pi f^{in}t \pm 2\pi\int\limits_0^t f^d(t)\,dt + \varphi'\right) + \\ & A(t)\cos\left(\pm 2\pi\int\limits_0^t f^d(t)\,dt + \varphi\right), \end{aligned} \tag{A.5}$$

where

$$A(t) = A^{in}A^{sc}(t), \quad \varphi' = \varphi^{sc} + \varphi^{in}, \quad \text{and} \quad \varphi = \varphi^{sc} - \varphi^{in}.$$

We represent the acquired signal s^{out} as

$$s^{out}(t) = A(t)\cos\left(\pm 2\pi\int\limits_0^t f^d(t)\,dt + \varphi\right) + \Lambda, \tag{A.6}$$

where

$$\Lambda = t_1 + t_2 + t_3$$

A.0.3 Derivation of CW radar output, raster measurements

As we already discussed in Section 2.2 in raster measurements the Doppler frequency f^d is zero, i.e. for all t, $f^d(t) = 0$. In this case the definition of CW radar output s^{out} given in (2.1) have to be reconsidered.

Let us omit high-frequency terms in $\Lambda = t_1 + t_2 + t_3$ in Appendix A.0.2 because they can not measured. Thus, we modify equation (A.6) as

$$\begin{aligned} s^{out}(t) &= \frac{(A^{in})^2}{2} + \frac{(A^{sc}(t))^2}{2} + A(t)\cos(\varphi) \\ &= \frac{(A^{in})^2}{2} + \frac{(A^{sc}(t))^2}{2} + A(t)\cos\left(2\pi\frac{2f^{in}}{c_0}R_0\right). \end{aligned} \tag{A.7}$$

In the latter equation the only signal which depends on time is A^{sc}. In other hand the amplitude A^{sc} in equation (1.11) is given as some function of radar-target distance R, electrical properties of the target ϵ_t, amplitude A^{in}. Since in raster measurement R is time-independent (i.e. constant), then the signal s^{out} is also time-independent. However, as we can see in equation A.7, the signal s^{out} depends on initial radar-target distance R_0.

For simplicity reason we do not introduce dependence of amplitudes A^{in}, A^{sc}, and A on media properties, material properties, physics of microwave propagation etc. This information belongs to the field of theoretical physics and can be found in the literature. We define the output CW radar signal s^{out} in raster measurements as the following

$$s^{out}(R_0) = A(R_0)\cos\left(2\pi\frac{2f^{in}}{c_0}R_0\right) + \mathrm{C}(R_0), \tag{A.8}$$

where $\mathrm{C}(R_0) = A^{in} + A^{sc}(R_0)$.

The equation (A.8) represents a cosine oscillating with varying range. Its period is determined by double transmitted frequency $2f^{in}$. Correspondingly, we expect that every $\lambda^{in}/2$ a cosine changes its argument from 0 to 2π, i.e. passes the full period. The amplitude A fades (or rises) with increasing (or decreasing) distance according to an R^{-4}- law (for reference see Section (1.4.2)).

We note that equation (A.8) is only a model of the measured signal s^{out}. This model describes a real Doppler signal only partially. In Chapter 2 we check the viability of this equation experimentally.

A.0.4 Entropy function concavity

In the current appendix we present a proof that the entropy function Ψ defined in equation (5.20) in Chapter 5 is concave. The concavity property ensures convergence of the Newton algorithm so that it provides a correct optimizer. First, we prove that $-\Psi$ is convex, what lead us to the concavity of Ψ. The definition of the convexity and some auxiliary theorems are given below.

Definition A.0.2 *A **convex function** is a continuous function whose value at an interior of every interval in its domain does not exceed the arithmetic mean of its values at the ends of the interval* [121]*.*

In other words, a function f is convex on an interval $[a : b]$ if for any two points x_1 and x_2 in $[a : b]$ and any λ such that $0 \leq \lambda \leq 1$ *the following holds*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2).$$

According to [122, page 217] *the definition of convexity given in the latter equation can be reformulated for multidimensional real-valued function $f : \mathbb{R}^n \to \mathbb{R}$ as*

$$f(\lambda\mathbf{x} + \mu\mathbf{y}) \leq \lambda f(\mathbf{x}) + \mu f(\mathbf{y}), \tag{A.9}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\lambda, \mu \geq 0$ with $\lambda + \mu = 1$.

Theorem A.0.3 *Function f is **strictly convex** if its second derivative exists and is positive definite, i.e.* $\det(\nabla^2 f > 0)$*, for reference see* [122, page 230] *and* [100, page 110]

Theorem A.0.4 *Let f be a continuous function. The second derivative of f is positive definite if its eigenvalues are positive and real* [123, page 131]*.*

Theorem A.0.5 *If $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})$ are convex functions on set $\mathbb{C} \in \mathbb{R}^n$, then the function*

$$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + \ldots + f_n(\mathbf{x})$$

is also convex (for more information see [124, page 56]*)*

Let us define the auxiliary theorem which we will use in the main theorem.

Theorem A.0.6 *Let us prove that inequality*

$$(\lambda\, a + \mu\, b)^2 \;\leq\; \lambda\, a^2 \;+\; \mu\, b^2$$

holds for all $a, b \in \mathbb{R}$*, constants* λ*, and* μ *defined in equation* (A.9)*.*

Proof:

Using definition of μ *we have:*

$$(\lambda\, a + (1-\lambda)\, b)^2 \;\leq\; \lambda\, a^2 \;+\; (1-\lambda)\, b^2.$$

Applying elementary mathematical transformations to this inequality we derive:

$$\lambda^2 (a-b)^2 \leq \lambda (a-b)^2$$

From the latter inequality we conclude that the claim of the theorem holds $\blacksquare$*.*

Let us prove the concavity of the entropy function Ψ.

