Acknowledgment

This report is submitted in partial fulfillment of the requirements for Master of Science (Candidatus Scientiarum) degree in Informatics at the Division of Mathematical Modelling, Department of Informatics, University of Oslo. The work was started in August, 1993 and finished in November 1994.

First and foremost, I would like to express my gratitude to my thesis supervisor, Professor Sverre Holm. It has been a great pleasure to work under an inspiring and patient tutor. His pleasant being, encouragement and persistent interest throughout the whole period of the project has been a major incentive for making an extra effort.

Special thanks to my friend, Trond Kleveland at Vingmed Sound AS, for excellent promotion of the field of ultrasound and for a deliberate first introduction into complicated topics.

Thanks also to Professor Hans Pécseli, Division of Plasma and Space Physics at the Institute of Physics, University of Oslo for constructive comments on the acoustic part of my thesis.

I am also grateful to my colleagues at work as a teaching assistant in Digital Signal Processing, Hongtao and Kamran for instructive discussions, good cooperation and many helps.

I wish to thank my colleagues at the Department of Informatics, Geir, Shanthan, Olav, Jon Ole and Bahee among others, for many creative discussions and a pleasant social environment.

Finally, I reserve my deepest appreciation for my dear wife, Eldbjørg, for her support, encouragement and patience during this laborious period, and to my one year old son, Jonas, tacitly accepting my absence.

Oslo, November 1994

Jan Ove Erstad

Contents

Chapter 1

Introduction

In our daily life we have to sense and interpret a large amount of information from our environment. To be qualified for this task we are created with sophisticated senses which takes care of both detection and processing as well as the interpretation.
Our eyes are able to detect electromagnetic radiation with wavelengths in the area of about 350 to 780 nm [1]. This radiation carries an immense amount of information. In a similar way the human ears detect acoustic radiation with frequencies from about 20 Hz to 20 kHz.
Light and sound are waves radiating in either air, fluids or solids. Sound is considered to be mechanical wave motion and light as a form of electromagnetic wave motion. The waves convey signals that may reveal information about the radiating source.
Similarly we can transmit a wave towards a system and study the reflected or *scattered* wave to obtain some information about the system. This is one of the basic principles of medical ultrasonics. Scattering is the effect caused by an obstacle or inhomogeneity hit by sound and thereby spreading out secondary sound in a variety of directions. Examples of scatterers are fish in the sea or a red corpuscle in a bloodstream.

Signals from our visual system and sense of hearing are subject to additional analysis in our brain. From this analysis we are able to make decisions and act on the basis of what we see and hear, for example to avoid danger. Human senses and our way of extracting necessary information form a pattern for many developments in science.
We are for example able to distinguish between sources from different directions with our two ears. It would not be possible to do this with only one ear, because then we would lack the directional information. This motivates the use of *array*[2] systems to detect signals carried by propa-

[1] nm = nanometer = $1 \cdot 10^{-9}$ meter

[2] I will use the term array for a collection of separate detectors (judiciously placed at distinct points in space).

gating waves.
A collection of sensors will therefore spatially sample the field which it lies in. Arrays provide directional information (i.e. the beampattern of the array) and also improve the signal-to-noise ratio (i.e. the gain of the array) compared to single sensor systems. This permits detection of signals from various directions as well as suppression of noise and signals from unwanted directions.
Sensors that gather signal energy over a finite area in space, are called *apertures*[23]. The shape and size of sensors and apertures varies as they are being designed for different purposes. For example will the size of sensors typically be determined by the wavelength of the propagating signal. An example of a sensor's shape is the parabolic radar dish. Similarly the configuration of sensors in an array varies according to the application it is designed for.
Since new applications and new imaging modes are being continuously developed and improved, the research on apertures and arrays remains a major activity.

1.1 Objective of the work

In ultrasound the transducer is the essential component, but also the main limiting factor that affects the quality of imaging and velocity measurements. Ultrasound scanners using $N \times 1$ elements, e.g. with $N = 128$, are now widely used.
However, in the last few years two-dimensional ($N \times K$) transducer arrays have frequently been addressed in the scientific literature e.g. in [52, 60]. A major problem arising is the large number of elements in a complete two-dimensional array. This motivates the use of *sparse arrays* (i.e. where non-active or missing elements exist) in order to reduce the amount of increased complexity demanded for the new technology.
Sparse arrays are also possible to use with low cost ultrasound scanners. In other words one could use sparse arrays with scanners built for e.g 128 element arrays, but with front-end electronic circuitry only for 10–30 channels. Synthetic aperture techniques have also been proposed to accomplish a similar effect [25] with these low cost scanners.

The goal of the work in this thesis is to find a way to optimize the responses from sparse and non-equally spaced arrays. This must be done with respect to a given criterion that ensures as low a sidelobe level as possible, while still giving a substantial resolution.
The motivation for this is obvious, due to the importance a large contrast resolution has in an ultrasound system. Since the use of sparse arrays causes a considerably increased sidelobe level that is unacceptable in a practical system, finding ways to subdue the sidelobes are strongly de-

sired.
In this study, we have chosen to formulate a routine based on the *Remez exchange algorithm* [8]. By using a spatial angular frequency axis an algorithm can be formulated, that has similarities with a FIR-filter optimization routine [37].
Because an ultrasound application of the arrays is an underlying intention, some theory related to this field must necessarily be studied. This includes both acoustics and ultrasound principles.

1.2 Organization of the text

In this text the work on an array design approach and some simulation results are presented. Secondary to this, it also contains a review of some of the basic theory regarding ultrasonics, acoustics and coarrays. Only with some knowledge of these underlying principles, a fully understanding of the work in this thesis is possible. Therefore some theory of these topics will also be included. This should give a better overall view of the problem treated in this thesis as well.

Including the current one, this thesis consists of 7 chapters, 2 appendices and a bibliography.
Chapter 2 is intended to give a brief introduction into the imaging process. The goal of the imaging process is to display the structures beneath the surface where the sound is transmitted, with a maximum quality. Interaction between the medium, the probe, the scanner and the output devices will all be influencing the final result. Only the basic ultrasound principles will be reviewed in this chapter.
In ***Chapter 3*** some physical principles of wave propagation are examined. In other words we will be considering the field that the acoustic sound waves are transmitted into and received from (Fig. 2.1). The wave equation is THE equation when considering propagating waves. It will be shown that this equation get additional terms when we are considering different boundary conditions.
The remaining sections of the chapter is in full a discussion on different solutions to the wave equation. Some of the assumptions that must be made in order to get simpler and more computational efficient solutions to the acoustic problem are pointed out.
Chapter 4 is in a discussion of properties of transducer arrays and coarrays. Fundamentals of coarrays are summarized and next some properties of the linear and planar arrays are discussed. Finally beampatterns are seen to be computable directly from the coarray lag values. This method offers an approach alternative to the routine already implemented in UltraSim [4, 59] and simulation results from these two methods are compared.
Chapter 5 is the main chapter with respect to array synthesis. A gen-

eral Remez exchange algorithm is formulated and implemented, that can be used for optimization od weight coefficients both for uniform, perturbed, sparse and non-uniformly spaced arrays. This algorithm is also used for further improvements on arrays proposed in the literature.
Chapter 6 is dedicated to results made in the simulations with various arrays. Our optimizing routine has been used to improve the sidelobe levels of some arrays.
The conclusions regarding the results made in this work is summarized in ***Chapter 7***. Propositions for further work are also given there.

A central philosophy in this thesis has been to present complicated material in simple manner, hopefully it is done without sacrificing the clarity of exposition.

Chapter 2

Principles of ultrasonics

This chapter is intended to give the motivation for the succeeding chapters. Ultrasonic principles is not the main subject in this work, but it is one of the areas where the problem of designing well behaving arrays arises. The basic principles of ultrasonic imaging and Doppler measurements will now be revealed. Special concern will be given to the properties of imaging quality and to transducer geometries. These are considerations that will be further elaborated in the coming chapters.
Arrays and antennas are also used in the fields of sonar, seismology, radar and radio telescopes [23], and array systems as such would lead to similar demands as for the ultrasonic ones.

2.1 Introduction

Ultrasound is acoustic sound waves with frequencies above the audible range (i.e. above 20 kHz), in contrast to infrasound which is waves below the audible range.
In this thesis ultrasound is viewed from a technical point of view, where the sound is produced by instrumental devices. For the curious reader and as a digression it can be mentioned that ultrasound is widespread in the nature. Bats, for example, orientates themselves by means of ultrasound and sea animals like shrimps and crabs also makes ultrasound.
In the 'technical world' ultrasound has won a wide application in medicine during the last decades. It is now used for diagnosis and therapy as well as a surgical tool. The ultrasonic waves are reflected from boundaries with approximately the same density and can in contrast to e.g. X-rays, be used without harmful effects within certain accepted limits. Ultrasound used for therapy have higher transmitted power than what is recommended and accepted for imaging and measurements. And for surgery even higher power is used.

Ultrasound waves can be transmitted as continuous waves (CW) as well as pulsed sound waves (PW). In case of PW the transducer acts as a receiver detecting wave reflections and echoes between the pulses. When using CW the ultrasound probe must transmit and receive simultaneously and therefore it contains separate transmit and receive aperture.

The basic principle is that sound waves are transmitted into the body and either the amplitude of the backscattered signal is used for tissue imaging or the frequency of the Doppler shifted signal is used for estimation of the velocities of the blood flow. The signals received by the transducer must undergo a large amount of signal processing before being supported to the displaying output (as Fig. 2.1 signifies). In order to make it more informative to the operator, additional processing on the images usually is performed e.g. coding blood flow in different directions with different colors.

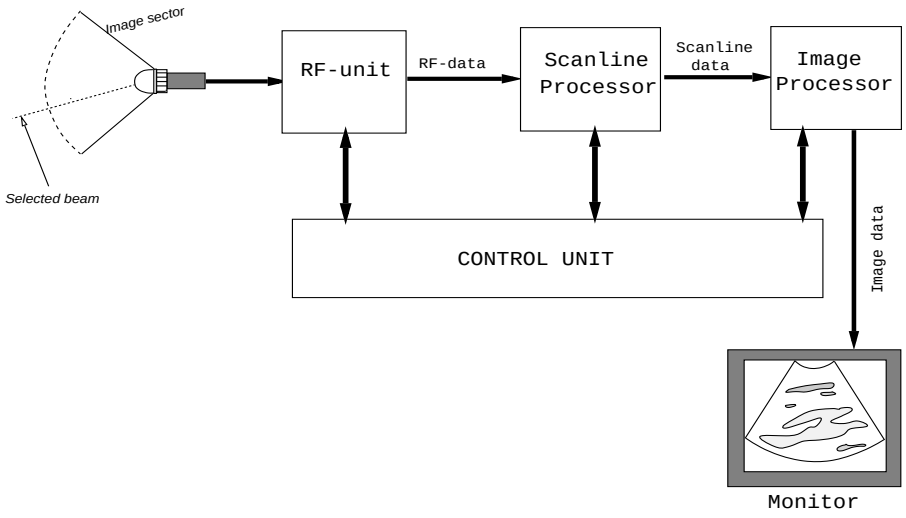


Figure 2.1: Block diagram over an ultrasound imaging system.

Images produced by a medical ultrasound system are interpreted by the medical staff who for example make a diagnosis based on the images or use the image to point the beam towards an area of interest for measurements. Making a precise diagnosis then relies heavily on the quality of the images. Pointing the beam depends on the quality of the images as well as the experience of the user.
There are several limitations and conflicts in the imaging process. Severe problems arise when the sound meets boundaries with relatively large density differences e.g. between bone and tissue or between air and skin. In addition causes fatty layers *aberration* that also degrades the image quality. Because of these problems special probes for insertion through body openings, have been developed. In that way one can for example get behind the ribs and close to the heart.

The imaging quality of a ultrasound system can be described in terms of the spatial resolution, the depth of field (DOF), the contrast resolution and the frame rate. The frame rate is particularly important when imaging fast moving structures. In designing a system one must be aware of the tradeoff between these desired qualities.

Destructive effects from ultrasound waves also exists. Intense ultrasound produces large density and pressure changes in tissue, within each small wavelength, and molecules are then forced to move rapidly. This produces heat and *cavitation* in some materials (as in liquids). Cavitation is the formation of bubbles of vapor caused by the mechanical fracture of the liquid in a region where the pressure is decreasing. These bubbles may then collapse violently. This is utilized in inhalation therapy, where the medication is broken into droplets fine enough to enter the alveoli of the lungs.
In physiotherapy the tissue heating effect is utilized in the treatment of for example inflammation of tendon sheath (tenosynovities) and injuried muscles. However, care must be taken in order to avoid damage caused by heating the tissue too much. The success of this treatment then relies on guessing an appropriate ultrasound power to emit, since a precise determination of the temperature in a particular tissue segment and hence the precise effect on the tissue to be treated, is difficult to establish.

2.2 Ultrasound transducer types and beams

There are several types of ultrasound probes specially designed for numerous distinct applications. A probe designed for insertion into the body has serious contraints in size, while arrays for imaging in the abdominal area do not have the same constraints. Accordingly, numerous distinct transducers are applied in ultrasound imaging and blood velocity measurements. In this text they are divided into linear, two-dimensional and annular transducers.
In principle the transducers can both be continuous apertures as well as discrete arrays. Only the latter type will be examined closer in this text. Curving of the transducer surface will introduce focusing of the beam. The geometrical focus will be in the center of the circle that is the prolongation of the transducer surface. A transducer can be curved to give focus both for positive or negative values on the z-axis. Transducers with negative focus (i.e. the focal point is behind the transducer face as in fig. 2.2) are used to get wider image fields. Transducers curved to give focus in front of the face will give both a narrow transmit beam and receive beam at the focal point.
From physics it is known that the imaging properties of an aperture (e.g. a hole in the wall or a lens) with finite spatial size is limited by

the diffraction limit. Due to the diffraction effect the sound from the transducer will be diffracted and the real focus will be wider than what is obtained with geometrical beams [1].
By mechanically tilting the transducer it can be steered in different directions and one can therefore sweep the beam over a sector. Thus we can shoot the beams sequentially over the sector and then scan the field to make a two-dimensional (2D) image. Beam steering is advantageous when we want to cover an area with a small transducer.
Focusing and steering of the beam is also possible to obtain electronically by arrays. This implies use of delay lines in order to delay the signal from element to element. Planar arrays allow beam steering and focusing in both elevation and azimuth dimensions (Fig. 2.2).
The delays in each array element can be varied with time in order to make the focus follow the reflections of the pulse from successive deeper depths. This sweeping of the focus is called *dynamic focusing*. Since the size of the aperture in relation to the distance to the focusing point determines the width of the focus , it is necessary to vary the size of the aperture to get an uniform width of the focus. This varying is called *dynamic aperture*.

Linear arrays have their elements distributed in one dimension (i.e. along one axis). Since they only have a single element in the elevation dimension they cannot be steered and focused in this direction. Thus steering and focusing with linear arrays can only be performed in the azimuth dimension. Depending on the application they can have different shapes and sizes. A further discussion on linear arrays with respect to resolution and sidelobe control is given in Sec. 4.3.
1.5-dimensional (1.5D) arrays are as the name indicates, arrays with characteristics between one- and two-dimensional arrays. They have three or five elements in the elevation dimension and can be curved in both dimensions. However, electronically steering is performed in the azimuth direction only.
Two-dimensional (2D) arrays point out to be important for improving the imaging quality of ultrasonic systems. These arrays can be both mechanically and electronically steered and focused. Several examples of improvements due to the use of such arrays are suggested in [52]. Because 2D arrays provide several advantages they will probably become valuable for introduction of new modes e.g. three-dimensional (3D) imaging with a mechanically tilted and rotated 2D array. The arrays will be most important for external imaging. It is also desired to be able to steer and focus these arrays electronically.
Unfortunately there are some fundamental problems of both constructing and using 2D arrays. In many applications there is a limit on the size of an array and this gives difficulties when connecting the tiny array elements. A filled two dimensional array will require an extremely large number of elements. The processing will be time consuming with this number of elements. Compared to a linear array the increase in operations will be the power of two.

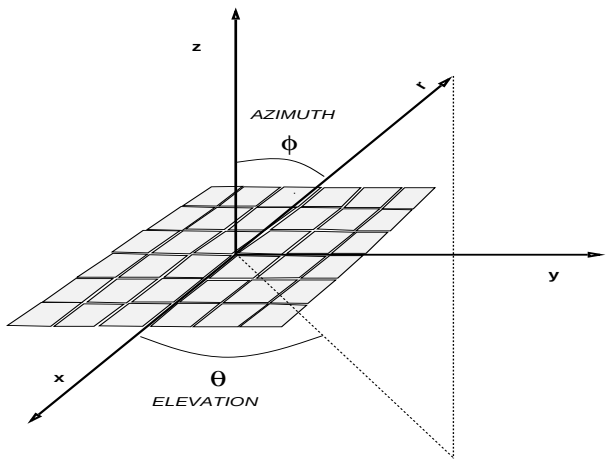


Figure 2.2: A two dimensional planar array both with cartesian and spherical coordinates given. The array could also be curved in both azimuth and elevation dimension e.g. in order to allow focusing.

Due to these problems one has to reduce the number of transducer elements by undersampling of the aperture. This can be done by either eliminating some elements from the aperture or by increasing the element size and the interelement distance. The latter solution will not be possible where there are restrictions on the aperture size e.g. transducer arrays for imaging of the heart through the chest (i.e. between the ribs). Additional problems also arise when it is not possible to use all array elements.
Moreover 2D arrays increase the complexity of analysis and synthesis considerably. This makes it more difficult to calculate optimal apodization for the array.
Annular transducer arrays (Fig. 2.3) are made up of concentric rings. This type can be focused both in the plane as well as transverse to the plane, in other words we have a symmetric focus. Since it consists of rings it can not be electronically steered and thus have to be mounted so that it can be mechanically rotated or moved linearly to produce an image.

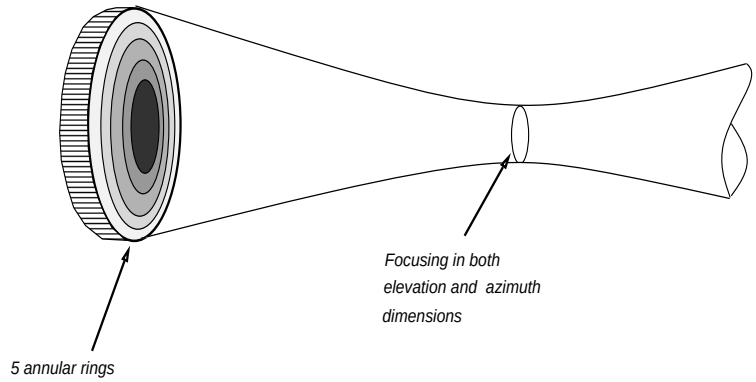


Figure 2.3: An annular array with 5 annular rings giving symmetric focus.

2.3 Imaging quality

Over the past decades different imaging modes have been introduced in ultrasound imaging.
In *B-mode (Brightness mode)* beams are sequentially fired at different angles and a sector is then covered by the beams. The amplitude of the received signal is displayed with intensity proportional to amplitude. For each beam, waves are received from a number of distinct dephts. In this way an intesity image is formed along each beam. Repeating this in different directions is equivalent to performing A-mode (Amplitude mode) for each beam over a sector.
Imaging in *M-mode (Motion mode)* will give a 2D image with time as one of the axis. A beam is fired in a particular direction. The amplitude of the reflected or backscattered waves are measured at a given number of depths. Then new beams are fired at the same direction and new echoes are received. By doing this repeatedly we get a real-time image of the moving structure in the beam direction.