Theorem A.0.7 *First, we prove that the function*

$$f(\mathbf{x}) = S(\mathbf{x}, \mathbf{b}) + \alpha \cdot \chi(\mathbf{x}, \mathbf{y}, \mathbf{h}, \sigma) \tag{A.10}$$

is convex.

Proof:

We prove the theorem in two steps. At the first step we will show that the function S *is convex. In the second step we prove convexity of* χ*. After that we apply Theorem* A.0.5 *and by using first and second steps we show the goal of the theorem.*

(1) *The Hessian* $\nabla^2 \mathbf{S}$ *of the function* S *is defined as*

$$\nabla^2 \mathrm{S}_{i,j} = \frac{\delta(i-j)}{\mathrm{x}_i} = \begin{pmatrix} \frac{1}{\mathrm{x}_0} & 0 & \dots & 0 \\ 0 & \frac{1}{\mathrm{x}_1} & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\mathrm{x}_{n-1}} \end{pmatrix}, \tag{A.11}$$

where $i, j \in [0 : n-1]$*. The latter equation shows that the hessian* $\nabla^2 \mathbf{S}$ *is a diagonal matrix*[1]*. According to theorem given in* [125, page 281] *a so-called characteristic polynomial* $\Delta_{(\nabla^2 \mathbf{S})}(t)$ *of a diagonal matrix* $\nabla^2 S$ *is given as*

$$\Delta_{(\nabla^2 S)}(t) = (\lambda_0 - \nabla^2 \mathrm{S}_{0,0}) \cdot (\lambda_1 - \nabla^2 \mathrm{S}_{1,1}) \dots (\lambda_{n-1} - \nabla^2 \mathrm{S}_{n-1,n-1}).$$

According to [125, page 284] *the eigenvalues* $\lambda_1, \lambda_2, \dots, \lambda_{n-1} \in \mathbb{R}$ *of* $\nabla^2 \mathbf{S}$ *can be defined from the latter equation by letting* $\Delta_{(\nabla^2 S)}$ *be* 0 *as*

$$\lambda_0 = \frac{1}{\mathrm{x}_0},\; \lambda_1 = \frac{1}{\mathrm{x}_1},\; \dots\; , \lambda_{n-1} = \frac{1}{\mathrm{x}_{n-1}} \tag{A.12}$$

[1] i.e. all its non-diagonal entries are zeros

Using the fact that $\mathrm{x}_i > 0$ (*see definition of the function* S *in equation* (5.23)) *and equation* (A.12) *we have that all eigenvalues are positive and real.*

$$\lambda_0,\, \lambda_1 \ldots, \lambda_{n-1} > 0. \tag{A.13}$$

From equation (A.13) *and* Theorem A.0.3 *follows that the function* S *is convex.*

(2) *In the second step of the proof we show that the second part of equation* (A.10), *i.e. function* χ *is convex. According to* Definition A.0.2 *the function* χ *is convex if equation* A.9 *holds. After substitution of definition of function* χ (*from equation* (5.24)) *and applying mathematical transformations, the goal of the second step has the following form:*

$$\sum_{i=0}^{n-1}(\mathrm{y}'_i - (\mathbf{h} * (\lambda\mathbf{x_1} + (1-\lambda)\mathbf{x_2}))_i)^2 \leq$$
$$\sum_{i=0}^{n-1}\lambda(\mathrm{y}'_i - (\mathbf{h} * \mathbf{x_1})_i)^2 + (1-\lambda)(\mathrm{y}'_i - ((\mathbf{h} * \mathbf{x_2})_i)^2$$

From this we derive

$$\sum_{i=0}^{n-1} A_i \leq \sum_{i=0}^{n-1}(B_i + C_i),$$

where

$$A_i = (\lambda(\mathbf{h} * \mathbf{x_1})_i + (1-\lambda)(\mathbf{h} * \mathbf{x_2})_i)^2$$
$$B_i = \lambda(\mathbf{h} * \mathbf{x_1})^2$$
$$C_i = (1-\lambda)((\mathbf{h} * \mathbf{x_2})_i)^2$$

Substituting in Theorem A.0.6 a *and* b *instead of* $(\mathbf{h}*\mathbf{x_1})_i$ *and* $(\mathbf{h}*\mathbf{x_2})_i$ *we conclude that for all* $i \in [0 : n-1]$ *hold that* $A_i \leq B_i + C_i$. *The latter relation implies that the claim of the second step of the theorem holds, i.e. the function* χ *is convex.*

Combining claims 1 *and* 2 *and using* Theorem A.0.5 *we conclude that function* f, *which is a sum of two convex functions* S *and* χ, *is also convex* ■.

By the definition the entropy function $\Psi = -f$, then according to [126, page 1132] the function Ψ is concave.