An important factor that determines the imaging quality of a system is the resolution.
The *spatial resolution* is determined by the frequency and the size of the aperture. Laterally i.e. transverse to the beam, the resolution is determined by the width of the beam. For a linear array this can be seen from where the sinc function (see page 38) has its first zero crossing. The first zero of Eq. 4.5 appears when $sin(\frac{M}{2}k_x d) = 0 \Leftrightarrow k_x = \frac{2\pi}{Md}$. Consequently the resolution depends on the array's overall spatial size (i.e. the aperture, Md). When the denominator is zero, grating lobes arise. In order to avoid grating lobes in the visible region (i.e. $\pm 2\pi/\lambda$) the element spacing, d, must be less than or equal to half the wavelength (i.e. $d \leq \lambda/2$). This also limit the size of the individual array elements.
The resolution along the beam is determined by the length of the transmitted pulse. To increase this resolution, shorter wavelengths should be chosen. However, this implies higher frequencies and they are more heavily attenuated in tissue than lower frequencies. This leads to a conflict between between wanting high resolution and sufficient penetration.
In practice one chooses the frequency dependent on the desired depth of penetration, and then get the best possible resolution at that particular depth. Theoretically one could increase the size of the aperture in order to circumvent this problem but in most applications the size of the aperture is limited and can therefore not compensate for lower frequencies.

The *contrast resolution* (also called the dynamic range) is limited by the level of the sidelobes and by multiple reflections of the ultrasound pulse. The latter effect is often called *reverberations*. Sidelobes causes signal from directions outside the main lobe to be picked up and added to the

receive signal. This gives noise in the image. To obtain a good image quality a low sidelobe level is demanded.
One way of reducing the sidelobes is by apodization of the aperture or array. The sidelobe level will then be controlled by the chosen weighting function. However, it will also cause a loss in spatial resolution.

2.4 Blood velocity measurements

Another major area of medical ultrasonics is using the Doppler effect to measure blood velocities or to produce color coded images of the blood velocity. The PW and CW Doppler measurements are used to quantify and display temporal variations of the velocities or in the flow.
As for tissue imaging the sound waves are transmitted into the body. When the scatterer is moving the frequency of the backscattered signal will deviate from the frequency of the emitted signal. The change in frequency is the Doppler effect.
The Doppler shift is given by the *Doppler equation*

$$f_d = 2f_0 \frac{v \cos\theta}{c} \tag{2.1}$$

here f_0 is the transmit frequency, c is the wave propagation velocity, θ is the angle between the velocity direction and the beam direction and v is the velocity of the scatterers. What is actually measured is the velocity component radially i.e. along the beam.
The signal received will in addition to the Doppler shift of the scatterers also contain noise in the form of strong echoes from slowly moving tissue as well as thermal noise. This implies difficulties in calculating the blood velocities because we have a distribution of frequencies contained in the signal and some of the frequencies are not from the signal. The tissue noise can be damped by a varying highpass filter that attenuates the lowest frequencies. Removing the thermal noise is far more complicated since it is distributed over the entire frequency band of the velocity spectrum.
A drawing illustrating a typical time-varying frequency spectrum is given in fig. 2.4. It consists of some 6-8 measurements producing the same number of spectrum estimates. The tissue frequency components are seen as a strong peak to the left in the figure. In the middle some components are present only in a portion of the measurements. This is where we expect the blood velocity components to be. Because of the pulse of the heartbeats the velocities are varying and frequency components of the blood will not be present in all subsequences.
There may be blood velocity components below the valley between the tissue and blood peaks, but these are drowned in the tissue and thermal noise. When we filter the tissue components with a higpass filter (in the area indicated by 'HP'), we will also loose the low blood velocity blood

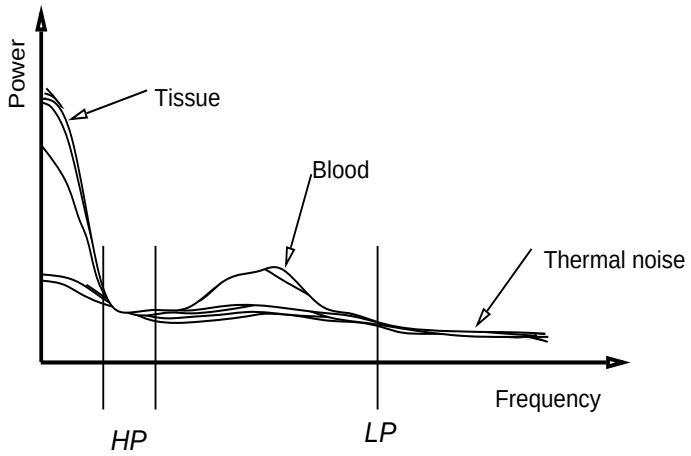


Figure 2.4: Estimated power spectrum for Doppler signal.

components. It is therefore important to send and receive the beams as parallel to the blood cells motion direction as possible. It must be mentioned that if very high velocities are to be measured the beam should be pointed at a larger angle to the blood flow.
The thermal noise can be regarded as approximately the floor in the spectrum of Fig. 2.4. Thus these noise frequencies cannot be subdued by filtering. Though a lowpass filter (LP in the figure) can be used to eliminate frequency components above the Doppler shift limit.
However, the amount of noise can be reduced by allowing a larger range cell to be selected [1]. From the received signals the directions of the blood velocities can be found. By coding them in different colours we get what is called a *colour flow image* (CFM).

2.5 Summary

The basic principles of ultrasound imaging and velocity measurements have been presented. A variety of distinct arrays can be fabricated, and is too. This allows transducer probes to be used in diverse areas of medicine. Ultrasound also has application to *non-destructive testing* e.g for testing the quality of materials and welds.
Since we will be considering array synthesis, and the resulting beamforms from arrays, we should take a closer look at the theoretical fundament that the understanding of propagating signals must be built on. A physical interpretation and mathematical tools for the propagation of sound waves from transducers, are offered by the theory of acoustics. This will be reviewed in the subsequent chapter.

Chapter 3

Acoustic theory review

In this chapter terms and equations related to wave propagation in the ideal case will be reviewed. The acoustic wave equation is derived for both the pressure and the velocity potential. Solutions known for three boundary conditions will also be reviewed. Finally several approximations usually introduced in order to make calculations less bothersome are mentioned and a brief review of some simulation methods is given at the end of the chapter.

3.1 Wave motion

Periodic disturbances and oscillations play an important role in many different areas in science and engineering. The phenomena of oscillations travelling in space are called *wave motion*. By throwing a stone into a pond one can observe wave motion by watching the ripples spreading across the surface of the water. Other examples are sound waves, optical waves or vibrations on a string.
The model of wave motion in one spatial dimension is described by the *scalar wave equation*[1]

$$\frac{\partial^2 u}{\partial s^2} - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0, \tag{3.1}$$

where u is the wave motion in a point of interest and c is the speed of sound which varies in different media. The wave motion can be both transverse and longitudinal. A vibrating string is an example of transverse waves, where the string displacement is normal to the direction of the wave motion. A solution to this equation can be written as

$$u(s,t) = f(s-ct) + g(s+ct) \tag{3.2}$$

[1] In math books this equation is often written more compact as : $u_{tt} - c^2 u_{xx} = 0$

The solution consists of the sum of two travelling waves. One wave, f, travelling to the left on the s-axis and the other wave, g, travelling to the right.
Similarly we have plane travelling waves in acoustics. In that case all the acoustic variables are functions of time and only one cartesian coordinate, s, and are independent of position along planes normal to the s-direction. From the form of the solution we see that the magnitude and waveforms are conserved as the wave propagates.
In acoustics the oscillations of the material are often called *particle motion*. This motion appears in the three dimensional space. Hence the wave equation must be formulated in three dimensions. The Laplacian operator ∇^2 in three dimensions (3D) is then used

$$\nabla^2 \triangleq \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

In case of *transverse waves* the particle movement is normal to the direction of propagation. For *longitudinal waves* the particle movement is along the direction of propagation.
In inviscid fluids, acoustic waves are longitudinal waves. This means that ultrasonic waves in biological tissue are longitudinal, since biological material is considered as mainly water with some solids added. Moreover, the waves have nearly the same wave velocity as in water. In fact the wave velocity in biological material is only slightly above that of water. Transverse waves can also exist but they are heavily attenuated in biological material [1].

Acoustic waves are caused by cyclic compression and expansion of regions in the medium. The fluid particles are moving back and forth in the direction of propagation of the wave and thereby are producing adjacent regions of compression and decompression. Compression causes the pressure to increase and expansion causes the pressure to decrease in a particular region of the fluid.
Then at a particular spatial point pressure variation will pass with the propagation velocity, c, and oscillate with the temporal frequency, f. At a particular instant the spatial period between pressure maxima is equal to the wavelength, λ. The relation between these parameters is very important for propagating signals in general

$$\lambda = \frac{c}{f} = \frac{2\pi c}{\omega}, \tag{3.3}$$

where ω is the temporal angular frequency equal to $\omega = 2\pi f$.
Another important relation is regarding the magnitude of the *wavenumber vector*

$$k = |\,\vec{k}\,| = \frac{\omega}{c} = \frac{2\pi}{\lambda} \tag{3.4}$$

The wavenumber is considered to be a directional spatial frequency. It should be noted that the wavenumber vector will be denoted by $\vec{k}$, while its magnitude will be written k.

Relations like $\omega = ck$, between the temporal frequency, ω, and the wave-number magnitude, k, (spatial frequency) are called dispersion relations [23]. Media where his relation is linear are called *dispersive media*.

3.2 The acoustic wave equation

Sound propagation in space is modelled by the acoustic wave equation, which is quite similar to Eq. 3.1. A detailed derivation of this equation can be seen in [28, 46]. The basis of the derivation is three relations.
First the *conservation of mass*, where the net rate of mass flowing into a fixed volume is considered equal to the increase of mass inside the volume. This gives a relation between density and particle velocity. The resulting equation is nonlinear, but assuming small vibrations validates a linearized equation of continuity (Eq. 3.5).
The next consideration, *Euler's equation of motion for a fluid* also gives a nonlinear equation, relating pressure and particle velocity. Linear approximation (due to the small disturbances assumption) gives Eq. 3.6. These equations are called the *basic linear equations of acoustics*.
Finally the relation between pressure and density is given by Eq. 3.7.

$$\frac{\partial \rho}{\partial t} = -\rho_0 \nabla \cdot \vec{v} \tag{3.5}$$

$$\rho_0 \frac{\partial \vec{v}}{\partial t} = -\nabla p \tag{3.6}$$

$$p = c^2 \rho \tag{3.7}$$

Here p is the acoustic pressure at an arbitrary point in space $\vec{x}$, $\vec{v}$ is the particle velocity, ρ is density at a point, ρ_0 is the equilibrium density of the medium, t is the time and c is the speed of sound in the medium.
From the linear acoustic equations above the acoustic wave equation regarding the pressure is easily derived. Using Eq. 3.7 as $\rho = p/c^2$ in Eq. 3.5 and then differentiate with respect to time, gives:

$$\begin{aligned} \frac{1}{c^2}\frac{\partial p}{\partial t} + \rho_0 \nabla \cdot \vec{v} &= 0 \\ \frac{\partial}{\partial t}\left(\frac{1}{c^2}\frac{\partial p}{\partial t} + \rho_0 \nabla \cdot \vec{v}\right) &= 0 \\ &\Downarrow \\ \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} + \rho_0 \nabla \cdot \frac{\partial \vec{v}}{\partial t} &= 0 \end{aligned}$$

Finally using Eq. 3.6 as $\partial \vec{v}/\partial t = -\nabla p/\rho_0$ in the previous equation, results in the *acoustic wave equation*[2]:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{3.8}$$

[2] In math books often written as $u_{tt} - c^2 \Delta u = 0$

The speed of sound c equals

$$c^2 = (\partial p/\partial \rho)_0,$$

evaluated at the equilibrium conditions of the pressure and density. In water this relation is written as [46] : $c = \sqrt{B/\rho_0}$, where B is the adiabatic bulk modulus.

3.2.1 The velocity potential

An alternative formulation leading to the wave equation is in terms of the velocity potential Φ. This formulation is often preferred because it is rather convenient to describe an acoustic field in terms of a single function from whom all field quantities can be derived.
From calculus we know that the curl of the gradient of a function must vanish i.e. $\nabla \times \nabla f = 0$. Then taking the curl of both sides of Eulers equation of the particle velocity (Eq. 3.6) yields $\nabla \times \nabla p = 0$ and thus $\nabla \times \rho_0 \frac{\partial \vec{v}}{\partial t} = 0 \Leftrightarrow \rho_0 \frac{\partial}{\partial t}(\nabla \times \vec{v}) = 0$. Then $\nabla \times \vec{v}$ is constant with time.
Most often one considers the initial value of $\nabla \times \vec{v}$ to be zero, thus it is always zero $\nabla \times \vec{v} = 0$, since $\rho_0 \neq 0$. This signifies that the particle velocity $\vec{v}$, must be irrotational (and hence conservative).
Consequently the velocity has a scalar potential function Φ, and $\vec{v}$ is its gradient :

$$v(\vec{x}, t) = \nabla \Phi(\vec{x}, t) \tag{3.9}$$

Substituting this equation in Eq. 3.6 yields:

$$\begin{aligned} \rho_0 \frac{\partial}{\partial t} \nabla \Phi(\vec{x}, t) + \nabla p(\vec{x}, t) &= 0 \\ \nabla \left(\rho_0 \frac{\partial}{\partial t} \Phi(\vec{x}, t) + p(\vec{x}, t) \right) &= 0 \\ p(\vec{x}, t) &= -\rho_0 \frac{\partial \Phi(\vec{x}, t)}{\partial t} \end{aligned} \tag{3.10}$$

Substituting Eq. 3.10 into Eq. 3.8 and integrating it with respect to time shows that Φ also satisfies the wave equation.
The wave equation in terms of the velocity potential is written

$$\nabla^2 \Phi - \frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2} = 0 \tag{3.11}$$

3.2.2 Spherically symmetric waves

If the source is generating motion with no preferred direction, it will produce waves that spreads out spherically. The wave equation may also be written in terms of the spherical coordinates r, ϕ and θ. Here the angle θ denotes the elevation and ϕ is the azimuth. The wave equation

for sperical waves will be quite intricate since the Laplacian in terms of spherical coordinates is defined [55]

$$\triangle_{3D} \triangleq \frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r} + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\Big(\sin\theta \frac{\partial}{\partial \theta}\Big) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2}{\partial \phi^2} \tag{3.12}$$

If the medium is infinite in extent, the waveform will only depend on time and the distance r from the source center [40]. In other words we have spherical symmetry. Thus the parameters of the wave equation will be independent of the angular coordinates ϕ and θ.
Consequently, the general wave equation for spherically symmetric problems becomes [40]

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial p}{\partial r}\right) - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{3.13}$$

With some manipulation, this equation can be rewritten with rp as the single dependent variable [28]

$$\frac{\partial^2 (rp)}{\partial r^2} - \frac{1}{c^2}\frac{\partial^2 (rp)}{\partial t^2} = 0$$

and we observe an equation similar to the one in Eq. 3.1. Thus a general solution of this equation is [46]

$$\begin{aligned} rp &= f(r-ct) + g(r+ct) \\ &\Updownarrow \\ p &= \frac{1}{r} f(t - r/c) + \frac{1}{r} g(t + r/c) \end{aligned}$$

The first term represents a wave travelling outwards from a point source at the origin and the latter term a wave propagating towards the source. It should be noted that with only one source present (and no influence from surrounding boundaries), waves can only move away from the source. Thus the latter function g is zero. Noting the factor $1/r$, we see that spherical waves (in contrast to plane waves) diminish in amplitude as they propagate outward, but the waveforms remains the same.

3.2.3 Inhomogeneous equations

The wave equations written so far describes linear and ideal medium propagation. Effects from lossy media (where the wave amplitude may decay rapidly) have not been taken into account nor have effects from dispersive media (where Eq. 3.4 will get additional nonlinear terms). Both these effects will give additional terms in Eq. 3.8 and thereby give an augmented wave equation. Attenuation will for example typically give an additional damping term like $\gamma \frac{\partial p}{\partial t}$.
Specially terms for the existence of a source that produces the acoustic disturbance have not been included. In that case we will have to modify

the hydrodynamic equations (on page 18) to include source terms. This leads to an inhomogeneous wave equation. Modification of the linearized equation of continuity (Eq. 3.5) results in [28]

$$\frac{\partial \rho}{\partial t} + \rho_0 \nabla \cdot \vec{v} = F(\vec{r}, t),$$

where $F(\vec{r}, t)$ expresses the mass contribution from the source.
By using the same procedure as we did on page 18 we end up with an inhomogeneous wave equation

$$\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = -\frac{\partial F(\vec{r}, t)}{\partial t} \tag{3.14}$$

The right hand side of Eq. 3.14 is regarded as a source term and consequently vanishes for all regions other than where the source is located.

3.3 Solutions to the wave equation

The remaining sections of this chapter are devoted to solutions of the acoustic problem. The solutions are derived by using Green's functions suited for different boundaries.
Finding solution to the acoustic wave equation is in general a very complex task. Pressure field calculation involves solutions of the wave equation both for the source (vibration problem) as well as the surrounding medium (acoustic problem). Due to the reciprocal influence of the radiated wave and the internal source structure this becomes a difficult problem. To overcome some of the problems one assumes that the source surface is infinitely rigid. Even more simplifying assumptions are used in practice concerning a planar or quasi planar radiating surface and neglecting contributions from the closing surface.
Solutions, U, to the homogeneous wave equation (Eq. 3.8) must satisfy the Helmholtz equation. If we are only interested in the spatial properties, then it is sufficient to find solutions to Eq. 3.15. This equation is in fact the wave equation for harmonic waves.
First we assume a harmonic solution $U(\vec{x}, t) = A(\vec{x}) e^{j(\omega t)}$. Then substitute this solution into the wave equation (Eq. 3.8) and next differentiate twice with respect to time, as done below.

$$\begin{aligned}
\nabla^2 A(\vec{x}) e^{j(\omega t)} - \frac{1}{c^2} \frac{\partial^2 A(\vec{x}) e^{j(\omega t)}}{\partial t^2} &= 0 \\
&\Updownarrow \\
\nabla^2 A(\vec{x}) e^{j(\omega t)} - \frac{-\omega^2}{c^2} A(\vec{x}) e^{j(\omega t)} &= 0 \\
&\Updownarrow \\
\nabla^2 A(\vec{x}) e^{j(\omega t)} + k^2 A(\vec{x}) e^{j(\omega t)} &= 0 \\
&\Updownarrow \\
(\nabla^2 + k^2) A(\vec{x}) e^{j(\omega t)} &= (\nabla^2 + k^2) U(\vec{x}, t) = 0 \qquad \blacksquare
\end{aligned}$$

Thus the homogeneous *Helmholtz equation* is written

$$\left(\nabla^2 + k^2 \right) U = 0 \tag{3.15}$$

and is a wave equation for harmonic waves.

Acoustic waves encountering a boundary will result in both reflected and transmitted secondary waves. The presence of a boundary will therefore affect the pressure field that originates from a source radiating in the vicinity of a boundary or from a part of the boundary. Boundary conditions then arise that, as will be seen later, the Green's functions must satisfy in order to give solutions to the wave equation.
When the wave travels perpendicular to the interface (i.e. normal incidence), the ratio of the reflected to incident pressure and velocity amplitudes, the *reflection coefficients* is written [1, 28]

$$R_p = \left(\frac{z_{i_2} - z_{i_1}}{z_{i_1} + z_{i_2}} \right) \qquad R_v = \left(\frac{z_{i_1} - z_{i_2}}{z_{i_1} + z_{i_2}} \right) \quad , \tag{3.16}$$

where $z_{i_1} = \rho_1 c_1$ is the characteristic acoustic impedance of the fluid where the incident and reflected waves travel. z_{i_2} is the impedance of the fluid where the transmitted wave travels.
Similarly the *transmission coefficients* is written [1, 28]

$$T_p = \frac{2z_{i_2}}{z_{i_1} + z_{i_2}} \qquad T_v = \frac{2z_{i_1}}{z_{i_1} + z_{i_2}} \tag{3.17}$$

We see that the amplitude of the reflected and transmitted pressure and velocity at boundaries is determined by the acoustic impedance differences. It is also seen that the reflection coefficients of pressure and velocity has opposite signs. An increase of the pressure therefore means a decrease of the velocity and vice versa.
The impedance is determined by the density and wave propagation velocity. From the expressions above it is therefore seen that density differences at boundaries influence on the propagation properties of waves. Moreover, the density differences determines the degree of reflection and accordingly the intensity of the reflected wave. Reflections from tissue boundaries with relatively small differences in densities is what is utilized in ultrasound imaging (see Ch. 2).
The coefficients above also implies that the boundary at which the transducer is mounted will influence on the properties of the transmitted and received waves. A transducer mounted in in a soft material will have wave field properties distinct from transducers mounted e.g. unto a ship's body or just surrounded by air. Thus in order to formulate and find correct solutions to the acoustic problem one must impose valid boundary conditions.

3.4 Green's functions

The Green's function is a kind of universal solution for harmonic functions in a domain. According to Strauss [55] all other harmonic functions can be expressed in terms of it. Green's functions appear to be very useful in finding solutions for the acoustical problem. Different functions must be chosen to meet the boundary conditions mentioned in the preceding section.
The basic tool is the *divergence theorem*[3] [12]

$$\iiint_V \nabla \cdot \vec{F} d\vec{x} = \iint_S \vec{F} \cdot \vec{n} dS \tag{3.18}$$

from whom one can derive *Green's second identity* [55]

$$\iiint_V (u\nabla^2 v - v\nabla^2 u) d\vec{x} = \iint_S \left(u\frac{\partial v}{\partial n} - v\frac{\partial u}{\partial n} \right) dS \tag{3.19}$$

In both equations, V, denotes the volume and, S, the body (i.e. the surface) of the volume, while u and v are functions of space. This identity is central when deriving the integral solution in Section 3.6.
Generally it is not easy to find the Green's function for a problem, but for the acoustic problem functions have been derived for different boundary conditions. These functions will be used and a description of how they apply in order to get solutions for three different boundaries are given.

3.4.1 Free-space Green's function

This situation corresponds to a situation where a harmonic source radiates outward into an unbounded medium (Fig. 3.1). In this case $z_{i_2} = z_{i_1}$ since we have no boundary and everything is transmitted (i.e. there is no reflection).

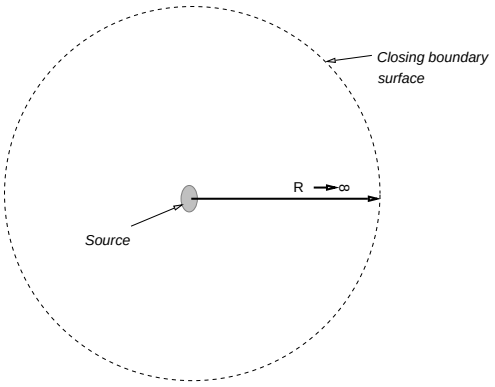


Figure 3.1: Source in Free-space, the closing surface is assumed to be infinitely far away and therefore can be omitted.

[3] The theorem is also known as the Gauss or Ostrogradski theorem.

The chosen function is [40]:

$$g_0(\vec{r}, \vec{r_0}; \omega) = \frac{1}{4\pi R} e^{jkR} \tag{3.20}$$

where $R^2 = |\vec{r} - \vec{r_0}|^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$. It corresponds to a spherical harmonic wave expanding outwards from the point $\vec{r_0}$. The free-space function is a solution of the inhomogeneous Helmholtz equation equation

$$\nabla^2 g_0(\vec{r}, \vec{r_0}; \omega) + k^2 g_0(\vec{r}, \vec{r_0}; \omega) = -\delta(\vec{r} - \vec{r_0}), \tag{3.21}$$

where $\delta(\vec{r} - \vec{r_0}) = \delta(x - x_0)\delta(y - y_0)\delta(x - x_0)$ i.e. a 3D delta function. This Green's function is a solution to the homogeneous Helmholtz equation $(\nabla^2 + k^2)\, g_0 = 0$ everywhere except at the source location $\vec{r} = \vec{r_0}$.

3.4.2 Image source Green's functions

We are now considering radiation from a part of the closing boundary surface or from a source nearby the surface (Figure 3.2). Introducing a surface will cause reflection at the boundary and now the total radiation will be the sum of the transmitted wave and the reflected wave. To deal with this problem it is common to include an image source term to make an appropriate Green's function [14].
In this way the original boundary-value problem of one source and boundary is replaced by a problem with two sources but no boundary [46]. That is combining the Green's function with the method of images (also called the method of reflection).
A common assumption is that the transducer is mounted in an infinite rigid *baffle*[4] [53]. However in [40] a formula for determining an approximate Green's function in case of not perfectly rigid baffles is given. This formula (Eq. 3.25) will be used when deciding appropriate Green's functions for the considered boundaries.
As a start let us consider the impedances on the surface and in the medium. The impedance of the surface z_s is defined

$$z_i \triangleq p/v_n \tag{3.22}$$

Here v_n is the normal velocity and is defined

$$v_n \triangleq \frac{\partial v}{\partial z} \tag{3.23}$$

This implies that the normal vector lies parallel to the z-axis.

[4] A baffle is the boundary surface surrounding the source e.g. a loudspeaker on one side of a large enclosure

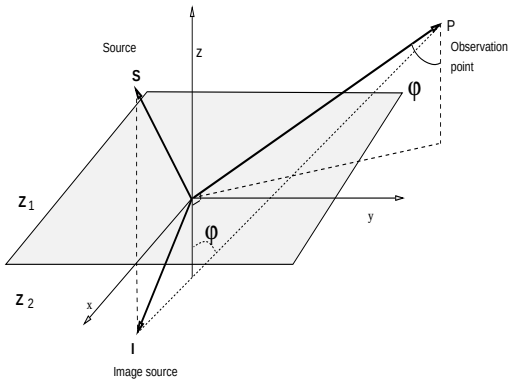


Figure 3.2: Source (S) mounted near a baffle with the image source (I) on the opposite side of the xy-plane. At $z > 0$ the wave will be the sum of the original and the reflected wave. The angle φ in the figure is denoted $\hat{\varphi}$ in the formula below.

The acoustic admittance of the surface is the reciprocal of the impedance and the dimensionless parameter, the *specific admittance* of the surface is defined

$$\beta = \frac{\rho c}{z_s}, \tag{3.24}$$

where ρc is the characteristic impedance of the medium. The Green's function can thus be chosen from

$$G(\vec{r}, \vec{r_0}; \omega) \approx \frac{e^{jkR}}{4\pi R} + \left(\frac{\cos\hat{\varphi} - \beta}{\cos\hat{\varphi} + \beta}\right) \frac{e^{jk\hat{R}}}{4\pi\hat{R}} \tag{3.25}$$

Here $\hat{\varphi}$ is the angle which the vector from the image point source to the field point, makes with the normal to the boundary. The value of $\hat{\varphi}$ has little importance in the further discussion since it is always bounded in the interval $\cos\hat{\varphi} \in [0, 1]$. A special case would be $\cos\hat{\varphi} = 0$ but then the image source would be on the same side of the boundary as the source.

Rigid baffle In case of $z_{i_2} \gg z_{i_1}$ (i.e. $z_{i_1}/z_{i_2} \to 0$) the reflection coefficient R_p (Eq. 3.16) will be approximately one. Thus there is only a slight reduction of the amplitude between the incident and reflected wave. The (normal) particle speed at the boundary is almost zero, $v_n \approx 0$, since $T_v \approx 0$ (Eq. 3.17). Moreover $z_s \to \infty$ (Eq. 3.22) and therefore β becomes very small i.e. $\cos\hat{\varphi} \gg \beta$.
The suitable Green's function is then from Eq. 3.25 chosen to be

$$g_+(\vec{r}, \vec{r_0}; \omega) = \frac{1}{4\pi R} e^{jkR} + \frac{1}{4\pi\hat{R}} e^{jk\hat{R}} \tag{3.26}$$

Soft baffle Considering the case when $z_{i_1} \gg z_{i_2}$ (i.e. $z_{i_1}/z_{i_2} \to \infty$). It can be observed that the reflection coefficient of Eq. 3.16 approaches the value $R_v \to 1$ and the transmission coefficient approaches zero $T_p \to 0$. This signifies that the acoustic pressure is

nearly zero at the boundary and a soft boundary is therefore called a ***pressure release boundary***. Since $p \approx 0 \quad \Rightarrow \quad z_s \rightarrow 0$, and thus β becomes very large i.e. $\beta \gg \cos\hat{\varphi}$.
Consequently the appropriate Green's function is (Eq. 3.25)

$$g_-(\vec{r}, \vec{r_0}; \omega) = \frac{1}{4\pi R} e^{jkR} - \frac{1}{4\pi \hat{R}} e^{jk\hat{R}} \tag{3.27}$$

In both cases $R^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$ and $\hat{R}^2 = (x - x_0)^2 + (y - y_0)^2 + (z + z_0)^2$ i.e. R denotes the distance between the source point and the field point of observation.

3.5 Transient fields

Deriving transient field expressions requires time to be added. The Green's function must then be defined in a more general form to satisfy the boundary and initial conditions at hand. It should not be restricted to harmonic wave excitation only.
The time dependent function is seen to be the inverse Fourier transform of the harmonic Green's function.

$$\begin{aligned} g(\vec{r}; \vec{r_0}; \tau) &= \mathcal{F}^{-1}\{g(\vec{r}; \vec{r_0}; \omega)\} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{4\pi R} e^{jkR} e^{-j\omega\tau} d\omega \\ &= \frac{1}{4\pi R} \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega(\tau - \frac{R}{c})} d\omega \\ &= \frac{1}{4\pi R} \delta\left(\tau - \frac{R}{c}\right), \end{aligned} \tag{3.28}$$

where $k = \omega/c$ is used. The time-dependent (free-space) Green's function is then written [40][5]

$$g_0(\vec{r}, t | \vec{r_0}, t_0) = \frac{1}{4\pi R} \delta\left(t - t_0 - \frac{R}{c}\right) \tag{3.29}$$

This function is a solution to the inhomogeneous wave equation

$$\nabla^2 g - \frac{1}{c^2} \frac{\partial^2 g}{\partial t^2} = -\delta(\vec{r} - \vec{r_0}) \delta(t - t_0) \quad , \tag{3.30}$$

now considering a pulse-wave from a point source located at $\vec{r_0}$ and at time t_0. Note that from the delta function in Eq. 3.29 it is seen that this function is zero until $t = (t_0 + R/c)$ i.e. the time for the pulse to reach the point $\vec{r}$. In fact it is nonzero only at this particular value of t.
The image source functions in Section 3.4.2 can be expressed as time-dependent Green's functions in a way similar to what here is done for the free-space function. Utilization of the time dependent Green's function leads to the transient solution integrals of the next section.

[5] The time dependent Green's function may also be written $g_0(\vec{r} - \vec{r_0} | t - t_0)$ and its sifting property is maybe clearer then.

3.6 Solution integrals

A rigorous solution for the acoustic pressure field when considering harmonic (radiated) waves is described by the *Helmholtz-Kirchhoff integral formula*. A derivation of this formula regarding optical disturbances is given in [14] and for acoustics in [46]. In this text a slightly different approach, as in [40], will be taken.
The equation $\nabla^2 p(\vec{r},\omega) + k^2 p(\vec{r},\omega) = -F(\vec{r},\omega)$ will be solved for the spatial factor of a pressure wave from a simple harmonic source $F(\vec{r},\omega)$. First we multiply the previous Helmholtz equation with $G(\vec{r},\vec{r_0}|\omega)$ and Eq. 3.21 with $p(\vec{r},\omega)$. Next $\vec{r}$ and $\vec{r_0}$ are interchanged and the symmetry of $G(\vec{r},\vec{r_0}|\omega)$ and $\delta(\vec{r}-\vec{r_0})$ are utilized. An integration over the volume will then lead to the formula

$$\iiint \left[G(\vec{r},\vec{r_0};\omega)\nabla_0^2 p(\vec{r_0};\omega) - p(\vec{r_0};\omega)\nabla_0^2 G(\vec{r},\vec{r_0};\omega)\right] dV_0 \quad = \quad \iiint p(\vec{r_0};\omega)\delta(\vec{r}-\vec{r_0};\omega)dV_0 - \iiint F(\vec{r_0};\omega)G(\vec{r},\vec{r_0};\omega)dV_0$$

The zero indicate derivation and integration with respect to the source coordinates.
Then by using Green's second identity (Eq. 3.19) and the sifting property of the delta function, we arrive at the solution integrals

$$\begin{aligned} p(\vec{r};\omega) &= \iiint F(\vec{r};\omega)G(\vec{r},\vec{r_0};\omega)dV_0 \\ &+ \iint_{S_0} \left[G(\vec{r},\vec{r_0};\omega)\frac{\partial p(\vec{r_0};\omega)}{\partial n_0} - p(\vec{r_0};\omega)\frac{\partial G(\vec{r},\vec{r_0};\omega)}{\partial n_0}\right] dS_0 \end{aligned} \quad (3.31)$$

The total field at $\vec{r}$ is thus the summation of the field from all elementary sources $F(\vec{r_0},\omega)$ plus the contribution from the boundary surface. Eq. 3.31 is the Kirchhoff formula for the Helmholtz equation and the latter integral (the surface integral) is recognized as the Helmholtz-Kirchhoff diffraction integral.
In case of a source located inside an unbounded medium radiating outwards, the surface integral vanishes. On the other hand if we have a source located in a part of the surface and no source inside the medium, the volume integral will disappear. Assumptions implicit in Eq. 3.31 are that the wave propagates in a linear, nonattenuating and homogeneous medium.

A similar formula can be derived in terms of a more general wave excitation that is not restricted to the harmonic case. In other words we search for a solution to Eq. 3.14 where we will denote the source term as $f(\vec{r_0},t)$ and use a general, time dependent Green's function $g(\vec{r},\vec{r_0}|t,t_0)$. This Green's function is a solution to Eq. 3.30 i.e. the wave equation with a space-time impulse source.
The derivation of the general formula is quite similar to the harmonic case. One has to assume that $\vec{r}$ is in V for the δ-function to be nonzero

and make assumptions regarding e.g. the deacying properties of the temporal terms. The time dependent Kirchhoff formula is written

$$\begin{aligned} p(\vec{r};t) &= \iiint_V \int_{-\infty}^{\infty} g(\vec{r},\vec{r_0}|t,t_0) f(\vec{r};t_0) dV_0 \\ &+ \iint_{S_0} \int_{-\infty}^{\infty} \left(g(\vec{r},\vec{r_0}|t,t_0) \frac{\partial p(\vec{r_0};t_0)}{\partial n_0} \right. \\ &- \left. p(\vec{r_0};t_0) \frac{\partial g(\vec{r},\vec{r_0}|t,t_0)}{\partial n_0} \right) dS_0 dt_0 \end{aligned} \tag{3.32}$$

The total field is the sum of contributions from sources inside the volume and sources outside the volume.

3.6.1 Acoustically rigid baffle

Remember that in this case the normal component of the velocity, v_n is zero on the boundary (Section 3.4). The appropriate Green's function is $g_+(\vec{r},\vec{r_0};\omega)$ in Eq 3.26. Its normal derivative is zero at the boundary i.e. $\partial g_+/\partial n = 0$ at $z = 0$.

$$\begin{aligned} \frac{\partial g_+}{\partial n} &= \frac{\partial}{\partial n}\left[\frac{1}{4\pi R} e^{jkR} + \frac{1}{4\pi \hat{R}} e^{jk\hat{R}}\right] \\ &\Updownarrow \quad R = \hat{R} \\ &= \frac{1}{4\pi R} e^{jkR} \left(jk - \frac{1}{R}\right) (\cos(\varphi) + \cos(\hat{\varphi}) \\ &\Updownarrow \quad \cos(\varphi) = -\cos(\hat{\varphi}) \\ \frac{\partial g_+}{\partial n} &= 0 \\ g_+(\vec{r},\vec{r_0};\omega) &= \frac{1}{4\pi R} e^{jkR} + \frac{1}{4\pi \hat{R}} e^{jk\hat{R}} \\ &\Updownarrow \quad R = \hat{R} \\ g_+(\vec{r},\vec{r_0};\omega) &= 2 \cdot \frac{1}{4\pi \hat{R}} e^{jk\hat{R}} = 2g_0(\vec{r},\vec{r_0};\omega) \end{aligned}$$

For the time dependent case the Green's function will be (see Sec. 3.5)

$$g_+(\vec{r},\vec{r_0}|t,t_0) = 2g_0(\vec{r},\vec{r_0}|t,t_0) \tag{3.33}$$

The two latter equations signify that the source and its image has coalesced to form a source of double strength. Because of the vanishing derivative of the Green's function the resulting integral only includes the normal derivative of p and therefore only the first term of the surface integral in Eq. 3.31 applies

$$p(\vec{r};\omega) = \iint_{S_0} \frac{\partial}{\partial n} p(\vec{r_0};\omega) g_+(\vec{r},\vec{r_0};\omega)\, dS_0$$

Expanding this solution to include time and at the same time using the transient form of $g_0(\vec{r},\vec{r_0};\omega)$ i.e. $g_0(\vec{r},\vec{r_0};t)$ gives

$$\begin{aligned} p(\vec{r},t) &= 2\iint_{S_0}\int_{-\infty}^{\infty}\frac{\partial}{\partial n}p(\vec{r_0},\tau)g_0(\vec{r},t|\vec{r_0},\tau)\,d\tau\,dS_0 \\ &= 2\iint_{S_0}\int_{-\infty}^{\infty}\frac{\partial}{\partial n}p(\vec{r_0},\tau)\frac{1}{2\pi R}\delta(t-\tau-R/c)\,d\tau\,dS_0 \\ &\Updownarrow \quad \text{applying the sifting property of the } \delta\text{-function} \\ p(\vec{r},t) &= \iint_{S_0}\frac{\frac{\partial}{\partial n}p(\vec{r_0},t-R/c)}{2\pi R}\,dS_0 \end{aligned} \tag{3.34}$$

This equation is known as the ***Rayleigh integral***. It may be written equivalently by using Eq. 3.6 as

$$p(\vec{r},t) = -\iint_{S_0}\rho\frac{\frac{\partial}{\partial t}v_n(\vec{r_0},t-R/c)}{2\pi R}\,dS_0 \tag{3.35}$$

and by using the velocity potential of Eq. 3.10 it can be written

$$\Phi(\vec{r},t) = \iint_{S_0}\frac{v_n(\vec{r_0},t-R/c)}{2\pi R}\,dS_0 \tag{3.36}$$

The obliquity factor (see the next section) is $\alpha_{RS} = 1$. The Rayleigh integral seems to be used most often in acoustics. At least when judged by its appearance in acoustical textbooks e.g. [40].