Appendix B

Appendix

B.0.5 Practical GMRES algorithm implementation

```
#define eps 2.2204e-016

Matrix& unit_vec(int k, int n);
void planerot(Matrix &x_, Matrix &G, Matrix &x);
void Orthogonalization(Matrix &v, Matrix &U, int k);
void SetJtempToEye(Matrix &Jtemp, int initer);
Matrix& Jtemp_x_J(Matrix &Jtemp, Matrix &J, int initer);
Matrix& J_x_v(Matrix &J, Matrix &v, int initer);

// 1----------------------------------------------------------------------
// M1 = U, M2 = L, where M1 and M2 are left and right  preconditioners, respectively

Matrix& GMRES(Matrix &A, Matrix &b, int restart, real tol, int maxit, Matrix &M2L, Matrix &M1U)
{
        Matrix *x = new Matrix("x"); // initial guess (solution) x0, is zero-vector
        Matrix w("w");
        Matrix J = Matrix::Eye("J", 1);
        int flag;
        real relres = 0;
        int m, n;
        m = A.Height();
        n = A.Width();
        Matrix Jtemp = Matrix::Eye("Jtemp", n, n);
        int restarted = 1;
        int outer, inner;
        if (restarted) { outer = maxit; inner = restart; }
        real n2b = b.Norm();
        if (n2b == 0)
        {
                *x = Matrix::Zeros("x", n, 1);   // if rhs is zero-vector then solution is also zeroes-vector
                flag = 0;                        // a valid solution has been computed
                relres = 0;                      // no iterations to be done
                return *x;
        }

//2.............................................................................
        int existM1U = 1;
        int existM2L = 1;
        Matrix x0("x0", 1, b.Height());
        *x = x0;

//3.............................................................................
        // set up for GMRES
        flag = 1;
        Matrix xmin("xmin");
        xmin = *x;                      // current solution
        int imin = 0;                   // number of outer iteration
        int jmin = 0;                   // number of inner iteration
```

```
    real tolb = tol * n2b;              // define tolerance

    Matrix r = b - A*(*x);              // compute the residual vector and its norm
    real normr = r.Norm();

    saveout(&r, "r", 11, 0);

    if (normr <= tolb){
        flag = 0;
        return *x;
    }

    Matrix r1 = Matrix::SolveTriUpCase(M1U, r);

    r = r1;

//4........................................................................
    normr = r.Norm();
    Matrix resvec = Matrix::Zeros("resvec", inner*outer+1,1);         // preallocate vector for norm of residuals
    resvec[0][0] = normr;                                             // resvec(1) = norm(b-A*x0)
    real normrmin = normr;                                            // norm of residual from xmin
    int rho = 1;
    int stag = 0;                                                     // stagnation of the method

//5........................................................................
    int initer;
    for (int outiter=0; outiter<outer; outiter++)
    {
        initer = 0;

        Matrix u = r + Matrix::Sign(r[0][0])*normr*unit_vec(0,n);
        u = u / u.Norm();

        Matrix U = Matrix::Zeros("U", n,inner);
        Matrix Psol = Matrix::Zeros("Psol", n,inner);
        Matrix R = Matrix::Zeros("R", n,inner);
        U.Set(0, 0, -1, -1, u);

        w = r; Orthogonalization(w, u, 0);

        for (initer=0; initer<inner; initer++)
        {
            cout << "#outer" << outiter << " #inner" << initer << " ";

            Matrix v = unit_vec(initer,n) - 2*u*(u[initer][0]);
            for(int k=initer-1; k>=0; k--)
                Orthogonalization(v, U, k);

            // give P1*P2*P3...Pm*em
            Psol.Set(initer, initer, -1, -1, v);


            v = A*v;

            v = Matrix::SolveTriUpCase(M1U, v);


            for(k=0; k<initer+1; k++)
                Orthogonalization(v, U, k);

            if (!(initer==v.Length() || v.IsZeros(initer, -1))) {
                u.SetToZeros();
                real alpha = (0-Matrix::Sign(v.Get(initer+1))) * v.Norm(initer+1, -1);
                u.Set(initer+1, -1, v.Get(initer+1, -1).SwapWH() - alpha*unit_vec(0,n-initer-1));
                u = u / u.Norm();

                U.Set(initer+1, initer+1, -1, -1, u);
                // apply Pm+1 to v
                Orthogonalization(v, u, 0);
            }
```

```
//6...............................................................................
            if (initer==0)
                J = Matrix::Eye("J", n, n); //J = SparseMatrix::Eye("J", n, n);
            else{
                J = Jtemp_x_J(Jtemp, J, initer);
                v.isTemporal = true;
                v = J_x_v(J, v, initer);
            }

//7...............................................................................
            // find Given's rotation Jm
            real a1_ = v.Get(initer+1);
            if (!( initer==v.Length() ))
                if (v.Get(initer+1)==0.0)
                    SetJtempToEye(Jtemp, initer);
                else{
                    SetJtempToEye(Jtemp, initer);
                    Matrix Jhat("Jhat"), v_("v_");
                    planerot( v.Get(initer, initer+1), Jhat, v_);
                    v.Set(initer, initer+1, v_);
                    Jtemp.Set(initer, initer+1, initer, initer+1, Jhat);
                    w.Set(initer, initer+1, Jhat*w.Get(initer, initer+1).Transp());
                }
            R.Set(initer, initer, -1, -1, v);

//8...............................................................................
            if (initer<inner){
                normr = fabs(w[initer+1][0]);
                resvec.Buff()[outiter*inner+initer+1] = normr;
            }

//9...............................................................................
            if (normr <= normrmin){
                normrmin = normr;
                imin = outiter;
                jmin = initer;
            }
//10..............................................................................
            if (normr < tolb){
                flag = 0;
                break;
            }
        }// ends innerloop

//11..............................................................................
        Matrix y = Matrix::SolveTriUpCase(R.Get(0, jmin, 0, jmin), w.Get(0, jmin).Transp());
        Matrix y_("y", 1, R.Width());
        y_.Set(0, y.Length()-1, y);
        Matrix additive = Psol*y_;
        *x = *x + additive;
        xmin = *x;

// 12.............................................................................
        r = b - A * (*x);
        r1 = Matrix::SolveTriUpCase(M1U, r);
        r = r1;
        normr = r.Norm();

        resvec.Buff()[outiter*inner+initer+1] = normr;

        if (normr <= normrmin){
            xmin = *x;
            normrmin = normr;
        }
// 13.............................................................................
        // test for stagnation on outer iterate
        if (flag!=2){
            Matrix stagtest = Matrix::Zeros("stagtest", n,1);
            Matrix ind = (*x)!=0.0f;
            for (int ii=0; ii<ind.Length(); ii++)
                if ((int)ind.Buff()[ii] == 1)
                    stagtest.Buff()[ii] = additive.Buff()[ii] / x->Buff()[ii];
```

```
                        if ( additive.NormInf() < eps ){
                                stag = 1;
                                flag = 3;
                                break;
                                // no changes in outer iterate
                        }
                }

//14................................................................................
                if (normr < tolb){
                        flag = 0;
                        break;
                }
        }// ends outer loop
//15................................................................................
        // returned solution is that with minimum residual
        if (flag == 0)
                relres = normr / n2b;
        else{
                (*x) = xmin;
                relres = normrmin / n2b;
        }

//16................................................................................
        // truncate the zeros from resvec
        if (flag <= 1 || flag == 3){
                resvec = resvec.Get(0, outiter*inner+initer+1);
                Matrix indices = resvec==0.0f;
                resvec = resvec.LogicalMap(!indices);
        }else{
                if (initer == 0)
                                resvec = resvec.Get(0, outiter*inner+1);
                else
                                resvec = resvec.Get(0, outiter*inner+initer);
        }

        // release memory
        Jtemp.~Matrix();
        J.~Matrix();
        w.~Matrix();
        x0.~Matrix();
        resvec.~Matrix();
        xmin.~Matrix();
        r.~Matrix();
        r1.~Matrix();
        if (A.isTemporal)
                A.~Matrix();
        if (b.isTemporal)
                b.~Matrix();
        if (M2L.isTemporal)
                M2L.~Matrix();
        if (M1U.isTemporal)
                M1U.~Matrix();
        //set ‘x’ to be temporal vector to preserve memory in a future
        x->isTemporal = true;
        return *x;
}

//------------------------------------------------------------------------

Matrix& unit_vec(int k, int n)
{
        Matrix *vec = new Matrix("*vec");
        *vec = Matrix::Zeros("unit_vec", n,1);
        (*vec)[k][0] = 1;
        //set ‘vec’ to be temporal vector to preserve memory in a future
        vec->isTemporal = true;
        return *vec;
}

//------------------------------------------------------------------------

// [G,y] = planerot(x) where x is a 2-component column
// vector, returns a 2-by-2 orthogonal matrix G so that y = G*x has y(2) = 0
void planerot(Matrix &x_, Matrix &G, Matrix &x)
{
        if (x_.Get(1) != 0)
        {
                bool toCall = x_.isTemporal;
                if (x_.isTemporal)
                        x_.isTemporal = false;

                real r = x_.Norm();
                G = Matrix("G", 2, 2);
                G.Set(0, 1, 0, 0, x_);
                G[1][0] = -x_.Get(1);
                G[1][1] = x_.Get(0);
                G = G / r;
                x = Matrix("x", 2, 1);
                x[0][0] = r;
                x[0][1] = 0;

        if (toCall)
                x_.~Matrix();
        }else{
                throw MyException("The case is not handled!");
        }
}

//--------------------------------------------------------------------------

// orthogonalization of vector 'v'
void Orthogonalization(Matrix &v, Matrix &U, int k)
{
        int l = v.Length();
        int Uw = U.Width();

        real *UBuff = U.Buff();
        real *vBuff = v.Buff();

        int ixUw = k;
        real t = 0;
        for (int i=0; i<l; i++)
        {
                t += UBuff[ixUw]*vBuff[i];
                ixUw += Uw;
        }

        ixUw = k;
        for (i=0; i<l; i++)
        {
                vBuff[i] = vBuff[i] - (t+t)*UBuff[ixUw];
                ixUw += Uw;
        }
}

//--------------------------------------------------------------------------

// reset matrix Jtemp to eye matrix
void SetJtempToEye(Matrix &Jtemp, int initer)
{
        if (initer > 0)
        {
                Jtemp[initer-1][initer-1] = 1;
                Jtemp[initer][initer] = 1;
                Jtemp[initer-1][initer] = 0;
                Jtemp[initer][initer-1] = 0;
        }
}

//--------------------------------------------------------------------------

// multiplication of matrices Jtemp and J as sparse matrices, but using simples
// algorithm, which is adopted for special case
Matrix& Jtemp_x_J(Matrix &Jtemp, Matrix &J, int initer)
{
        if (Jtemp.Width() != J.Height())
                { throw MyException("Dimentions is not coresponds!"); }

        int w, h;
        w = J.Width();
        h = Jtemp.Height();
        Matrix *res = new Matrix("res", w, h);

        real *resBuff = res->Buff();
        real *JtempBuff  = Jtemp.Buff();
        real *JBuff = J.Buff();

        int wJtemp = Jtemp.Width();
        int wJ = J.Width();

        // make eye matrix
        for (int i=0; i<w; i++)
                resBuff[i+i*w] = 1;
        // copy unchangeable part
        for (i=0; i<initer-1; i++)
                for (int j=0; j<initer; j++)
                        resBuff[j+i*w] = JBuff[j+i*wJ];

        // range of submatrix a
        int ax0_ = initer-1;
        int ax1_ = initer;
        int ay0_ = ax0_;
        int ay1_ = ax1_;

        // range of submatrix b
        int bx0_ = 0;
        int bx1_ = ax1_;
        int by0_ = ay0_;
        int by1_ = ay1_;

        real c;
        for (int j = by0_; j<by1_+1; j++)
                for (int i = bx0_; i<bx1_+1; i++)
                {
                        c = 0;
                        for (int k = by0_; k<by1_+1; k++)
                        {
                                real a = JtempBuff[k+j*wJtemp];
                                real b = JBuff[i+k*wJ];
                                c += a*b;
                        }
                        resBuff[i+j*w] = c;
                }
        // set ‘res’ to be temporal vector to preserve memory in a future
        res->isTemporal  = true;
        return *res;
}
//----------------------------------------------------------------------

// multiplication of matrices J and v as sparse matrices,
Matrix& J_x_v(Matrix &J, Matrix &v, int initer)
{
        if (J.Width() != v.Height())
                { throw MyException("Dimentions is not coresponds!"); }

        Matrix *res = new Matrix("res", 1, v.Height());

        real *JBuff = J.Buff();
        real *vBuff = v.Buff();
        real *resBuff = res->Buff();

        int wJ = J.Width();

        // copy unchangeable part
        int hv = v.Height();
        for (int i=initer+1; i<hv; i++)
                resBuff[i] = vBuff[i];
        real c;
        for (i=0; i<initer+1; i++)
        {
                c = 0;
                for (int j=0; j<initer+1; j++)
                {
                        real a = JBuff[j+i*wJ];
                        real b = vBuff[j];
                        c += a*b;
                }
                resBuff[i] = c;
        }

        // elease memory
        if (J.isTemporal)
                J.~Matrix();
        if (v.isTemporal)
                v.~Matrix();
        / set ‘res’ to be temporal vector to preserve memory in a future
        res->isTemporal = true;
        return *res;
}
```