3.6.2 Acoustically soft baffle

In this case the pressure p is zero at the surface. The appropriate Green's function is then $g_-(\vec{r},\vec{r_0};\omega)$ (Eq. 3.27).

$$\begin{aligned} \frac{\partial g_-}{\partial n} &= \frac{\partial}{\partial n}\left[\frac{1}{4\pi R}e^{\,jkR} - \frac{1}{4\pi\hat{R}}e^{\,jk\hat{R}}\right] \\ &\Updownarrow \quad R=\hat{R} \\ &= \frac{1}{4\pi R}e^{\,jkR}\left(jk-\frac{1}{R}\right)\Big(\cos(\varphi)-\cos(\hat{\varphi})\Big) \\ &\Updownarrow \quad \cos(\varphi) = -\cos(\hat{\varphi}) \\ \frac{\partial g_-}{\partial n} &= 2\cos\varphi\left(jk-\frac{1}{R}\right)\frac{1}{4\pi R}e^{\,jkR} \\ g_-(\vec{r},\vec{r_0};\omega) &= 0 \end{aligned}$$

In this case the function itself $g_-(\vec{r},\vec{r_0};\omega)$ disappears on the boundary i.e. $g_- \equiv 0$ and only the second term in the surface integral of Eq. 3.31 applies. We then have that

$$\begin{aligned} p(\vec{r};\omega) &= \iint_{S_0} -p(\vec{r_0};\omega)\frac{\partial G(\vec{r},\vec{r_0};\omega)}{\partial n_0}dS_0 \\ &= -\iint_{S_0} p(\vec{r_0};\omega)2\cos\varphi\left(jk-\frac{1}{R}\right)\frac{1}{4\pi R}e^{\,jkR}dS_0 \end{aligned}$$

By assuming $R = |\vec{r} - \vec{r_0}| \gg 1/k$ we can omit the $1/R$ term. In addition, from [10] we have $\frac{\partial p(\vec{r_0};\omega)}{\partial n} = -jkp(\vec{r_0};\omega)e^{j(\omega t - kR_n)}$, and by using this we get

$$p(\vec{r};\omega) = \iint_{S_0} \frac{\frac{\partial}{\partial n}p(\vec{r_0};\omega)}{2\pi R}\cos\varphi \quad dS_0 \tag{3.37}$$

The equivalent transient form of this equation is given by [14]

$$p(\vec{r},t) = \iint_{S_0} \frac{\frac{\partial}{\partial n}p(\vec{r_0},t - R/c)}{2\pi R}\cdot\cos\varphi \quad dS_0 \tag{3.38}$$

This formula is called the ***Rayleigh-Sommerfeld integral*** and is equivalent to the Rayleigh integral except for the obliquity factor $\alpha_{RS} = \cos\varphi$.

3.6.3 Source in free space

In this case we have no influence from the boundary and the latter part of Eq. 3.31 disappears. The closing surface in an practically unbounded medium has no influence due to Sommerfeld's radiation condition. By applying the free space Green's function and do derivations similar to the two preceding procedures, a new solution formula arise. This formula is known as the ***Kirchhoff integral*** and its transient form is written

$$p(\vec{r},t) = \iint_{S_0} \frac{\frac{\partial}{\partial n}p(\vec{r_0},t - R/c)}{2\pi R}\cdot\frac{1+\cos\varphi}{2}dS_0 \tag{3.39}$$

Note that only the obliquity factor $\alpha_K = \frac{1+\cos\varphi}{2}$ is different from the Rayleigh integral. This solution is used particularly in the field of optics.

3.7 Calculating the transient acoustic field

So far three particular solutions to the acoustical problem regarding three different boundary conditions have been discussed. Calculation of these solutions and hence simulations of the transient field will now dicussed briefly.

First, the similarities between the solutions are easily observed. Only in the directivity (i.e. obliquity) factor, α, they differ. It is shown in Figure 3.3 that the deviations between the directivity factors are minor when considering solutions at narrow angles. The deviation between α_R and α_{RS} is less than 5% at angles $\pm 18°$ and the deviation between α_R and α_K is less than 5% for angles at about $\pm 25°$.
The effect the obliquity factor has on the accuracy of the calculations depends on the deviation between the impedance of the transducer and the

assumed boundary for the chosen integral. The Rayleigh integral may be an acceptable approximation when the point of observation lies close to perpendicular to any point of the source (i.e. $\alpha \approx 1$ and is herefore ignored).
According to Kino[27], a transducer element (like one in an ultrasound array) has an response between the rigid baffle case and the pressure release baffle.

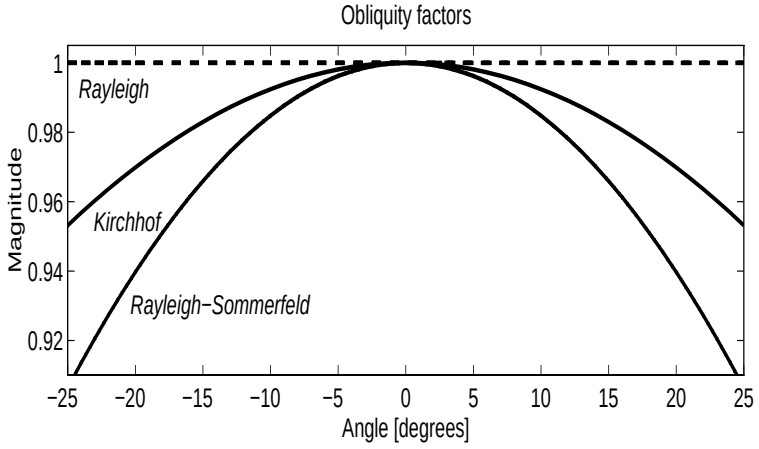


Figure 3.3: The obliquity factors are plotted for angles from $-15°$ to $15°$ off-axis. The deviation is observed to be minor when we are close to the axis.

The geometrical shape of the transducer is crucial for the calculation of the field. Planar transducers will lead to simplified expressions. Other important parameters that affect the field are the excitation and the characteristics of the media.
Further simplifications that often are utilized, comes from near-field and far-field assumptions. The near-field is often called the *Fresnel region*. By using the Fresnel approximation one assumes that the phase of the wavefront in the aperture has a quadratic dependence on aperture coordinates [14]. Accepting the validity of the Fresnel approximation reduces the solution formula to a two-dimensional convolution [14].
In the far-field the waves are considered to be planar waves. This region is called the *Fraunhofer region* and starts approximately at a distance farther from the transducer than $D^2/2\lambda$, where D is the maximum linear dimension of the aperture. By using the stronger Fraunhofer approximation, the solution formula simplifies to a two-dimensional Fourier transform of the wavefield in the aperture [14].

3.7.1 Numerical integration

A robust way of computing the field is to perform the integration of the formulas e.g. the Rayleigh integral of Section 3.6, numerically. However

this will often be a laborious task since a high number of operations is required to compute the whole field. Each field point will be the sum of contributions from every transducer element. In Fig. 3.4 we see that in order to calculate the field at the observation point, we need to compute the contributions from all elements and add them together.
If we want to calculate the field over an observation plane $z = z_{observation}$, then we must repeat this procedure for every point in this plane. Making a 3D plot of the field requires the calculations to be repeated for selected values of z.
The main advantage with this method is its simplicity and generality. It can be used to calculate the field from transducers of any shape. A similar approach is taken in the UltraSim program [19].

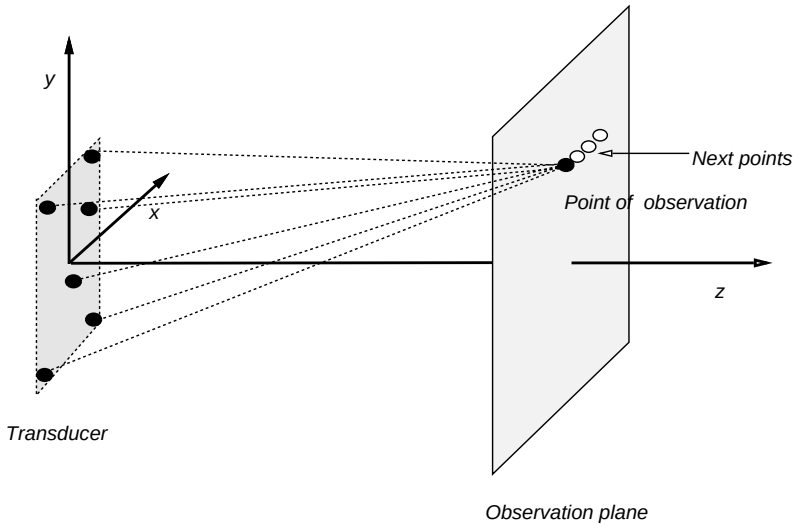


Figure 3.4: This is an attempt to visualize that points in the aperture all make a contribution to the field at the observation point (only a few points in the aperture is drawn).

3.7.2 Angular spectrum method

Only very briefly we will mention some terms of this method.
The angular spectrum, or Fourier decomposition method was originally developed in the study of optical diffraction [14]. It has later also been applied to acoustical wave propagation and has turned out to be a powerful tool in that manner [43]. With this method a pressure distribution over a plane surface is decomposed into an equivalent 2D spectrum of plane waves. Propagation from one plane to another parallel plane is modelled by multiplication of the spectral component with the appropriate phase factor. Thus by adding the contributions from the plane waves and consider the phase shifts they have undergone during the propagation to the point of observation, we can calculate the field amplitude in

every point of interest. The efficiency of the method is due to its numerical implementation with the 2D FFT.
Assuming a wave incident on the xy-plane i.e. the source decomposition plane is located at $z = 0$, then the angular spectrum decomposition is expressed as a spatial 2D Fourier transform

$$A_0(f_x, f_y) = \iint_{-\infty}^{\infty} U(x, y, 0) e^{-j\,2\pi(f_x x + f_y y)} dx\, dy \tag{3.40}$$

f_x and f_y are the spatial frequencies in x- and y- directions respectively. The wave vectors then are $k_x = 2\pi f_x$,$k_y = 2\pi f_y$ etc. and propagate in the direction $\vec{k}$. There is a continuous distribution of plane waves arising from the decomposition and this is the *angular spectrum.*
The further discussion on this method will be omitted, for further details the reader can turn to the excellent texts of e.g. [14] and [43].

3.7.3 Impulse response method

A powerful technique of evaluating the formulas of Section 3.6 is the ***impulse response method*** [53]. It is also referred to as a convolution integral method. We will use the Rayleigh integral (Eq. 3.36) when giving a brief summary of the principles of this method. Thus the velocity potential in a spatial and temporal point is determined from

$$\Phi(\vec{r}, t) = \iint_{S_0} \frac{v_n(\vec{r_0}, t - R/c)}{2\pi R}\, dS_0$$

The velocity function can be expressed as a temporal integral term

$$v_n(t - R/c) = \int v_n(t_0)\delta(t - R/c - t_0) dt_0 \tag{3.41}$$

This integral is inserted into Eq. 3.36 and by exchanging the order of integration we get a new equation

$$\Phi(\vec{r}, t) = \int v_n(t_0) \int \frac{\delta(t - R/c - t_0)}{2\pi R} dS_0 dt_0 \tag{3.42}$$

Then the impulse response $h(\vec{r}, t)$ is defined as

$$h(\vec{r}, t) = \int \frac{\delta(t - R/c)}{2\pi R} dS_0 \tag{3.43}$$

In every instant $h(\vec{r}, t)$ mainly gives the amplitude of the contribution from every point on the source that is equidistant from the point of observation. The impulse response function is thus the time dependent velocity potential at a spatial point resulting from an impulse velocity of a transducer of any shape. From Equations 3.42 and 3.43 we note that the velocity potential can be expressed as a convolution process.

$$\Phi(\vec{r}, t) = v_n(t) * h(\vec{r}, t) \tag{3.44}$$

Therefore, when the velocity potential is determined, the pressure is easily calculated by Eq. 3.10. Alternatively, we can express the convolution in terms of the pressure

$$p(\vec{r},t) = -\rho \frac{\partial v_n(t)}{\partial t} * h(\vec{r},t) \tag{3.45}$$

The convolution integral is a quite simple integral to evaluate numerically. Calculation of the field originating from an aperture is then a matter of evaluating the impulse response for its geometry considered. Once $h(\vec{r},t)$ has been determined, the following convolution integral has to be evaluated

$$p(\vec{r},t) = -\rho \int_{t_1}^{t_2} \frac{\partial v_n(t-\tau)}{\partial t} h(\vec{r},\tau) d\tau \tag{3.46}$$

Here t_1 denotes the minimum time of arrival from the source to the point of observation and t_2 the maximum time of arrival. This is clearly a one dimensional integral in contrast to the two dimensional integral of Eq. 3.36.
Analytic solutions of $h(\vec{r},t)$ have been derived for several geometries. For example plane circular piston in [53, 54] and for a curved radiator in [45]. It is not always possible to find closed form analytic expressions for the impulse response in case of more complex geometries.

However, a variation of this method has been proposed in [22] that can simulate transducers with *'any transducer apodization and excitation'*. Two assumptions have been made there.
First a large and slightly curved transducer is assumed. Secondly, a separability between the excitation and the transducer geometry is assumed. Thus the surface vibration is split into a spatial component and a temporal component.
A simulation method is proposed that emit a spherical wave from a point on the aperture and then sum all spherical waves at the field point. The waves are weighted by the inverse of the distance between the aperture point and the field point.
The transducer surface is divided into small rectangles and the position and orientation of these are first calculated in the simulation program. The responses from these rectangles are then added and the field is calculated.
Utilization of simulation methods based on the impulse response method is seemingly widespread for calculation of the acoustic field, due to the methods computational efficiency.

3.8 Summary

In this chapter some basics of acoustics have been reviewed. Special concern has been devoted to three particular boundary cases and to the

non-harmonic excitation of apertures. An equation for calculation of the transient field for a rigid baffled transducer has been derived more thoroughly than the soft baffle and free-space cases.
Several assumptions are useful when calculation of an acoustic field is desired.
Finally, some attention has been made to methods for calculation or simulation of transient acoustic fields. The simulation of ultrasound fields is often a necessary tool used when designing new transducer geometries.

Chapter 4

Beampatterns by the Coarray method

Array systems are used to detect signals conveyed by propagating waves and thus sample wavefields spatially. A large collection of distinct array processing algorithms allow us to steer and focus arrays in particular directions, thereby providing directional information of the unfolding signals. These algorithms are called beamformers. In this text only conventional beaformer algorithms are taken into consideration.
Signals received by the sensor elements are used in a phased manner in order to preferencially detect signals from a desired direction. Beamforming can be regarded as attempting to estimate the wavenumber frequency spectrum of the wave field and is thus an analogy to bandpass filtering in case of 1-D spectral analysis [23]. The beampatterns disclose the array's ability to focus on a given signal (a *beam*) while discarding noise signals from other directions.
The objective when designing a beamformer is to make the beampattern as close to an impulse as possible. As we are using only a finite number of sensors we know that there must be a discrepancy between the objective and the physical reality, and that this portion is highly dependent on the number of sensors and the area that the aperture spans.
In this chapter the coarray is seen to offer an alternative way of calculating the beampattern, under given circumstances, and discussions on the developed coarray calculation and visualization tools are given in the last section of this chapter. Some discussion on different element configurations of arrays have also been included in this chapter
However, let us first take a look at some sampling relations that are valid for propagating signals. In other words, relations valid for spatio-temporal signals.

4.1 Simultaneous time and space sampling

The array systems sample the wave field in both time and space. This puts limitations on the sampling intervals in both domains. From digital signal processing we know that in order to avoid temporal frequency aliasing, we have to sample at least twice as fast as the fastest variation of the signal (the *Nyquist rate*).
If the highest frequency component of a time-space signal, $s(\vec{x},t)$, is ω_0, a sampling frequency, $\omega_s \geq 2\omega_0$ must be chosen. Thus the time between each sample is at most

$$T = \frac{2\pi}{\omega_s} \leq \frac{\pi}{\omega_0} \tag{4.1}$$

The input signal to a system with sampling frequency, ω_s, must then be band limited to frequencies below one-half the sampling frequency. Undersampling appears when we sample too slow for the frequency content of the signal. In an undersampled system, the Nyquist sampling theorem states that frequencies in the continuous signal above the Nyquist rate, $\omega_s/2$, will appear as frequencies below this rate. This is called *aliasing*.
The Nyquist criterion also applies to spatial sampling. This implies that in order to avoid spatial aliasing, the spatial sampling frequency, k_s, must be at least two times the highest component k_0 of the time-space signal, $s(\vec{x},t)$ i.e. $k_s \geq 2k_0$.
In space we sample with sensors separated by d meters. Consequently, the spatial sampling interval must be

$$d = \frac{2\pi}{k_s} \leq \frac{\pi}{k_0} \tag{4.2}$$

For propagating signals, the relation between wavenumber and frequency is (for non-dispersive media) $k = \frac{\omega}{c}$. Thus a temporal band limited propagating signal implies that the signal is spatially limited as well.
Combining the sampling relations above, yields the coupling between temporal and spatial sampling

$$d = \frac{\pi}{k_0} = \frac{\pi}{\frac{\omega_0}{c}} = c \cdot \frac{\pi}{\omega_0} = \frac{cT}{2} = \frac{\lambda}{2} \tag{4.3}$$

The spatial sampling characteristics are determined by the inter sensor spacing, and they are therefore fixed once the array is made. We are now assuming that the array spacings cannot be dynamically altered by connecting, disconnecting or displacing array elements during or between operations (as is the case for e.g. very large arrays (VLAs)).
The temporal sampling properties will be determined by the electronics e.g. the time delays between each sample. Because of the close relationship between the sampling in the two domains, the risk of undersampling in one domain while oversampling in the other, exists. Undersampling in the spatial domain will give grating lobes in the visible

region. Signals propagating in directions corresponding to the grating lobes cannot be distinguished from the signal with direction corresponding to the mainlobe.
The relation between the magnitude of the wavenumber vector in the x-direction and the angle of incident[1] is written

$$k_x = -k \sin\phi = \frac{-2\pi}{\lambda} \sin\phi \tag{4.4}$$

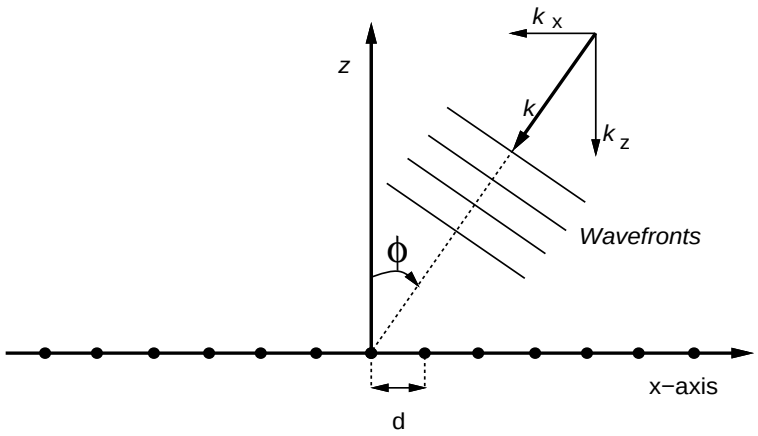


Figure 4.1: Illustrations of a propagating signal impinging on a linear array with an azimuth angle ϕ. The directions of the wavenumber components are indicated by drawing them as basis vectors.