B.0.6 Practical OGE algorithm implementation

```
Matrix& OptGAUSSMethod(Matrix &A, Matrix &b, int m)
{
        // initialization
        Matrix *x = new Matrix("x", 1, A.Width());
        int N = A.Width();
        Matrix A_("A_"), b_("b_");
        A_ = A;
        b_ = b;
        int w, h;
        w = A.Width();
        h = A.Height();
        int n = w = h;
        real c;

        //  Gauss-Elimination of matrix A (first step)
        for (int s=1; s<N-(m-1)-m+1; s++)
        {
                for (int i=1; i<m-1+1; i++)
                {
                        c = A_[i+s-1][s-1]/A_[s-1][s-1];
                        for (int j=1;  j<m+1; j++)
                                A_[i+s-1][j+s-1-1] = A_[i+s-1][j+s-1-1] - A_[s-1][j+s-1-1]*c;
                        for (j=1; j<m-1+1; j++)
                                A_[i+s-1][N-j+1-1] = A_[i+s-1][N-j+1-1] - A_[s-1][N-j+1-1]*c;
                        b_[i+s-1][0] = b_[i+s-1][0] - b_[s-1][0]*c;
                }

                for (i=1; i<m-1+1; i++)
                {
                        c = A_[i+1+N-m-1][s-1]/A_[s-1][s-1];
                        for (int j=1;  j<m+1; j++)
                                A_[i+1+N-m-1][j+s-1-1] = A_[i+1+N-m-1][j+s-1-1] - A_[s-1][j+s-1-1]*c;
                        for (j=1; j<m-1+1; j++)
                                A_[i+1+N-m-1][N-j+1-1] = A_[i+1+N-m-1][N-j+1-1] - A_[s-1][N-j+1-1]*c;
                        b_[i+1+N-m-1][0] = b_[i+1+N-m-1][0] - b_[s-1][0]*c;
                }
        }

        // Gauss-Elimination of matrix A (second step)
        for (s=N-2*m+2; s<N+1; s++)
        {
                for (int i=1; i<N-s+1; i++)
                {
                        c = A_[i+s-1][s-1]/A_[s-1][s-1];
                        for (int j=1; j<N-s+1+1; j++)
                                A_[i+s-1][j+s-1-1] = A_[i+s-1][j+s-1-1] - A_[s-1][j+s-1-1]*c;
                        b_[i+s-1][0] = b_[i+s-1][0] - b_[s-1][0]*c;
                }
        }

        // solution of triangular system of linear equation
        real sm;
        for (s = N; s>=1; s--)
        {
                sm = 0;
                for (int i = N; i>=s; i--)
                        sm = sm + A_[s-1][i-1]*(*x)[i-1][0];
                 (*x)[s-1][0] = (b_[s-1][0]-sm)/A_[s-1][s-1];
        }

        // release memory
        if (A.isTemporal)
                A.~Matrix();
        if (b.isTemporal)
                b.~Matrix();
        A_.~Matrix();
        b_.~Matrix();

        return *x;
}
```

Index

Bibliography

[1] Kurt Fenske and Devendra Misra. Dielectric materials at microwave frequencies. *Applied Microwave & Wireless*, 12(11):92–100, 2000.

[2] Alfred J. Bahr. Experimental techniques in microwave ndt. *Review of Progress in Quantitative Non-destructive Evaluation*, 14:593–600, 1995.

[3] Patric O. Moore. *Nondestructive Testing Handbook*, volume 5. American Society for Nondestructive Testing, 2004.

[4] Reza Zoughi. *Microwave Non-Destructive Testing and Evaluation*, volume 4. Kluwer Academic Publishers, 2000.

[5] N. Ida. *Microwave NDT*, volume 10. Kluwer Academic Publishers, 1992.

[6] A. F. Harvey. *Microwave Engineering.* Academic Press, 1963.

[7] Ebbe Nyfors & Pertti Vainikainen. *Industrial Microwave Sensors.* Artech House, Inc., 1989.

[8] Mike Gilmore Barry Elliott. *Fiber Optic Cabling.* Newnes, 2 edition, 2002.

[9] Raymond L. Camisa Paul R. Karmel Gabriel D. Colef. *Introduction to Electromagnetic and Microwave Engineering.* John Wiley & Sons, Inc., 1998.

[10] Naval Air Systems Command. Ew and radar systems, engineering handbook. http://ewhdbks.mugu.navy.mil/contents.htm, 2000.

[11] Smolskiy S.M. Komarov I.V. *Fundamentals of Short-Range FM Radar.* Artech House, 2003.