Since $|\sin\phi| \leq 1$, k_x can be real only between $\pm 2\pi/\lambda$. The region that k_x span when $|k_x| \leq 2\pi/\lambda$ is the called the *visible region*, and corresponds to angles between $\pm 90°$ for a given value of λ.
A linear array like the one in Fig. 4.1 with uniform weighting, will have an array pattern

$$W(k_x) = \frac{sin(\frac{M}{2} k_x d)}{sin(\frac{1}{2} k_x d)} \tag{4.5}$$

It is easily observed that this function is periodic with period $k_x = 2\pi/d$ and the spatial frequency, $k_x = \pi/d$, is often called the *folding* frequency. We recognize this frequency as the limit of the spatial sampling interval (Eq. 4.2). Consequently, the sampling characteristics will be determined by the inter-sensor spacing and the design wavelength.
The array pattern for a linear array can also be expressed in terms of the azimuth angle, ϕ i.e. the angle between the incident wave propagation direction and the normal to the plane. Simple geometrical considerations show that $k_x = -k \sin\phi$. Hence we can write the array pattern in terms of the angle

$$W(\phi) = \frac{sin(\frac{kMd}{2} sin(\phi))}{sin(\frac{kd}{2} sin(\phi))} \tag{4.6}$$

[1] this is the azimuth angle in case of a linear array

Let us now assume that we have chosen a inter-element spacing of $d = \lambda$. A grating lobe will then arise at $\phi = \pm 90°$, since $\sin\phi_g = \pm\frac{\lambda}{d} = \pm 1$. Propagating waves impinging normal to the array, can then not be distinguished from the waves coming from the x- or $-x$-direction. If we increase the size of d, stronger undersampling results and we introduce grating lobes at angles closer than $90°$.
By choosing $d \leq \lambda/2$ oversampling occurs and the k_x-values up to the folding frequency span a wider field than $\phi = 90°$. The spatial frequencies above $2\pi/\lambda$ define the *invisible region*. No propagating signals energy can appear hear, but noise signals can contribute to a calculated spectrum. Algorithms that assume no energy in this region may then be confused.

4.2 Fundamentals of coarrays

The term coarray was first introduced by Haubrich [17] apparently as a short name for 'correlation array'. He discussed array design in case of incoherent radiation by means of getting high resolution. In this text we will not restrict our examination to discussions on getting the minimum beamwidth but will also consider minimization of the sidelobe level.
Primarily the discussion is with respect to incoherent radiation. Nevertheless, some basics for the coarray with respect to coherent imaging will also be reviewed, where the term coarray might be regarded as short for 'convolution array'. With incoherent imaging, is meant the process of measuring a field, originating from an emitting source in space, over an aperture.
Using an aperture to transmit radiation and next measure the radiation reflected from objects in space over a receive aperture, is referred to as coherent imaging. Examples of the latter are radar and ultrasound.
The discussions in the subsequent introduction to coarrays are primarily with respect to linear arrays. Later two-dimensional (planar) arrays (Sec. 4.4) will be discussed, and differences and similarities compared to linear arrays will hopefully become clearer than.

4.2.1 The difference coarray

For *incoherent imaging*, the coarray is defined as the set of all vector spacings [2] between points in the aperture. Formally it is defined [18]:

$$C_d(A_1, A_2) \triangleq \{y \mid y = x_1 - x_2 \quad \text{for} \quad x_1 \in A_1, x_2 \in A_2\} \tag{4.7}$$

where A_1 and A_2 are the sets of position vectors for elements in *two* receive apertures. This coarray will be referred to as the *difference coar-*

[2] often called baselines

ray. It can also be denoted the *morphological autocorrelation*[18] of A because most frequently we have $A_1 = A_2 = A$ and the operation is equivalent to correlate A with itself. The term morphological denotes calculation with respect to the spatial shape and structure of the arrays or apertures.
For discrete arrays this correlation process will be a discrete sum over all element positions, while a continuous aperture requires integration over the aperture area. The coarray for a continuous aperture is defined

$$c(\vec{\chi}) = \int_A w(\vec{x})w(\vec{x}+\vec{\chi})d\vec{x} \tag{4.8}$$

A definition of the discrete coarray function is [23]

$$\begin{aligned} c(\vec{\chi}_m) &= \sum_{(a,b)\in\vartheta(\vec{\chi})} w_a w_b^* \quad , \quad m = 1,2,\ldots,M_d \\ \vartheta(\vec{\chi}) &= \left\{(a,b)|\vec{\chi}_m = \vec{x}_a - \vec{x}_b\right\}_{m=1}^{M_d} \end{aligned} \tag{4.9}$$

Here $\vec{\chi}$ denote the particular lag or baseline and $\vartheta(\vec{\chi})$ the set of indices (a,b) given by the pairwise differences of sensor locations. The value of the coarray at a particular lag will be the sum of the products of $w_a w_b$ that constitutes this lag i.e. the sum of the weight contributions from equal baselines. The coarray values vary between zero (no baseline of this particular length) and N, the number of elements in the array. The coarray value for lag zero is always N for difference coarrays, unless we have applied a nonuniform apodization to the array.
In that case the maximum lag value at lag zero is equal to the sum of the squared sensor weights i.e.

$$c(\vec{0}) = \sum_{(a,a)\in\vartheta(\vec{0})} (w_a)^2$$

The maximum number of elements in a coarray appears when there are no repeated vector differences in the array (i.e. no redundancy). This is equivalent to maximization of the resolution. In that case the maximum number of distinct elements in the difference coarray is

$$M_d = N(N-1)+1 \tag{4.10}$$

The *redundancy ratio*[3] is given as [5]

$$\mathcal{R} = \frac{N(N-1)}{2L} \tag{4.11}$$

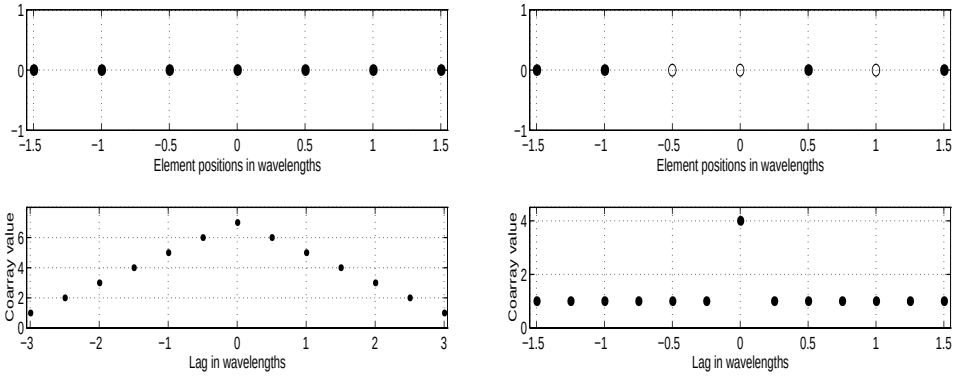
Here N is the number of sensors in the array and L is the length of the array (the aperture). For an array with N equally spaced sensors the length will be $L = N-1$.
It should be noted that some other authors (e.g.[3, 39]) denote the array length by the equivalent *greatest multiple of unit spacing* and that

[3] Sometimes called the redundancy degree or redundancy factor. In some articles it appears in a 'upside down' form.

Eq. 4.11 is the inverse of the quality ratios in some articles e.g. in [17, 23]. This property is thus expressed in a variety of ways. However, its purpose is to quantify a reference grid so that the various arrays can be compared with respect to the degree of sparseness.
The aperture of an equi-spaced array can be determined from [33]

$$A = \frac{N(N-1)}{2} - R + H \tag{4.12}$$

where R and H denote the number of redundancies and holes in the coarray respectively, counted for positive lags. This formula will for an equi-spaced array be equal to the array length, L.
A redundancy in the coarray appears when there are more then one pair of sensors separated by this particular distance. Holes appears when there are no pair of sensors separated by a given distance. If there are neither holes nor redundancies, the array is called a *perfect* array (see Fig. 4.2). There exists no perfect linear arrays with more than four sensors.



(a) Uniform array

(b) Perfect array

Figure 4.2: A filled regular linear transducer array giving a difference coarray with redundant lags but no holes is given in (a). In (b) the largest perfect linear array from an underlying regular grid is shown. The circles denotes missing elements in the grid. The minimum grid spacing is $d = \lambda/2$ in both arrays.

The redundancy factor thus shows how well one has succeeded in making a perfect array. Designing arrays with minimum redundancy is motivated by the desire to approximate a perfect array. These arrays are characterized by high resolution, but suffer from a high sidelobe level that limit their usefulness in applications where noise signal affects the field.
It has been shown [3] that for minimum redundancy arrays the limit of the redundancy factor is $4/3$. Minimum redundancy arrays have a coarray of maximum length, with the constraint that there exists no missing

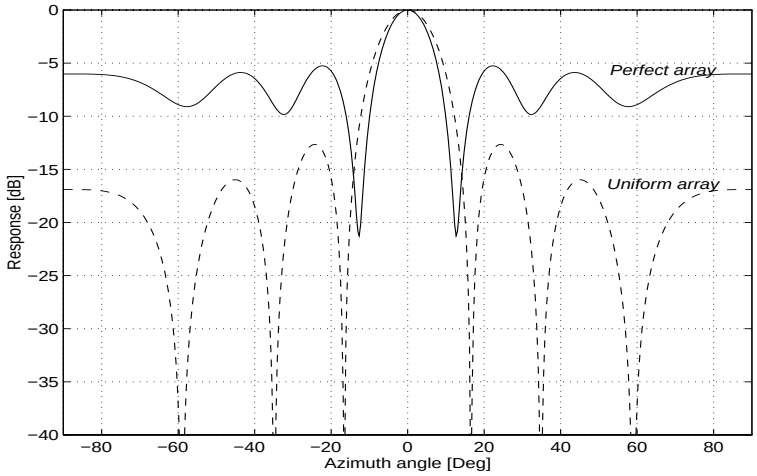


Figure 4.3: The angular responses of the arrays in figure 4.2 are calculated from their coarray values.

lags. The coarray values are required to be $C_d \geq 1$ (see examples in [33]). A nonredundant array is the coarray of greatest length without any redundancies. In other words we allow missing lags to exist (see examples in [23]) and their coarrays have values limited to the interval $C_d \in [0, 1]$. Both minimum redundancy linear arrays (MRLA) and minimum missing lag (nonredundant) arrays give maximum resolution for a given aperture. The largest perfect array also is the largest array where a MRLA resembles a nonredundant array with the same number of elements. For all arrays with more than four elements, the nonredundant array will span a larger aperture than the MRLA. Consequently, it will also span the largest region in the lag-space and hence give the best resolution.

4.2.2 The sum coarray

For *coherent imaging* the coarray is the set of vector sums of all point location vectors in the aperture [18]

$$C_s(A_T, A_R) \triangleq \{y \mid y = x_1 + x_2 \quad \text{for} \quad x_1 \in A_T, x_j \in A_R\} \tag{4.13}$$

where A_T represents the transmit aperture and A_R the receive aperture. In the further writing it will be denoted the *sum coarray*. It may as well be referred to as the *morphological convolution* of set A_T and A_R because of its equivalence to a convolution between the two arrays. With discrete arrays this implies a convolution sum, while a continuous array will demand a convolution integral. Identical receive and transmit aperture gives an auto-convolution.

The coarray for a continuous aperture is defined

$$c(\vec{\chi}) = \int_A w(\vec{x})w(\vec{x} - \vec{\chi})d\vec{x} \tag{4.14}$$

A definition of the discrete coarray function is [18]

$$\begin{aligned} c(\vec{\chi}_m) &= \sum_{(a,b)\in\Psi(\vec{\chi})} w_a w_b \quad , \quad m = 1, 2, \ldots, M_s \\ \Psi(\vec{\chi}) &= \left\{(a,b)|\vec{\chi}_m = \vec{x}_a + \vec{x}_b\right\}_{m=1}^{M_s} \end{aligned} \tag{4.15}$$

Here $\vec{\chi}$ denote the particular sum-lag or baseline and $\Psi(\vec{\chi})$ the set of indices (a, b) given by the pairwise sum of sensor locations. M_s is the number of lags in the sum coarray.
The maximum number of elements in a sum coarray appears when there are no repeated vector sums in the array (i.e. no redundancy). In case of nonredundant arrays the maximum number of distinct elements for the sum coarray is[18]

$$M_s = \frac{N(N+1)}{2} \tag{4.16}$$

4.2.3 Comparison between the two types of coarrays

When compared to Eq. 4.10, the latter equation signifies that the sum coarray will contain fewer points than the difference coarray in nonredundant arrays. To the author's knowledge there has been no thorough investigation on arrays with minimum redundancy in the sum coarray. The sum and difference coarrays will be identical for some arrays. With linear arrays this happens when the arrays are equal and symmetric with respect to origin ($x = 0$ on the x-axis).
Two-dimensional arrays with equivalent difference- and sum coarrays exist. For a planar (2D) array this will be the case when the arrays are equal and symmetric with respect to both the x- and y-axis. This should be obvious since convolution implies a rotation of one of the arrays around both axes before correlating the arrays (i.e. before the element weights are multiplied and summed up). When the coarray is symmetric to both axis, a rotation of the array has no influence on the array and the convolution operation is equivalent to the correlation of the arrays.
It should also be noted from the definitions that the sum coarray is sensitive to the location on the aperture plane, while the difference coarray is not. Moreover, the sum coarray has lags farther from the origin than the difference coarray has.
An important point about coarrays that should be emphasized is that different aperture geometries may share the same lag-space. When the coarray lag-spaces cannot be told apart, the apertures are said to be *coarray equivalent*. It should be observed that the coarray values of coarray

equivalent arrays will be distinct. Their equivalence is with respect to the spatial extension and shape in the lag domain. This is observed in the succeeding figure where coarray equivalent two-dimensional arrays are plotted (Fig. 4.4).

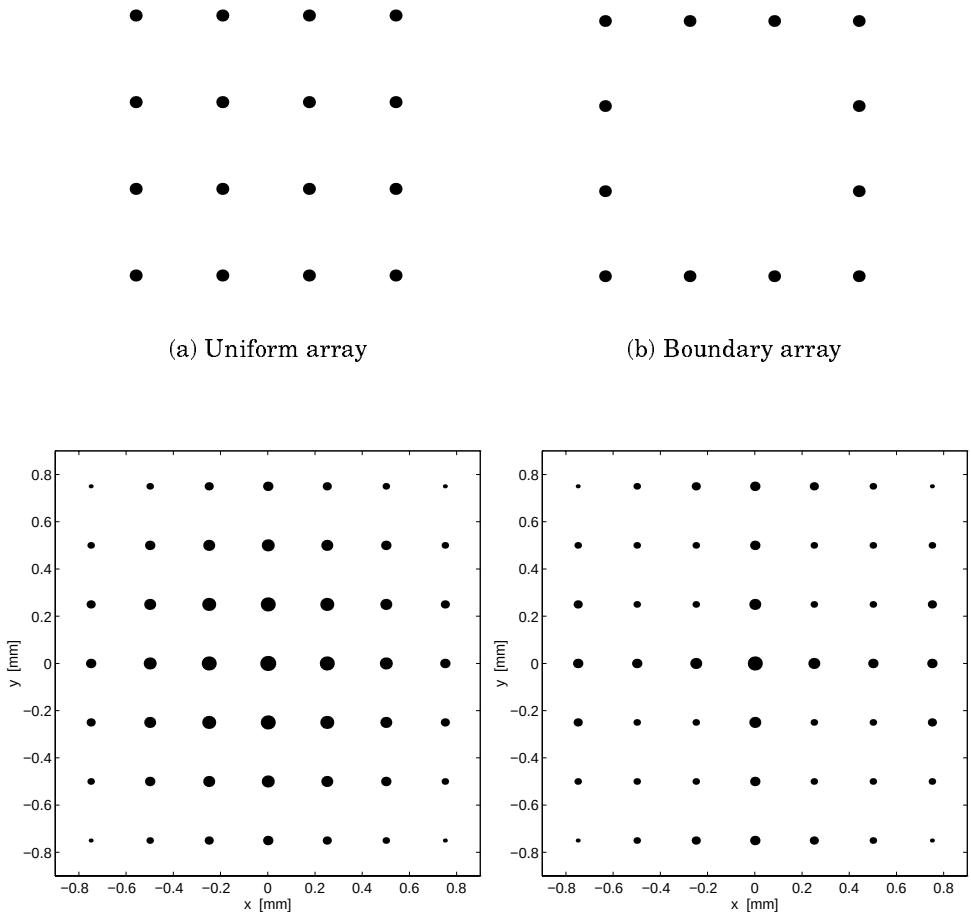


Figure 4.4: Examples of coarray equivalent apertures. A rectangular boundary array is seen to span the same lag space as the uniformly filled rectangular array. The minimum grid spacing is $d = \lambda/2$ in both arrays. The coarrays for the array in (a) is shown in the lower left window and the coarray to the right corresponds to the array in (b).

Other arrays that are coarray equivalent with the two in figure 4.4 exist e.g. a 10 element U-shaped array. Similarly will a thin annular aperture be coarray equivalent to a disc aperture, but not equivalent in coarray values.
A general principle is that *'any two apertures that are coarray equivalent*

can be used to produce images with the same point-spread function' [18]. This means that different apertures can give the same beampattern, if sufficient processing is performed to 'fill in' the contributions from the missing array elements.
Throughout this chapter properties of the difference coarray will be discussed. It will be seen that the coarray displays both the regularities and irregularities of a given array. Regular linear arrays will always give regular coarrays. Irregular linear arrays will always give irregular coarrays. However, with arrays of higher dimensions, irregular coarrays may occur from regular arrays.

4.3 Linear arrays

Linear arrays can be divided into regular and irregular linear arrays. With regular, an equi-spaced separation between the elements is meant. The first has elements along one axis laid out on an underlying regular grid, while the latter has no such underlying grid.
An array need not have every grid position occupied by a sensor or element. Linear arrays with some elements not connected are called *sparse linear arrays*. The elements of an array may be placed in several ways dependent on the particular application it is designed for. In this thesis we distinguish between four types of element configurations.

1. Uniformly filled array
2. Sparse arrays
3. A perturbed array is an array where the elements are displaced from their original equi-spaced grid position. The displacement is often random distributed and is of course within certain limits e.g. [56].
4. A fixed non-equally spaced arrays has elements separated by increasing or decreasing inter-element spacings. The placement of the sensors are according to a formula e.g. a geometric series (or equivalently a logarithmic sequence) or a Gaussian quadrature formula as in [57].

It is known that for a given number of elements in the transducer array one achieves maximum resolution by minimizing the number of redundant spacings in the array [39]. Thus the coarray parameter revealing the resolution of an array is its redundancy factor. A lot of work has been done in order to find minimum redundancy linear arrays. The number of articles on this subject is quite comprehensive, though the search is extensive even for arrays with relatively few elements.

Fortunately the search for these arrays has an equivalent in finding a restrictive difference basis set from a number theory viewpoint [31]. This equivalence seems to have been helpful in the laborious task the search implies. Search algorithms have been presented for equi-spaced sparse arrays in e.g. [5] and [32].
A similar coarray factor showing the degree of sidelobe control has to the authors knowledge not appeared in the literature so far. Generally we know that the sidelobe level is strongly influenced by the number of sensors in the array, but we know little about the relationship with the coarray. In [13] examples are given of very distinct coarrays that almost resembles in magnitude angular response. It is suggested that the differences in coarray values would appear as distinct phase characteristics. This may then indicate that finding at least a single coarray parameter showing the sidelobe level, will not succeed.
However, we anticipate that a further investigation on the relation between the properties in the frequency domain and the morphological properties of the coarray could be a feasible way of determining at least the sampling characteristics of an array. It should also be noted that an improvement in the sidelobe region is observed when non-unity weights are applied to the outer elements of some arrays[4] and that investigations on distinct transmit and receive arrays (as used in [51]) have not been considered.

The discrete arrays of sensors are similar to the FIR (Finite impulse response) filters because of the FIR filters finite temporal extent and the arrays finite spatial extent. Thus the theory and methods of filter design by the window method applies to arrays as well. However, for apertures it represents the relative weighting of the field within the aperture. For arrays, different weights are applied in different sensor locations and is thus a spatial window. There are numerous windows with various properties[16] that have been proposed for filter design.
A feasible way of reducing the sidelobe level is by choosing a window that weights the array. This weighting is often called *apodization* (or *shading* or *tapering*) in case of arrays and antennas and is similar to using a window in spectral estimation. Normally it is done by degrading the excitation amplitude towards the end of the array or by reducing the amplification of the signals received by the sensors towards the ends. The cost of the this reduction in sidelobe level, is as in signal processing, a degraded resolution. In other words there is a tradeoff between the high resolution goal and the will for noise suppression.
It must be mentioned that the maximum achievable spatial resolution in the lateral dimension is fixed once the array's geometry is made. Therefore it is only possible to lower the sidelobes at the cost of broadening the mainlobe of the beam, when weighting the array.