[12] Torgovnikov G.I. *Dielectric Properties of Wood and Wood-Based Materials.* Springer-Verlag, 1993.

[13] Orfanidis J.S. Electromagnetic waves and antennas. http://www.ece.rutgers.edu/ orfanidi/ewa/.

[14] www.thefreedictionary.com/radar.

[15] Merill I. Skolnik. *Radar Handbook.* McGraw-Hill Book Company, 1970.

[16] Boualem Boashash. Estimation and interpreting the instantaneous frequency of a signal - part 1: Fundamentals. *Proceeding of the IEEE*, 80(4):520–538, 1992.

[17] Constantine A. Balanis. *Antenna Theory, Analysis and Design.* Wiley-Interscience, third edition edition, 2005.

[18] Arfken G. *Mathematical Methods for Physicists.* Florida Academic Press, 1985.

[19] Donald R. Wehner. *High-Resolution Radar.* Artech House, INC., second edition edition, 1995.

[20] Albrecht Ludloff. *Praxiswissen Radar und Radarsignalverarbeitung.* VIEWEG, 2. auflage edition, 1998.

[21] Sklarczyk C. Netzelmann U. Kreier P. Gebhardt W. Nondestructive and contactless determination of layer and coating thickness. In Gyekenyesi S., editor, *Nondestructive Evaluation and Health Monitoring of Aerospace Materials, Composites and Civil Infrastructure*, volume 5767. Proc. of SPIE, 2005.

[22] Otto J. Mikrowellen messen fuellstaende. *Sensortechnik*, 10, 1997.

[23] Otto J. 27th european microwave conference (eumc97). In *Radar applications in level measurements, distance measurements and nondestructive material testing*, volume 2, 1997.

[24] Tschuncky R. Entwicklung eines mustererkennungs- und klassifikationsmoduls fuer die indirekte charakterisierung von werkstoffeigenschaften. Master's thesis, Universitaet des Saarlandes, 2004.

[25] Heike Laqua aus Mayen. *Beruehrungslose Geschwindigkeitsmessung von Strassen- und Schienenfahrzeugen mit Mikrowellensensoren.* PhD thesis, Fakultat fur Maschinenbau, Universitat Fridericiana Karlsruhe, 1995.

[26] David Brandwood. *Fourier Transform in Radar and Signal Processing.* Artech House, Inc., 2003.

[27] Alfred Mertins. *Signal Analysis.* JOHN WILEY & SONS, 1999.

[28] S. Bochner and K. Chandrasekharan. *Fourier Transforms.* Princeton University Press, 1949.

[29] D. Sundararajan. *The Discrete Fourier Transform.* World Scientific, 2001.

[30] Duncan W. Mills Ronald L. Allen. *Signal Analysis, Time, Frequency, Scale and Structure.* IEEE Press, 2004.

[31] E. Bedrosian. A product theorem for hilbert transforms. *Proceedings of the IEEE*, 51:686–689, 1963.

[32] Albert. H. Nutall. On the quadrature approximation to the hilbert transform of modulated signal. *Proceedings of the IEEE*, 54:1458–1459, 1966.

[33] Jaehne Bernd. *Digital Signal Processing.* Springer, 5 edition, 2002.

[34] L.J.Van Vliet I.T.Young, J.J. Gerbrands. *Fundamentals of Image Processing.* Printed in The Netherlands at the Delft University of Technology, 1998.

[35] William K. Pratt. *Digital Image Processing.* John Wiley & Sons, Inc., 2001.

[36] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing.* Prentice-Hall, Inc., 2001.

[37] K. J. Falconer H. T. Croft and R. K. Guy. *Unsolved Problems in Geometry.* Springer-Verlag, 1991.

[38] National Instruments Corporation. *LabVIEW^{TM}and LabWindows/CVI^{TM}, Signal Processing Toolset*, December 2002.

[39] Antonia Papandreou-Suppappola. *Applications in Time Frequency Signal Processing.* CRC Press, 2003.

[40] M. Joppich. *Schaetzverfahren zur Genauigkeitsmessung der Geschwindigkeitsmessung ueber Grund nach dem Dopplerprinzip.* Number 440 in 8. VDI VERLAG GmbH, 1994.

[41] J. Carson and T. Fry. Variable frequency electronic circuit theory with application the theory of frequency modulation. *Bell System Tech*, 16:513–540, 1937.

[42] B. Van der Pol. The fundamental principles of frequency modulation. *Proc. IEE*, 93(3):153–158, 1946.

[43] D. Gabor. Theory of communications. *Proc. IEE*, 93(3):429–457, 1946.

[44] M. Guitton D. Menard J. Crestel, B. Emile. A doppler frequency estimate using the instantaneous frequency. In *13th International Conference on Digital Signal Processing*, volume 2, pages 777–780, 1997.

[45] Petar M. Djuric and Steven M. Kay. Parameter estimation of chirp signals. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 38(12), 1990.

[46] James R. Zeidler Paul C. Wei and Walter H. Ku. Adaptive recovery of a chirped signal using the rls algorithm. *IEEE Transactions on Signal Processing*, 45(2):3085–3101, 1997.

[47] Young Bok Ahn and Sing Bai Park. Estimation of mean frequency and variance of ultrasonic doppler signal by using second-order autoregressive model. *IEEE Transaction on Ultrasonics, Ferroelectrics and Frequency Control*, 38(3):172–182, 1991.

[48] Mingui Sun and Robert J. Sclabassi. Discrete-time instantaneous frequency and its computation. *IEEE Transactions On Signal Processing*, 41(5):1867–1880, 1993.

[49] Peter J. Fish Yuanyuan Wang. Arterial doppler signal simulation by time domain processing. *European Journal of Ultrasound*, 3:71–81, 1996.

[50] Vyacheslav P. Tuzlukov. *Signal Processing Noise (Electrical Engineering and Applied Signal Processing Series).* CRC Press LLC, 2002.

[51] Peter J. Fish. Non-stationarity broadening in pulsed doppler spectrum measurement. *Ultrasound in Medicine* & *Biology*, 17:147–155, 1991.

[52] Boualem Boashash. Estimation and interpreting the instantaneous frequency of a signal - part 2: Algorithms and applications. *Proceeding of the IEEE*, 80(4):540–568, 1992.

[53] R Vijayarajeswaran K Padmanabhan, S Ananthi. *A Practical Approach to Signal Processing.* New Age Publishers, 2001.

[54] Guan-Chyun and James C. Hung. Phase-locked loop techniques - a survey. *IEEE Transactions On Industrial Electronics*, 43(6):609–615, 1996.

[55] Someshwar C. Gupta. Phase-locked loops. *Proceedings of The IEEE*, 63(2):291–306, 1975.

[56] Lloyd G. Griffiths. Rapid measurement of digital instantaneous frequency. *IEEE Transactions On Acoustics, Speech and Signal Processing*, 23(2):207–222, 1975.

[57] Simon Haykin. *Adaptive Filter Theory.* Prentice Hall, 4th edition, 2002.

[58] Steven M. Kay. *Modern Spectral Estimation: Theory and Application.* Prentice Hall, 1988.

[59] P. O'Shea B. Boashash and M. Arnold. Algorithms for instantaneous frequency estimation: A comparative study. In *Proc. SPIE Conf. Advanced Algorithms and Architectures for Signal Processing*, volume 1384, 1990.

[60] David C. Rife and Robert R. Boorstyn. Single-tone estimation from discrete-time observations. *IEEE Transactions on Information Theory*, IT-20(5):591–598, 1974.

[61] Boualem Boashash. *Time Frequency Signal Analysis and Processing.* Elsevier, 1-th edition, 2003.

[62] Paulo G. Olivier L. Franois A., Patrick F. Time-frequency toolbox for use with matlab. http://tftb.nongnu.org/.

[63] Robert C. Williamson Brian C. Lovell and Boualem Boashash. The relationship between instantaneous frequency and time-frequency representation. *IEEE Transactions On Signal Processing*, 41(3):1458–1461, 1993.

[64] Boualem Boashash and Peter O'Shea. Use of the cross wigner-ville distribution for estimation of instantaneous frequency. *IEEE Transactions on Signal Processing*, 41(3), 1993.

[65] Paulo Goncalves Olivier Lemoine Franois Auger, Patrick Flandrin. *Time-Frequency Toolbox for Use with MATLAB.* Rice University (USA), CNRS (France), 1996.

[66] F. Nave Antonio C.A. Figueiredo, Maria Filomena and EFDA-JET Contributors. Improved time-frequency visualization of chirping mode signals in tokamak plasmas using the choi-williams distribution. *IEEE Transactions On Plasma Science*, 33(2):468–469, 2005.

[67] Bilin Zhang and Shunsuke Sato. A time-frequency distribution of cohen's class with a compound kernel and its application to speech signal processing. *IEEE Transaction On Signal Processing*, 42(1):54–64, 1994.

[68] Shimon Peleg & Boaz Porat. The sixteenth conference of electrical and electronics engineers in israel. In *Estimation and Classification of Signals with Polynomial Phase*, pages 1–4, March, 7-9 1989.

[69] Shimon Peleg & Boaz Porat. Estimation and classification of polynomial-phase signals. *IEEE Transactions on Information Theory*, 37(2):422–430, 1991.

[70] Shimon Peleg. *Estimation and Detection with Discrete Polynomial Transform.* PhD thesis, University of California, 1993.

[71] Boaz Porat & Benjamin Friedlander Shimon Peleg. The achievable accuracy in estimating the instantaneous phase and frequency of a constant amplitude signal. *IEEE Transactions on Signal Processing*, 41(6):2216–2224, 1993.

[72] Jonas Gomes. *Image Processing for Computer Graphics.* Springer, 1997.

[73] Boualem Boashash & Peter J. Black. An efficient real-time implementation of the wigner-ville distribution. *IEEE Transactions on Acoustic, Speech and Signal Processing*, ASSP-35(11):1611–1618, 1987.

[74] Shing chow Chan and Ka leung Ho. Efficient computation of the discrete wigner-ville distribution. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 38(12):2165–2168, 1990.

[75] M. Graca Ruano Jose C. Cardoso, Peter J. Fish. Proceedings of the 20th annual international conference of the ieee engineering in medicine and biology society. In *Parallel Implementation of A Choi-Williams TDF for Doppler Signal Analysis*, volume 20 of *3*, pages 1490–1492, 1998.

[76] Daniel T. Barry. Fast calculations of the choi-williams time-frequency distribution. *IEEE Transaction On Signal Processing*, 40(2):450–455, 1992.

[77] Cedric Richard and Regis Lengelle. 3rd international conference on signal processing. In *Fast Implementation of Time-Frequency Representations Modified by the Reassignment Method*, volume 1, pages 343–346, 1996.

[78] Richard G. Lyons. *Understanding Digital Signal Processing.* Prentice Hall PTR, 2001.

[79] Ronald W. Schafer Oppenheim A.V. *Digital Signal Processing.* Prentice-Hall, Inc., 1975.

[80] Franz Winkler. *Polynomial Algorithms in Computer Algebra.* Springer, 1996.

[81] Lathi B.P. *Linear Systems and Signals.* Oxford University Press, 1992.

[82] Ahmed I. Zayed Ahmed F. Ghaleb Mourad E. H. Ismail, M. Zuhair Nashedm, editor. *Mathematical Analysis, Wavelets, and Signal Processing.* American Mathematical Society, 1995.