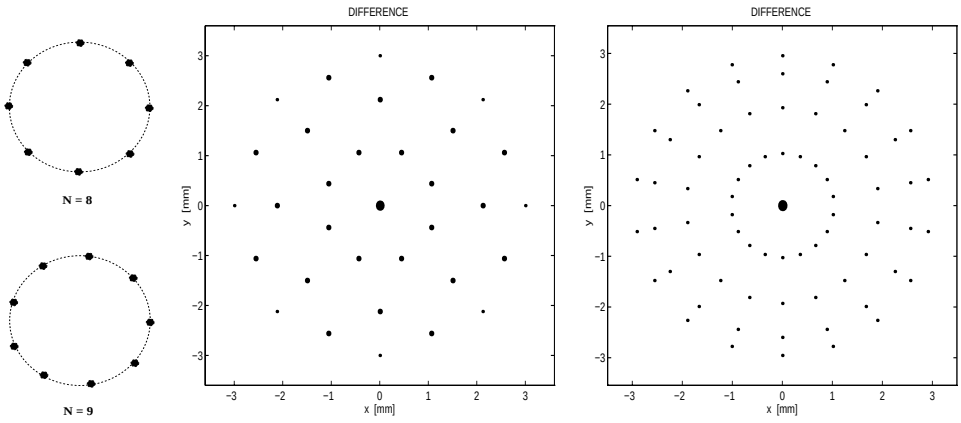
[4] This may indicate that a particular class of arrays can get better performance by a simple weighting. However, simulations with several different arrays reveals that most of them will not display any improvement.

In order to get optimal weights for the array, the Parks-McClellan algorithm [44] can be used. This algorithm is based on the *Remez exchange algorithm* and gives the optimum approximation of filters in the Chebyshev sense when the impulse response is equi-spaced and symmetric. In array signal processing the equivalent demands are that the array must be uniformly filled and have equi-spaced elements symmetrically positioned around the array's origin. The weighted array is then the *minimax approximation* to a desired beampattern, when optimized by this routine. With this algorithm we have superior control over the sidelobes (that will be equi-ripple) as long as the sufficient conditions are met.

4.4 Two-dimensional arrays

The 2D planar arrays have elements placed in both x- and y- direction (or equivalently in both azimuth and elevation direction). The placement can either be on a regular grid or on a plane where no underlying grid can be recognized.
It has already been mentioned that planar rectangular arrays provides focusing and steering in both elevation and azimuth directions. This will reduce the thickness of the scan slices and therefore give better lateral resolution. A serious drawback for two-dimensional arrays is the considerable increase in number of sensors to be handled, when compared to one-dimensional arrays. To have 64-elements in both azimuth- and elevation dimensions demands an array with $64^2 = 4096$ elements. This complexity has been a major incentive for developing sparse array systems.

It can be seen that coarrays of two-dimensional transducer arrays having similar geometries vary widely. Regular two- and three- dimensional arrays can have irregular coarrays [23]! Examples of regular and irregular arrays are the circular 8- and 9-element array in Figure 4.5, respectively. The transducer array with 8 elements is symmetric with respect to both axes and therefore has identical sum- and difference coarrays. It is also a redundant array since lag-values larger than one, appears in the coarray. It should also be noted that this array has an underlying grid and this can be seen as two four element arrays with one rotated by an angle of $\pi/4$. The nine element array has no such underlying grid structure and the coarray displays this irregularity by its complicated structure. Equivalent coarray examples are given in [23, 29].
Several nonredundant two-dimensional (planar) arrays have been proposed. One example is the nine element array in figure 4.5 and another one is the triangular shaped Haubrich-array with 6 elements [17]. The search for arrays with low redundancy is much more laborious with 2D-arrays than with linear arrays. Several configurations for arrays with this property appears in the literature e.g.[6, 7, 15]. However these ar-



(a) Arrays (b) Coarray for N=8 (c) Coarray for N=9

Figure 4.5: An example of circular transducer arrays of similar shape but with distinct difference coarrays.

rays are designed for the case of one propagating signal to be detected. We consider the situation where several signals are present (both signals with information we are interested in and noise signals). So despite these arrays' comparatively high resolution, their sidelobe levels far exceed the desirable for our applications.
As for linear arrays, weighting can be applied to reduce the sidelobe level. In this case the apodization will be both in elevation and azimuth directions. However, attaining control over the skirt of sidelobes for 2D arrays is more difficult than in the 1D case. For example is the mathematics describing two-dimensional systems (functions) less complete than for one dimensional systems. There is no fundamental theorem of algebra for polynomials in two independent variables [26]. Moreover, there are a lot more degrees of freedom in 2D systems.

Nevertheless, methods for expanding some 1D systems to two dimensions have been proposed [21]. A good symmetrical one-dimensional window $w(x)$ may be extended to $w_{2D}(x,y) = w(\sqrt{x^2+y^2})$ which then, becomes a good circular symmetric two-dimensional window[5].
In [24] an approximation for two-dimensional FIR-filters in the Chebyshev sense using the Remez exchange algorithm is presented. One should be aware that approximation functions defined in a two-dimensional domain does not satisfy the *Haar condition* [48]. Therefore the optimum solution is not necessarily unique and we may also have convergence problems. Moreover, finding the parameters is a problem of considerable size since the number of unknowns in the 2D case is N^2, in contrast

[5] The use of circular symmetric filters are well known in the field of image processing [47] and is used for both filtering, compression and image improvement. Image processing can be regarded as a special case of two-dimensional signal processing and thus have some similarities to our field.

to N in the 1D case. Thus finding optimal weights for the planar array case remains a complex problem though some approaches have been proposed in the open literature.
Finally a note on sidelobes in the two-dimensional case, should be given. For linear arrays it is often desirable to minimize the total area under the sidelobes. This measures the energy contribution from signals outside the mainlobe. For two-dimensional arrays the similar will be to minimize the volume under the sidelobes.

4.5 Sensor placement and thinning of arrays

As stated earlier in this chapter the arrays can have various configurations. In this section we will mention three ways of making linear arrays with distinct element locations. The latter method also applies to two dimensional arrays.

4.5.1 Locations based on Gaussian quadrature formulas

In [57, 56] a Gaussian quadrature formula is used to determine the location of the unequally spaced sensors. These formulas are known from the numerical integration theory of mathematics. The famous mathematician Karl F. Gauss discovered that by special placement of the *calculation nodes*, the accuracy of the numerical integration could be greatly enhanced.
A large family of such formulas act in accordance to

$$\int_a^b f(x)dx \approx \sum_{n=1}^{L} A_n f(x_n)$$

Hence only the 'nodes' $\{x_n\}_{n=1}^{L}$ and the 'weights' $\{A_n\}_{n=1}^{L}$ needs to be known [9]. Tables for these values exist for different values of L. The roots of the Legendre polynomial are known to give the nodes for the Gaussian quadrature on the interval $[-1, 1]$ [9].
In [57] a 10 element array is designed from a Gaussian quadrature. No formula is supported in this article, so a brief description will be given here.
A Gauss-Legendre quadrature formula table that includes $L = 10$ i.e. that have 10 nodes, is given in appendix C in [58]. Streit's approach is to use these nodes and scale them with respect to the aperture ends or outer coordinates of an equi-spaced array. For a 10 element linear array the outmost sensor will be located 2.25λ from the array origin. Since the aperture is fixed, this point must be equal to the η times the outmost node (x_1 or x_{10}). Consequently, the scaling factor that must be applied

Nodes	**Weights**
$x_{L-n+1} = -x_n$	$A_{L-n+1} = A_n$
0.9739065285	(-1) 0.6667134430
0.8650633666	0.1494513491
0.6794095682	0.2190863625
0.4333953941	0.2692667193
0.1488743389	0.2955242247

Table 4.1: Nodes and weights from the Gauss-Legendre quadrature formula table in [58] for the value of $L = 10$. Note that only the 10 most significant decimals are included here.

to get a 10 element array with desired aperture is

$$\eta = \frac{2.25\lambda}{0.97390665285} \approx 2.3102833118\lambda$$

The sensor locations and the wavelength normalized location vectors are given in table 4.2 and the values are also recognized from the ξ_ns in [57].

Sensor locations	$\lambda/2$**-location**
$x_n = -x_{-n}$	$\xi_n = -\xi_{-n}$
0.34394190λ	0.68788380
1.00126615λ	2.00253229
1.56962859λ	3.13925717
1.99854146λ	3.99708292
2.25000000λ	4.50000000

Table 4.2: Sensor locations, x_k, and wavelength normalized locations, ξ_n, calculated from the nodes given in the table 4.1. We recognize the values of ξ_n from [57].

Simulations with this array are reported in section 6.2.1.

4.5.2 Locations determined from geometric series

The elements may also be placed in locations determined from geometric series. In that case the spacing is determined from $d_n = a^n$ for $n = 1, 2, \ldots, N$, where the inter-element spacing will decrease if $0 < a < 1$ and increase if $a > 1$.
Thus the nominal sensor locations will be

$$\begin{aligned} \tilde{x}_1 &= 1 \\ \tilde{x}_n &= \tilde{x}_{n-1} + d_{n-1} \quad \text{for } n = 2, 3, \ldots, N \end{aligned} \tag{4.17}$$

When the nominal locations have been determined, they must be scaled according to the desired aperture, A and according to the array origin so that they are symmetric and span the aperture.
First the value of the mid-element $\tilde{x}_{\frac{N+1}{2}}$ is subtracted from the values of $\tilde{x}_n$ for an odd length array. For an even length array, the origin is between the two mid-sensors and this value of their origin is subtracted from the set $\{\tilde{x}_n\}$.
Next, the aperture scaling factor is applied:

$$\{x_n\}_{n=1}^{N} = \tilde{x}_n \cdot \frac{A}{\tilde{x}_N - \tilde{x}_1} \tag{4.18}$$

and we have the N sensor locations spanning the entire aperture, A. The aperture scaling is just the rate between the desired aperture and the nominal aperture.
A routine for calculating the sensor locations from a geometric series, has been programmed and implemented in the UltraSim library. Simulations with two arrays of this kind is reported in section 6.2.2.

4.5.3 Thinning of an array

The performance of the thinned array will be distinct from the unthinned array, with differences depending strongly on the degree of thinning. Obviously, the end elements are *never* thinned or displaced. Removing the end elements will give a narrower aperture and thus a narrower coarray. As a result from this the resolution is reduced.
We consider thinning of the end elements as making a completely new array distinct from the initial one. This is due to the fact that a comparison between these two arrays would be meaningless when both the number of sensors and the apertures are distinct.
One must (or at least one should) distinguish between two types of thinning:

1. Removing elements from a $\lambda/2$-spaced array. The result is an undersampled array where the mean spacing is larger than the half-wavelength and the result is a *sparse array*.

2. Thin elements from an array with initially spacing less than $d = \lambda/2$, until the mean spacing is $\lambda/2$. The result is a *perturbed array*. Normally the perturbation is done by displacing the elements within certain limits (generally half the spacing -$\pm\lambda/4$). When perturbing with a thinning procedure we might get inter-sensor spacings larger than $\frac{3}{4}\lambda$ (i.e the limit for normal perturbation) and this will give less ideal performance of the resulting unweighted array.

The two types of thinning will lead to distinct spatial responses. A perturbed array will demand a different weighting than the non-perturbed,

despite having the same mean spacing, to resemble in response. Undersampling of an array will introduce grating lobes in the visible region.
The removal of elements may be done in various ways. In the preceding sections we have seen that thinning can be done in order to give minimum redundancy and thus maximum resolution. Arrays may as well be designed by picking or placing elements with locations drawn from a random distribution. In this work simulations have been performed mainly with thinning of this kind. The distributions used are *uniform, Gaussian* and *anti-Gaussian distributions*. With the latter distribution the probability of picking an element is larger away from the center of the array.
From these simulations it is observed that an anti-Gaussian thinning is superior to uniform and Gaussian thinning with respect to the sidelobe level. Thus it seems likely that one should keep the central elements active and do thinning on the wings (though not the end elements).

4.6 Calculating the beampattern

Arrays act as spatial filters. They detect signals from certain directions and (try to) suppress all other signals. The shape, size and f-number[6] of an array are some of the factors that affect the shape of the beam.
The beamformers are algorithms in the RF-unit in Fig. 2.1, that are steering and focusing arrays in desired directions and at desired depths. Depending on whether the sources (e.g. a scatterer or reflecting boundary) is located in the near field or far field, they will operate differently. In the far field the waves will be plane waves and we can steer the array towards the radiating (or reflecting) source. However, the ability of extracting range is only possible for sources in the near field where we will have spherical waves.
An old and well known beamforming algorithm is the delay-and-sum beamformer [23]. Its response to a harmonic wave is called the array pattern and determines the array's directivity pattern. The *array pattern* is given by

$$W(\vec{k}) = \sum_{n=0}^{N-1} w_n e^{j\vec{k}\vec{x}_n} \tag{4.19}$$

Here $\vec{k}$ is the wavenumber frequency, $\vec{k} = (k_x, k_y, k_z)$, the element locations are, $\vec{x} = (x, y, z)$ and w_n are the applied weights. Thus it is the Fourier transform of the apodized array. As already stated, in beamforming the array's output for signals from particular directions is important. Therefore one wants to calculate the beampattern

$$W(\vec{k} - \vec{k_0}) = \sum_{n=0}^{N-1} w_n e^{j(\vec{k} - \vec{k_0})\vec{x}_n} \tag{4.20}$$

[6] It is the ratio between the focal length and the diameter of the aperture

Clearly the array pattern can be recognized as the beampattern when the array is 'looking' in the z-direction i.e. $\vec{k_0} = 0$. For a linear array this can be written in terms of the azimuth angle, ϕ, by using the relation $k_x = -k \sin\phi$:

$$W(\sin\phi_0 - \sin\phi) = \sum_{n=0}^{N-1} w_n e^{jk(\sin\phi_0 - \sin\phi)\vec{x}_n} \tag{4.21}$$

The beampattern can be expressed in terms of cosine basis functions when the array displays even symmetry around the origin. For an odd length linear array this can be noticed from any of the beampattern formulas above.
For simplicity the array pattern (Eq. 4.19) is used during the derivation. For an odd length array we define $M = \frac{N+1}{2}$.

$$\begin{aligned} W(\vec{k}) &= \sum_{n=-M}^{M} w_n e^{j\vec{k}\vec{x_n}} \\ &= \sum_{n=-M}^{-1} w_n e^{j\vec{k}\vec{x_n}} + w_0 + \sum_{n=1}^{M} w_n e^{j\vec{k}\vec{x_n}} \\ &\Updownarrow \qquad w_n = w_{-n} \\ W(\vec{k}) &= w_0 + 2\sum_{n=1}^{M} w_n \cos\vec{k}\vec{x_n} \end{aligned} \tag{4.22}$$

One should note the resemblance with symmetric FIR-filters. Performing the same procedure as above, with an even length array will give the same equation except that $w_0 = 0$ and the limit $M = \frac{N}{2}$. A general formulation of this equation for a symmetric linear array, is often written as

$$W(u) = w_0 + 2\sum_{n=1}^{M} w_n \cos(\xi_n u) \tag{4.23}$$

where $\xi_n = \frac{2x_n}{\lambda} \quad n = 1, 2, \ldots, M$
and $u = \pi \sin\phi$
x_n : sensor location in mm from the origin
ξ_n : λ-normalized sensor position
λ : the design wavelength
w_n : the weight factor value at element n

The λ-normalized sensor position is the distance from the sensor to array center, measured in half-wavelengths. It is sufficient to know the positions of the elements and the propagating signal's wavelength to calculate the beampattern. This illustrates the mutual influence both from the signal excitation, the medium and the aperture.
By knowing the transmitted frequency (the central frequency) and the propagation velocity in the actual medium, we can calculate the current

wavelength, λ_0. The beampattern can then be calculated and plotted with u or $\phi = \arcsin(u/\pi)$ as spatial axis. Trade-offs exist in choosing frequency and aperture size, as mentioned in Ch. 2. With steering included we will have the expression

$$W(u - u_0) = w_0 + 2\sum_{n=1}^{M} w_n \cos\Big(\xi_n(u - u_0)\Big) \tag{4.24}$$

where u_0 is the steering angle.

A recent article [35] give an excellent review on beamforming and the inherent trade-offs, both for conventional beams and *limited diffraction beams*. The latter beams are a special class off solutions to the homogeneous scalar wave equation. They are characterized by a large depth of field and good focusing, but suffer from high sidelobes.
Several beamforming algorithm are presented in [41]. With arrays we are sampling a wavefield, hence quantization of the measured values will be needed. In this text these effects have not been discussed. An investigation on the effect phase quantization errors have on the sidelobe level in a steered and focused beamformer is given in [20].

4.7 Beampatterns from the coarray

Arrays provide sampling of wavefields. We can assume the acoustic field containing both the array and the propagating waves to be a wide sense stationary (WSS) random field. Moreover, one can assume the propagating signals to be plane waves i.e. we are in the far-field region. The WSS field criterion requires that the mean function does not depend on the location of a point and that the autocorrelation function depends only on the distance and orientation between two points in space. In this context the manner by which the array samples the spatial correlation function of the WSS field is focused.
We observe the wavefield through a spatially limited continuous aperture, with aperture function $w(\vec{x})$. The signal from the aperture will then be $z(\vec{x}, t) = w(\vec{x})f(\vec{x}, t)$. Assuming the field to be stationary in both domains, the correlation function of the aperture output can be averaged over all baselines to give an estimate of the wavefield's correlation function [23]

$$\begin{aligned}\int_A R_z(\vec{x}_1, \vec{x}_1 + \vec{\chi}; \tau) &= R_f(\vec{\chi}, \tau)\int_A w(\vec{x}_1)w^*(\vec{x}_1 + \vec{\chi})d\vec{x}_1 \\ &= c(\vec{\chi})R_f(\vec{\chi}, \tau)\end{aligned} \tag{4.25}$$

Here R_z and R_f are the aperture's estimated and the wavefield's correlation functions, respectively.

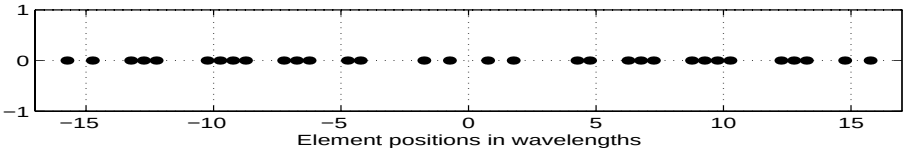
For an array, the expected value of the spatiotemporal correlation function can be expressed as [23]

$$\mathcal{E}\left[\sum_{(a,b)\in\vartheta(\vec{\chi})} y_a(t)y_b^*(t+\tau)\right] = c(\vec{\chi})R_f(\vec{\chi},\tau) \tag{4.26}$$

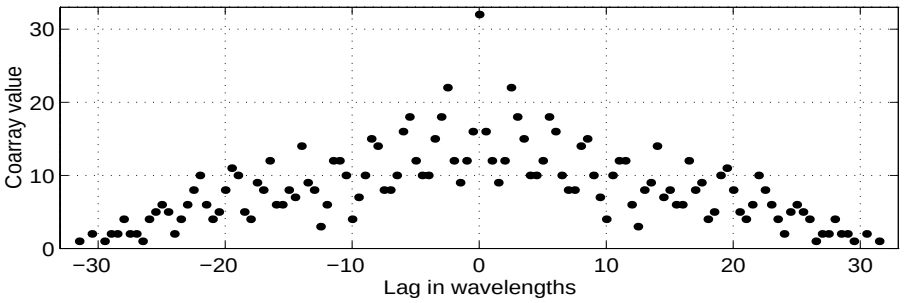
Here we consider the output from sensor $m = a$ as $y_a(t) = w_a f(\vec{x}_a, t)$, where w is the weight and f is the signal. The coarray of an array is clearly the set of points where the array spatially samples the wavefield's correlation function.
In the frequency domain the estimated power spectrum of the wavefield, $S_f(\vec{k},\omega)$, will be a smoothed version of the true spectrum. The Fourier transform of the coarray is acting as a smoothing filter in this domain. This is similar to the smoothing properties of windows in spectral estimation algorithms and to the beampattern algorithms of array processing.
This suggests that the coarray is the inverse Fourier transform of the far-field beampattern, and thus knowing the coarray values for a given array geometry offers an alternative way of calculating the beampattern under the assumptions made above. It simplifies to a Fourier transformation of the coarray lag values.
In principle it is, from the discussion above, possible to decide how well the array samples the field by viewing in what manner the array samples the spatial correlation function. As the coarray determines the points where the correlation function is sampled, it is desirable to have a geometry that produces as many samples of the correlation function as possible from an array with a given number of elements. The minimum spacing between neighbouring coarray elements must be small enough to avoid aliasing in the power spectrum function. This follows from the Fourier relation between the autocorrelation function and the power spectrum [26].
The beampattern or squared aperture function is equal to a Fourier transformation over the coarray lag space

$$B(\vec{k}) \quad = \quad \mathcal{F}\{c(\vec{\chi})\} = \sum_{\vec{\chi}} c(\vec{\chi})e^{j\vec{k}\vec{\chi}} \tag{4.27}$$

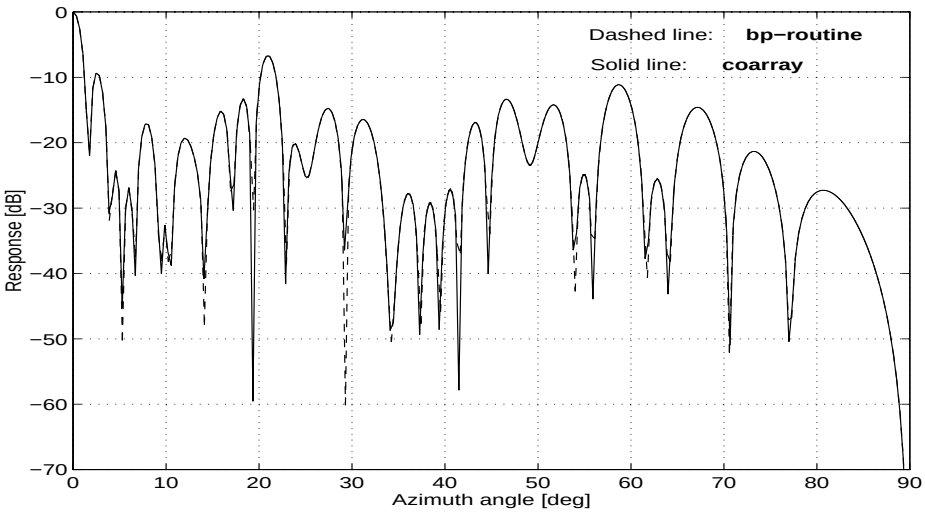
Results from both UltraSim's beampattern routine and from the coarray tools of this work are given in Figs. 4.6 and 4.7.



(a) Transducer array



(b) Unweighted coarray



(c) Beampattern

Figure 4.6: Example of beampattern produced by the coarray approach and by the UltraSim beampattern routine. The upper panel displays the array that has an underlying grid of $\lambda/2$. Originally, the array has 64 elements and aperture 32λ, but has been thinned 50% by removing elements randomly (according to a uniform distribution). The middle panel shows the non-weighted coarray of the thinned array and the lower panel contains the produced beampatterns from either methods.

The beampatterns from the two methods are equivalent when sufficient conditions are met. A minor discrepancy at the angles where the response approaches zero, is observed in the beampattern for the arrays in fig. 4.6. Everywhere else the calculated responses coincide.
In Fig. 4.7 we have calculated the angular response from the same array as above, but now we observe the field at a closer range (i.e. 60 mm). A stronger deviation is now observed increasing with larger angles and for this reason it can be stated that observating the field at this distance from the transducer not validates the planar wave assumption.

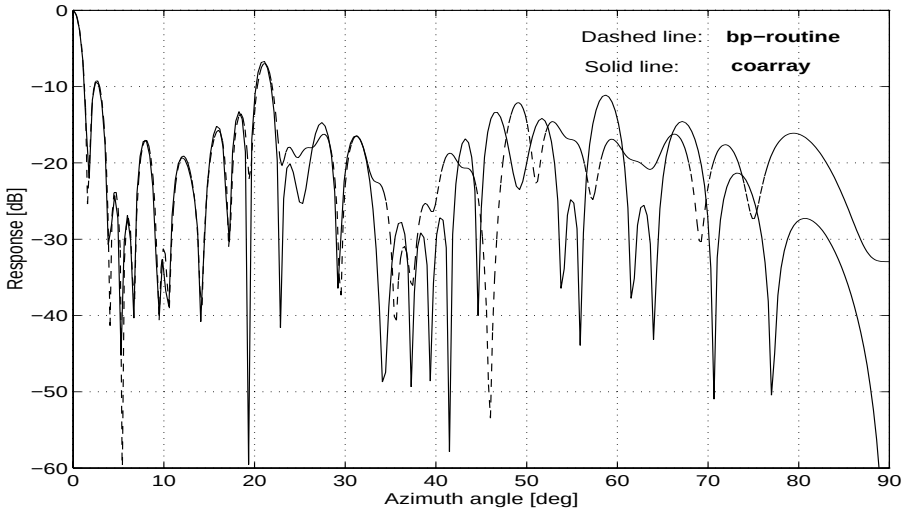


Figure 4.7: Example of beampatterns produced by the coarray approach and by UltraSim's **bp**-routine when the far-field condition is not met. It is observed that the discrepancy increases with azimuth angles.

4.8 Coarray calculation and visualization tools

Two of the routines developed in this thesis, calculates the coarrays and visualize them in a plotting window. In this section we will present these routines. Since both uniform, sparse and nonequally spaced arrays will be considered, the algorithms must have a general formulation. The coarray calculation algorithm developed here will definitely be slower than the fast methods appearing in the literature. However, our algorithm give correct results for all the above mentioned element configurations both for one- and two-dimensional arrays. The method in e.g. [32] demands a uniformly filled linear array with sensors separated by an integer multiple of a unit spacing d.
The visualization routine written, visualizes the coarray both in the correlation domain and in the spatial frequency domain. The preceding figures with arrays, coarrays and beampatterns are all produced by these two routines.

4.8.1 Calculating the coarray values

The routine that calculates the coarray values *cocomp* will now be described. Input to this routine is the array's x- and y-coordinates $\{\vec{x}_k\}_{k=1}^{N}$. Output is the coarray values and their x- and y- lag coordinates as the set $\{c(\vec{\chi_m})\}_{m=1}^{M_d}$.
Only planar arrays are considered, so calculations on the $z-$ coordinates have been omitted here. Consequently, the effects of curving the trans-

ducer surface cannot be investigated in this program. However, the calculation routine can easily be extended to 3-D arrays as well.
The main principles of the ***cocomp*** algorithm for the difference coarray case, can be formulated as

for k = 1 : N $\quad \%\forall\{x_k\}_{k=1}^{N}$
  $\Psi_k = \vec{x}_k - X$ *%in case of a difference coarray*
  if $\Psi_k^i \in \chi_{k-1}$
    % Update lag values of this lag space member
    $\chi_k = \chi_{k-1} + w_k * w_i$
  else $\% \forall\ \Psi_k^i \notin \chi_{k-1}$
    % Update current lag space by including new member
    $\chi_k^{1:2} = \left[\chi_{k-1}\ \Psi_k^i\right]$
    % Calculate and include lag value of new member
    $\chi_k^3 = \chi_{k-1} + w_k * w_i$
  end *% if*
end *% for loop*

The terms used in this algorithm are

$$
\begin{aligned}
X &= \{\vec{x_i}\}\ , i = 1, 2, \ldots, N \\
\Psi_k &= \left[\Psi_k^1 \Psi_k^2 \cdots \Psi_k^N\right] \quad \text{where } \Psi_k^i = \vec{x}_k - \vec{x}_i \\
\chi_k &: \textit{current lag space} \\
\chi_{k-1} &: \textit{lag space of previous iteration}
\end{aligned}
$$

In the calculation, the element locations are considered to be vectors from the array center, $(0, 0)$, to their coordinates, (x_i, y_i). Then vector-subtraction or addition is performed sequentially over the aperture as shown by the loop in the algorithm.
Equal lags are summed according to Eqs. 4.10 and 4.16, and distinct lags are included in the updated lag space. It is this search that is the most time consuming part of the algorithm and that presumably would gain most to make more efficient.

4.8.2 Visualizing the coarray

A routine for visualization of the coarray values and some options (like plotting their frequency responses) have also been developed during this work. The visualization tools includes, among others, these options

- *Plotting of a linear coarray along with its error coarray.*
  The error coarray will maybe be misguiding (or at least difficult to interpret) when optimal weighting is applied. This is due to the fact that the error coarray algorithm is written to calculate the deviation array (i.e. the error between the filled and the sparse array) in the non-apodized case. When calculating optimal weights, the weight for each element in the sparse case, will be distinct from what they would be in the uniform case.
  In case of weighting with conventional (e.g. Hamming) apodization the individual weights are calculated independently of the sparsity structure of the arrays. The missing element's weights are in that case just set to zero and the remaining elements have weight values that are not influenced by the missing elements.
  As a result, with optimal weighting, adding the coarray and the error coarray will not give the optimal uniform coarray as a result (as would be the case for uniform weighted arrays and conventional apodization).

- *Two-dimensional coarrays* are plotted in their x- and y- lag space with coarray values corresponding to the size (or area) of the plot objects. An optional plot shows the coarray as points in a 3D space.

- For linear array the possibility of plotting its *spatial frequency response* is available. This has been shown to be equal to the array's beampattern.
  This transformation was first based on the FFT. It turned out to be slower than using the array pattern formula directly. The reason for this was that the FFT had to be supported with time consuming tests, in order to give reliable results when the array was thinned or unequally spaced.
  For a sparse or non-equally spaced array a regular underlying grid had to be recognized. It had to be a dense grid in order to minimize the discrepancy between the actual lag-coordinate and the grid coordinate. Grid locations without any sensor was then set to zero, before a FFT calculation was performed. Because of the great number of operations necessary to make a grid and check it, the array pattern of Eq. 4.27 was chosen for our calculations.

4.9 Summary

The fundamentals of coarrays with some of their properties regarding linear and planar arrays, have now been presented. Different element configurations for linear and two-dimensional arrays have also been presented. Some key words have been explained. Various thinning methods are reviewed and some necessary restrictions on thinning are mentioned. We have also seen that the coarray offers an alternative ap-

proach to the calculation of the far-field beampattern of an array.
Two program routines are written to take care of the calculation and visualization of the coarrays. The prior routine is algorithmically described, while for the latter the most important of the available functions have been summarized. A few examples displaying resemblance with known examples in [23] and with the already existing beampattern routine in UltraSim have been included as well.

In the following chapter we will take a closer look at the problem of designing arrays both in case of uniform-, sparse-, perturbed- and non-equally spaced arrays. We will there consider the possible optimization of weights for improving the sidelobe level.
A general Remez algorithm has been made to allow opimization of weights for irregular spaced and sparse arrays. This routine will be a central point when the achieved results of this thesis are reported.

Chapter 5

Methods for array synthesis

Designing an array system will imply several trade-offs. One of them is between giving high resolution and at the same time subdue noise generated by the sidelobes. Doing this with respect to a given criterion in an optimal way, is desired. This has been an important research activity for many years and numerous articles on this problem have been published. Thus several different approaches have been proposed for optimization of the array performance with respect to both resolution and sidelobe control. They can be divided into four main groups:

1. Fixed uniform sensor locations and optimization of weights.
2. Fixed non-uniform sensor locations and optimization of weights.
3. Uniform weighting and optimization of the sensor locations.
4. Simultaneous optimization of both the sensor positions and the weights.

Our work is mainly restricted to the first group and the main objective in this chapter is to formulate and implement an algorithm for optimization of equi-spaced and non-equally spaced arrays as well as sparse and perturbed arrays.

5.1 Formulating the optimization problem

The objective is to minimize the maximum weighted approximation error between a desired response, D, and the actual angular response, P.

This is the *minimax* or *Chebyshev* criterion given by

$$min\Big\{\|E\|\Big\} \quad = \quad min\Big\{max_{u\in F}\Big|W(u)\Big(D(u)-P(u)\Big)\Big|\Big\} \tag{5.1}$$

where W is the error weighting function that allows different weights to be applied in different regions of F, the set of frequencies upon where the optimization is performed. Equivalently $\|E\|$ is the l_∞-norm of $E(u)$. The actual response is the array pattern in Eq. 4.23, written in a polynomial form

$$P(u) \quad = \quad \sum_{k=0}^{r}\alpha_k\cos(\xi_k u) = b_0 + 2\sum_{n=0}^{M} b_n\cos\xi_n u \tag{5.2}$$

$$\text{with} \quad \xi_k \quad = \quad \frac{2x_k}{\lambda}, \quad u=\pi\sin\phi \tag{5.3}$$

$$\text{and} \quad b_0 \quad = \quad \alpha_0, \quad b_n = b_{-n} = \frac{\alpha_n}{2} \quad \text{when odd} \tag{5.4}$$

$$b_n \quad = \quad b_{-n} = \frac{\alpha_n}{2} \quad \text{when even}$$

The optimization problem can be stated as:
Given an array of elements $\Big\{x_k\Big\}_{k=1}^{N}$*; find the set of optimum weights* $\Big\{b_k\Big\}_{k=1}^{N}$ *that makes the maximum error,* E*, a minimum, for a given input desired response,* D*.*

The cosines in Eq. 5.2 form the set of $r+1$ basis functions

$$\Big\{\cos(\xi_0 u),\cos(\xi_1 u),\ldots,\cos(\xi_r u)\Big\}$$

used in the optimization. In order for $P(u)$ to be a unique and optimal solution of the problem, the set of basis functions must obey the Haar condition[8]. In other words each $\cos(\xi_i u)$ is continuous and every set of $r+1$ vectors of the form $\Big\{\cos(\xi_0 u),\cos(\xi_1 u),\ldots,\cos(\xi_r u)\Big\}$ is independent. Because our method implies solution of equations and matrix inversion, the set of basis functions should also be well conditioned for numerical work. When strong spatial oversampling occurs, this becomes a problem in our algorithm and causes divergence of the operation. However, this is considered a special case of operation.
The Remez exchange algorithm is based on the alternation theorem:

Theorem 1 (Alternation theorem) *If* $P(u)$ *is a linear combination of* $r+1$ *cosine functions of the form*

$$P(u) = \sum_{k=0}^{r}\alpha_k\cos(\xi_k u)$$

then for $P(u)$ *to be the unique, best weighted Chebyshev approximation to a continuous function* $D(u)$ *on* F *i.e. a compact subset of* $[0,\pi]$*, the error*

function $E(u)$ *must exhibit at least* $r+2$ *extremal frequencies in* F. *In other words there exist* $r+2$ *points* u_i *in* F *such that* $u_0 < u_1 < \cdots < u_{r+1}$, $E(u_i) = -E(u_{i+1})$ *and* $\|E(u_i)\| = max_{u \in F}|E(u_i)|_{i=0}^{r+1}$.

In our case the desired response is given by

$$D(u) = \begin{cases} 1 & : \quad 0 \leq u \leq u_p \\ 0 & : \quad u_s \leq u \leq u_O \end{cases} \tag{5.5}$$

Here u is an angular spatial frequency axis, where u_p is the desired cut-frequency of the main beam, u_s is the initially chosen angle for where the maximum sidelobe level should be reached and u_O is the upper limit of the frequency region where the optimization takes part. It should be noted that in this formulation the interval of optimization can be limited to $u \in [0, u_O]$ (with $u_O < \pi$), when desired.
The reason is that with this method we only have a limited number of parameters to weight down peaks in the sidelobe region. When an array is thinned it is observed that the number of peaks in this region is larger than the number of sensors and hence larger than the number of weight coefficients. We presume that this is because the number of peaks is determined by the underlying regular grid and not by the number of sensors.
We have observed that in case of sparse arrays, the optimization region is limited to frequencies from $0°$ to approximately

$$u_O = arcsin\Big(\frac{N_{sparse}}{N_{grid}}\Big) \tag{5.6}$$

i.e. the rate between the number of sensors in the sparse array and the uniformly filled array.
Whenever an odd length uniform array is optimized $u_O = \pi$. Even length arrays tend to develop an extra sidelobe at $u = \pi$ [56] for some special choices of inter-element spacing and cut-angles. Extreme care must therefore be taken when the array deviates from the $\lambda/2$-case, and u_O may have to be reduced from the value, $u_O = \pi$, in order to avoid the development of badly suppressed sidelobes or grating-lobe like responses.
The error function is written

$$E(u) = W(u)\Big(D(u) - P(u)\Big) \tag{5.7}$$

where the weight function is

$$W(u) = \begin{cases} 1 & : \quad 0 \leq u \leq u_p \\ K = \dfrac{\delta_p}{\delta_s} & : \quad u_s \leq u \leq u_O \end{cases} \tag{5.8}$$

Here δ_p is the amplitude of the ripple in the mainlobe region and δ_s is the amplitude of the ripple in the sidelobe region.
From the alternation theorem (theorem 1) a set of extremal frequencies $\{u_i = \pi \sin\phi_i\}$ and a cosine polynomial, $P(u)$ exists, such that

$$E(u_i) \quad = \quad W(u_i)\Big(D(u_i) - P(u_i)\Big) = (-1)^i\delta\,, \quad i = 0, 1, \ldots, r+1 \tag{5.9}$$

Here δ is an unknown constant. We rewrite this equation to

$$P(u_i) + \frac{(-1)^i \delta}{W(u_i)} = D(u_i) \quad \text{for} \quad i = 0, 1, \ldots, r+1 \tag{5.10}$$

and recognize it as a matrix equation system with $r+2$ unknowns

$$\boldsymbol{A}\vec{\boldsymbol{\alpha}} = \boldsymbol{D} \tag{5.11}$$

with the matrices defined as follows

$$\boldsymbol{A} = \begin{bmatrix} \cos\xi_0 u_0 & \cdots & \cos\xi_r u_0 & \frac{\delta}{W(u_0)} \\ \cos\xi_0 u_1 & \cdots & \cos\xi_r u_1 & \frac{-\delta}{W(u_1)} \\ \vdots & \vdots & \vdots & \vdots \\ \cos\xi_0 u_{r+1} & \cdots & \cos\xi_r u_{r+1} & \frac{(-1)^{r+1}\delta}{W(u_{r+1})} \end{bmatrix} \quad \vec{\boldsymbol{\alpha}} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_r \\ \delta \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} D(u_0) & D(u_1) & \cdots & D(u_r) & D(u_{r+1}) \end{bmatrix}^T$$

Consequently, if the extremal frequencies are known, the coefficients α_i can be determined by matrix inversion of the system above. Moreover, invertibility of the matrix $\boldsymbol{A}$ is guaranteed by the Haar condition on the basis functions [8]. Then as the coefficients are found the angular response of the array can be calculated.