[83] Tuzlukov V. P. *Signal Processing Noise.* CRC Press, 2002.

[84] Brian D. Jeffs Matthew Willis and David G. Long. A new look at maximum entropy image reconstruction. In *Conference on Signals, Systems, and Computers*, volume 2, pages 1272–1276, 1999.

[85] J. S. Fleming D. M. McGrath, G.J. Daniell. Maximum entropy deconvolution of low count nuclear medicine images. In *Conference on Image Processing and Its applications*, volume 1, pages 274–278, 1997.

[86] Jie Tian Xiping Luo. Multi-level thresholding: Maximum entropy approach using icm. In *15-th International Conference on Pattern Recognition*, pages 778–781, 2000.

[87] Yuqing Gao Hong-Kwang Jeff Kuo. Maximum entropy direct models for speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 1–6, 2003.

[88] Joshua Goodman. Classes for fast maximum entropy training. In *IEEE international conference on Acoustics, Speech and Signal Processing*, volume 1, pages 561–564, 2001.

[89] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1984.

[90] http://mathworld.wolfram.com.

[91] Radan Kucera. *Counting and Configurations: Problems in Combinatorics, Arithmetic, and Geometry.* Canadian Mathematical Society, 2000.

[92] Ward G. Jackson J. C. Surface inspection of steel products using a synthetic aperture microwave technique. *British Journal of NDT*, 33(8), 1991.

[93] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 3 edition, 1968.

[94] C.T Kelley. *Iterative Methods for Optimization.* Society for Industrial and Applied Mathematics, 1999.

[95] Claude Lemarechal Claudia A. Sagastizabal J. Frederic Bonnans, J. Charles Gilbert. *Numerical Optimization, Theoretical and Practical Aspects.* Springer, 2000.

[96] Fletcher R. *Practical Methods of Optimization.* John Wiley & Sons, Inc., 2 edition, 2003.

[97] Wright Stephen J. Nocedal J. *Numerical Optimization.* Springer, 1999.

[98] Lange Keneth. *Optimization.* Springer, 2004.

[99] ADRIAN S. LEWIS JONATHAN M. BORWEIN. Convex analysis and nonlinear optimization, theory and examples.

[100] Leonard D. B. *Convexity and optimization in* $\mathbb{R}^n$. John Wiley & Sons, Inc., 2002.

[101] Bhatti M. Asghar. *Practical Optimization Methods with Mathematica*^{TM}*Applications.* Springer, 1998.

[102] Charles F. Van Loan Gene H. Golub. *Matrix Computations.* The Johns Hopkins University Press, third edition edition, 1996.

[103] J.D. Hoffman. *Numerical methods for Engineers and Scientists.* Marcel Dekker, Inc., second edition edition, 2001.

[104] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Yousef Saad, 2 edition, 200.

[105] Y. Saad and M. Schultz. Gmres a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856869, 1986.

[106] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* Society for Industrial and Applied Mathematics (SIAM), 1995.

[107] A. Iserles at alii P.G. Ciarlet. *Iterative Krylov methods for large linear systems*, volume 13 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, 2003.

[108] Homer F. Walker. Implementation of the gmres method using householder transformation. *SIAM J. Sci. Stat. Comput.*, 9:152–163, 1988.

[109] Owe Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1996.

[110] G. Meurant. *Computer Solution of Large Linear Systems.* North-Holland, 1999.

[111] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems on which the coefficient matrix is a symmetric m-matrix. *Math. Comp.*, 31:148–162, 1977.

[112] Ke Chen. *Matrix Preconditioning Techniques and Applications*, volume 19 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, 2006.

[113] T.F. Chan and H.A. van der Vorst. Approximate and incomplete factorizations. In A. Sameh D.E. Keyes and V. Venkatakrishnan, editors, *Parallel Numerical Algorithms*, pages 167–202. ICASE/LaRC Interdisciplinary Series in Science and Engineering, 1997.

[114] M. Benzi. Preconditioning techniques for large linear systems: A survey. *J. Comput. Phys.*, 2003.

[115] Henk A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems.* Cambridge University Press, 2003.

[116] C.D. Meyer. *Matrix analysis and applied linear algebra.* SIAM, 2000.

[117] Jose M. Tribolet. A new phase unwrapping algorithm. *IEEE Transactions On Acoustic, Speech, And Signal Processing*, ASSP-25(2):170–177, 1977.

[118] F. Bonzanigo. An improvement of tribolet's phase unwrapping algorithm. *IEEE Transactions On Acoustic, Speech, And Signal Processing*, ASSP-26(1):104–105, 1978.

[119] Richard McGowan and Roman Kuc. A direct relation between a signal time series and its unwrapped phase. *IEEE Transactions On Acoustic, Speech, And Signal Processing*, ASSP-30(5):719–726, 1982.

[120] Kenneth Steiglitz and Bradley Dickinson. Phase unwrapping by factorization. *IEEE Transactions On Acoustic, Speech, And Signal Processing*, ASSP-30(6):984–991, 1982.

[121] Rudin W. *Principles of Mathematical Analysis.* Rudin, 1976.

[122] Webster R. J. *Convexity.* Oxford University Press, 1994.

[123] Luetkepohl H. *Handbook of Matrices.* John Wiley & Sons, Inc., 1996.

[124] Sullivan F. E. Peressini A. L. *The Mathematics of Nonlinear Programming.* Springer, 1998.

[125] Lipschutz Seymour. *Schaum's outline of Theory and Problems of Linear Algebra.* McGraw-Hill, 1991.

[126] Gradshteyn I. S. and I. M Ryzhik. *Tables of Integrals, Series, and Products.* CA: Academic Press, 6 edition, 2000.