5.2 Optimization of sensor weights when sensor locations are fixed

This optimization problem has, in the linear array case, two approaches.

1. In case of a uniformly filled array with sensors separated by equal spacings, the widely known and used Dolph-Chebyshev shadings applies [11]. This is an equivalent problem to conventional FIR filter design by using the Remez algorithm of [44].
2. In this case the sensors are not separated by equal spacings. This demands a more general formulation than in the previous case. Optimization for this problem is reported in [57, 56]. We will use the formulation above to optimize arrays of this kind.

5.2.1 The equi-spaced array case

A special case of the problem formulated above is when the arrays are uniformly filled and are equally spaced with sensors located at an integer times the unit spacing, d. An equi-spaced array will have sensors

at $x_k = kd = k\gamma\lambda$ for $k = -M, \ldots, 0, \ldots, M$ and hence $\xi_k = 2k\gamma$. The well-known Dolph-Chebyshev shadedarrays are in this family and have a unit-spacing of one-half the wavelength i.e. $\gamma = 0.5$, $d = \lambda/2$ and the cosine argument of Eq. 5.2 simplifies to

$$d = \frac{\lambda}{2} \quad \Rightarrow \quad \xi_n u = \begin{cases} nu & \text{, odd length array} \\ \frac{2n-1}{2}u & \text{, even length array} \end{cases}$$

In the case of an equi-spaced array the powerful algorithm of Parks-McClellan can be used [37, 44]. Their notation differs from the one given in the preceding section.
First it is noted that the array pattern in Eq. 5.2 can be written on the form

$$P(u) = Q(u)R(u) = Q(u)\sum_{k=0}^{r} \alpha_k \cos(\xi_k u) \tag{5.12}$$

For an odd length array $Q(u) = 1$ and for an even length array $Q(u) = cos(u/2)$. In [37] the optimization is performed over a different frequency grid (from 0 to 0.5) than in the previous section (where we were optimizing over $[0, \pi]$).
A new approximation problem is next formulated as

$$\begin{aligned} E(u) &= W(u)\left[D(u) - Q(u)P(u)\right] \\ &= W(u)Q(u)\left[\frac{D(u)}{Q(u)} - P(u)\right] \\ &= \hat{W}(u)\left[\hat{D}(u) - P(u)\right] \end{aligned} \tag{5.13}$$

From the alternation theorem the system is written :

$$\begin{bmatrix} 1 & \cos 2\pi u_0 & \cdots & \cos \pi n u_0 & \frac{\delta}{\hat{W}(u_0)} \\ 1 & \cos 2\pi u_1 & \cdots & \cos 2\pi n u_1 & \frac{-\delta}{\hat{W}(u_1)} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & \cos 2\pi u_{r+1} & \cdots & \cos 2\pi n u_{r+1} & \frac{(-1)^{r+1}\delta}{\hat{W}(u_{r+1})} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_r \\ \delta \end{bmatrix} = \begin{bmatrix} \hat{D}(u_0) \\ \hat{D}(u_1) \\ \vdots \\ \hat{D}(u_r) \\ \hat{D}(u_{r+1}) \end{bmatrix} \tag{5.14}$$

The system above is always nonsingular [8]. However, the possibility of numerical ill-conditioning remains and solving the system becomes time consuming with large system. Therefore a faster and more stable approach has been taken in this algorithm.
First δ is analytically deduced by solving with *Cramers rule* applying calculation of the *Van der Monde determinant*[36]. Thus the parameter δ is deduced from

$$\delta = \frac{\sum_{k=0}^{r+1} a_k \hat{D}(u_k)}{\sum_{k=0}^{r+1} \frac{(-1)^k a_k}{\hat{W}(u_k)}} \tag{5.15}$$

Next $P(u)$ is interpolated on the $r+2$ frequency points by means of the barycentric form of the Lagrange interpolation formula [42]. The interpolated response $P(u)$ is then given by [44]

$$P(u) = \begin{cases} C_k & \text{for extremal frequencies} \\ \dfrac{\displaystyle\sum_{k=0}^{r} \frac{\beta_k}{\cos u - \cos u_k} C_k}{\displaystyle\sum_{k=0}^{r} \frac{\beta_k}{\cos u - \cos u_k}} & \text{otherwise} \end{cases} \tag{5.16}$$

$$\text{where} \quad a_k = \prod_{\substack{i=0 \\ i \neq k}}^{r+1} \frac{1}{\cos u_k - \cos u_i}$$

$$C_k = \hat{D}(u_k) - \frac{(-1)^k \delta}{\hat{W}(u_k)}$$

$$\beta_k = \prod_{\substack{i=0 \\ i \neq k}}^{r} \frac{1}{\cos u_k - \cos u_i} = a_k(\cos u_k - \cos u_{r+1})$$

$$k = 0, 1, \ldots, r$$

The algorithm starts with an initial guess on $r+2$ extremal frequencies. Next, the error, $E(u)$, is calculated over a dense grid, u, and the extrema of $E(u)$ are considered candidates for the new set of extrema.
The r extrema with the largest error and the initial extrema at the beginning and end of the transition region, i.e. the initial cut-frequencies u_p and u_s, are used to update the set extrema for the next iteration. This is the most laborious part of this algorithm and is the part where most improvements of this routine have been proposed e.g. in [2]. The algorithm terminates when no change has occurred since the previous iteration.
Finally, the sensor weights are calculated by an inverse discrete Fourier transform of the response. This algorithm provides a fast and reliable way of calculating the optimal sensor weights in case of a uniformly filled linear array.

5.2.2 A general formulated algorithm

Since our objective is to optimize sparse and non-uniformly filled arrays, the algorithm will take a slightly different form than in the preceding section. In this case there will generally be no simplification of the cosine argument, $\xi_k u$. From the alternation theorem we can formulate equation 5.9 and form the system of Eq. 5.11

$$\boldsymbol{A}\vec{\boldsymbol{\alpha}} = \boldsymbol{D}$$

We are now using $\{u_k = \pi \sin\phi_k\}$ as the set of extremal frequencies for $k = 0, 2, \ldots, r+1$ and the matrices as defined on page 64.
This formulation is valid both for odd and even length arrays and no new approximation formulation is needed to solve for even length arrays in contrast to the Parks-McClellan formulation. Another difference with the earlier described algorithm is that 1 (one) will be a basis only with odd length arrays where the center element is active i.e.$x_0 = \emptyset$. We should also note that in the general case optimize in the $u \in [0, \pi]$ region and that the system matrix, $\boldsymbol{A}$, therefore not resembles the similar matrix in the equi-spaced case.
In our solution to the problem we employ matrix inversion. As mentioned in the problem formulation in Sec. 5.1 this may give an ill- conditioned matrix system when large and oversampled arrays are used. However, since our main concern is undersampled (i.e. sparse) arrays and non-equally spaced array, this will not be any problem in this context.
A sparse array will typically have a mean spacing, $\tilde{d}$, larger than one-half wavelength, often from $\tilde{d} = \lambda$ to more than 2λ. A non-equally spaced array generally has a mean spacing around the Nyquist rate i.e. $\tilde{d} = \lambda/2$. Consequently, only for very special arrays will bad conditioning cause problems and that will not occur for the arrays that we consider.

5.2.3 Inputs to the algorithm

There are several input parameters to this algorithm. Firstly, we need to know the sensor locations $\{x_k\}_{k=0}^{N}$. Thus an input vector contains these coordinates.
We also need to know the propagation speed of the media, c , and the design frequency, f, in order to calculate the wavelength, λ. This is supplied by the media- and excitation vectors.
In our routine it is also possible for the user to alter the number of frequency points per ripple in the calculations i.e. the *oversampling rate*. This is useful for special geometries, where a more dense grid sometimes is necessary to get convergence. Our experience is that this occurs only for some very special perturbed arrays.
The parameters that are given as direct inputs by the user are:

1. The initial cut-angle of the beam or mainlobe, ϕ_{i_p}, are given in degrees ($^\circ$).

2. The angle where it is desired that the maximum minimum sidelobe level is reached, ϕ_{i_s} in degrees, i.e. where the equi-ripple sidelobe level starts. This is the first angle where the error or response is δ.

3. The stop angle of the optimization region, ϕ_O in degrees. For uniformly filled arrays and non-equally spaced arrays with mean spac-

ing $\tilde{d} = \lambda/2$ this angle should be chosen to $90°$, but for sparse arrays it has to be reduced. As a rule of thumb this angle should be chosen to

$$\phi_O = \arcsin\left(\frac{N_{sparse}}{N_{filled}}\right)$$

where N_{filled} is the number of elements in a filled equi-spaced array of equivalent aperture.

4. The approximation error weight factor, K. This ratio specifies the difference in error between the two optimization intervals. In our application it is always larger than or equal one, but in most cases it will not have any effect in our applications and the error in the sidelobe region will be the smallest possible minimum regardless of the chosen K. The reason is that we look at beamforms that has a narrow mainlobe with no extremal frequencies except from ϕ_{i_p}.

5.2.4 Brief description of the algorithm

The first part of the algorithm is checking whether the input array meets the conditions implicit on the approximation criterion with respect to symmetry around the array origin.
Next a dense grid is set up according to the input angles and the oversampling rate. An initial set of extremal frequencies are chosen and the matrix system is solved

$$\vec{\alpha} = \boldsymbol{A}^{-1}\boldsymbol{D}$$

Now both the polynomial, $P(u)$ and the error function, $E(u)$ is available.

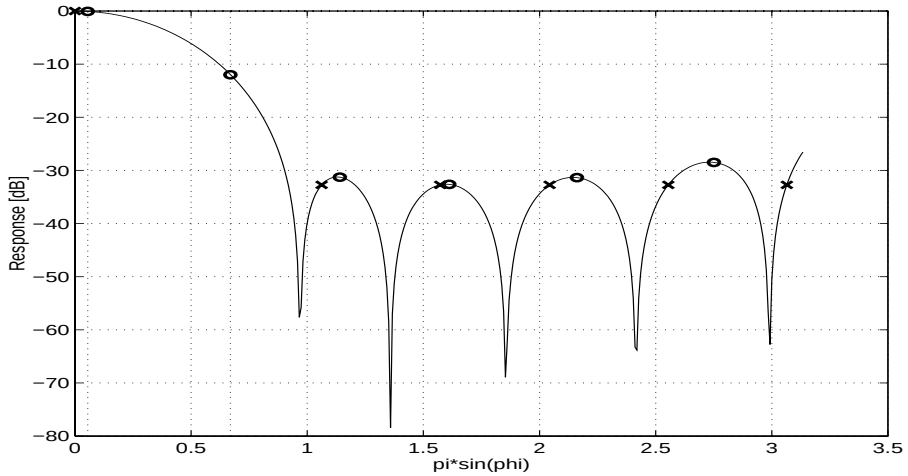


Figure 5.1: This is an example of the first iteration of the algorithm. The crosses indicates response values for the initial chosen extremal frequencies after the matrix has been solved. The level of the crosses are $20\log_{10}(\delta/K) \approx -33$ dB. The circles indicates the extrema determined from the sign changes of the differentiated response that will be the updated extremal frequencies to be used in the next iteration.

Then we will find the extremal frequencies and determine whether they are equivalent to the ones in the previous iteration or not.
To find the maximums of the error (and equivalently, the response since $D \equiv 0$ in this interval), we have used a numerical derivative estimation. Because of the similarity between P and E the extremal frequencies are found for exactly the same values. Therefore we have differentiated the response,P. Moreover, since we are considering a situation with a narrow mainlobe, the extremal frequencies with maximum error will be expected to be in the sidelobe interval of the optimization region.
The estimations of the derivatives are done by

$$\mathcal{D}\{P(u)\} = \frac{P_{k+1} - P_k}{u_{k+1} - u_k} \tag{5.17}$$

Now the frequency values where $\mathcal{D}\{P(u)\}$ changes sign are considered candidates for the new extrema. The r candidates alternating in sign and with largest error, are chosen to be the extrema of the next iteration. Additionally, u_p and u_s are included in this set. Then a new iteration is performed.The algorithm terminates if no change has occurred in the last iteration or if forced to by the user.
Finally, the weight function and the response are plotted. The angle value of the first zero-crossing, the maximum sidelobe level and the *dynamic range of the weight coefficients*, Δ_D, are calculated and written to the user. The Δ_D factor is the ratio between the largest and smallest weight value.
In each iteration there has to be exactly $r+2$ extremal frequencies. Convergence is normally reached after only a few iterations $(3-7)$.

The differences between this algorithm and the one given in [37] can be summerized as:

1. The cosine arguments are chosen to resemble with Dolph's in [11]. Therefore an optimization region spanning from 0 to π is used rather than from 0 to 0.5. Introducing these parameters also makes it possible to calculate both even and odd length array from the same matrix formulation by applying the alternation theorem. Moreover no 'adjustment factor' $Q(u)$ is needed.

2. Instead of applying Lagrange interpolation we solve the matrix system by matrix inversion. A motivation for this choice was given in section 5.2.2.

3. Because we are interested in beamforms resembling a narrow lowpass filter, the search can be simplified to search for extrema in the sidelobe interval only. This involves a saving in the number of operations.

4. Because we give the user the possibility of changing the region where the optimization is performed over and, at the same time,

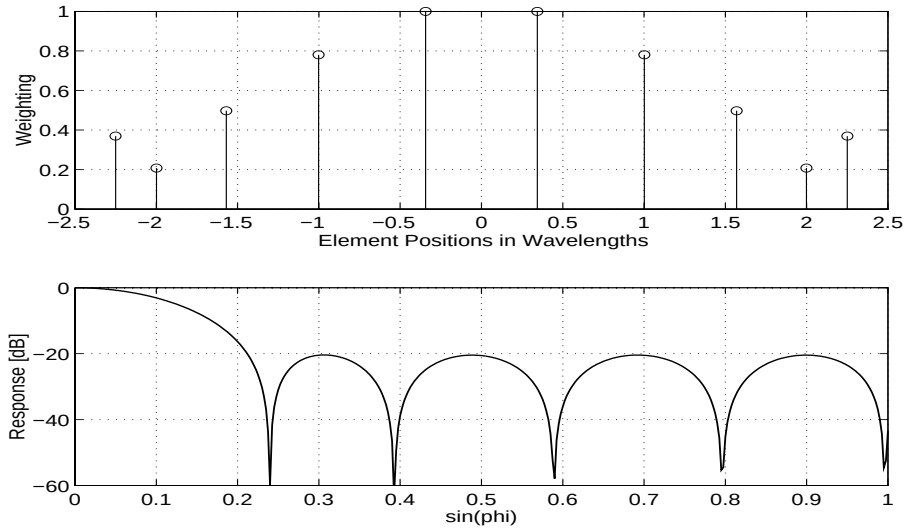


Figure 5.2: This is an example of an end result from the optimization routine. The upper plot is the weights that will be applied to the corresponding element, normalized with respect to the greatest weight value. In the lower panel the weighted response is showed. The same array as in Fig. 5.1 is used and it is an unequally spaced array with 10 elements (see fig. 6.4b). The first zero appears at $4.22°$, the maximum sidelobe level is -23.56 dB and the weight dynamic factor is 2.76. Convergence was reached after 4 iterations.

the possibility of changing the oversampling factor, a wide class of arrays can be optimized within regions of the angular domain. This includes sparse arrays, perturbed arrays and non-equally spaced arrays in addition to equi-spaced arrays.

5. Several key numbers are also supplied to the user in order to make it easier to compare different arrays and different element configurations.

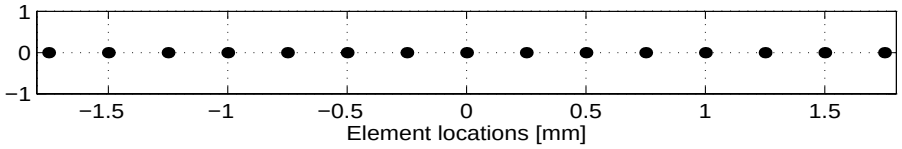
5.3 Other optimization approaches

In the scientific literature other optimal and near-optimal approaches have been proposed that either optimize by judicious placement of the sensors or by optimizing both the weighting and sensor locations. Typically these arrays will have sensors located at $k\lambda$, when k is a real non-integer. The author doubts the usefulness of such arrays for ultrasonic use. Making these non-uniform spaced arrays for ultrasonic applications, will meet severe difficult because of the short wavelengths in ultrasound, and thereby also the limited size of ultrasound arrays.
Realization of such arrays will mainly find application in fields where very large arrays are applied i.e. where the non-uniform (decimal fractioned wavelength) placement is realizable. However, these arrays offer interesting points of view, that in some special cases may shed light unto

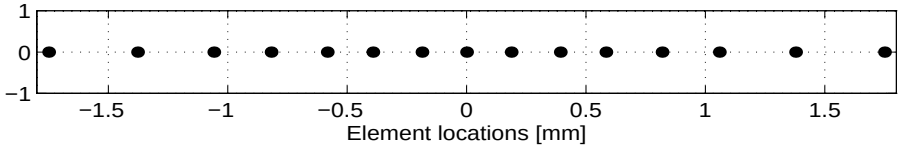
the synthesis process of ultrasonic arrays.
These methods will not be given any thorough discussion here, but some selected examples will be simulated and short comments on the results will be given. In that way we will also show that by applying our generalized optimization routine, we can further improve the sidelobe level.

5.3.1 Optimization of sensor locations

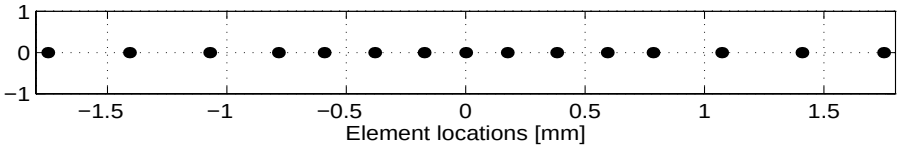
A family of optimization algorithms appearing in the literature, use uniform or partial-uniform weighting and do the optimization on the sensor locations solely. The optimization of weights is a linear problem, whereas the problem of optimizing the sensor locations will be nonlinear. Consequently one has to use non-linear optimization methods or by using linear approximation methods.
Schjær-Jacobsen *et al.* [50] have proposed a nonlinear minimax optimization method for minimum sidelobe synthesis of linear arrays by varying the element spacings, while keeping a fixed weighting function equal to unity at all sensor locations. It is an iterative technique, where the location vector, $\vec{x}_k$ is incremented in each iteration and updated to this new value if sufficient conditions are met. The step length is adjusted during the iteration process.
From now on we call this array, the 'Schjær array'. These arrays from will have the outer sensors located at a distance equal to $\frac{N-1}{4}\lambda$ from the origin of the array. Not surprisingly the aperture of the Schjær array is equivalent to the equi-spaced array's and also have the same number of elements. Because of this fact we can now simulate an array of this kind, with non-equally spaced elements and compare its response with the equi-spaced array's response.



(a) Equi-spaced array



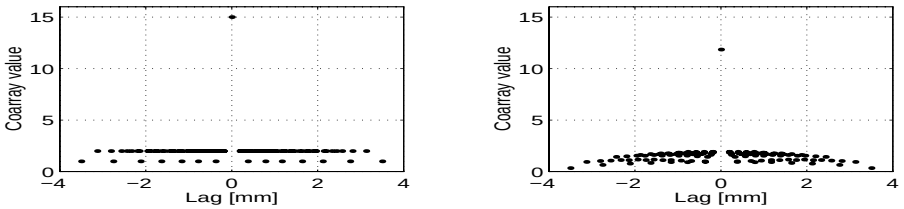
(b) The Schjær array



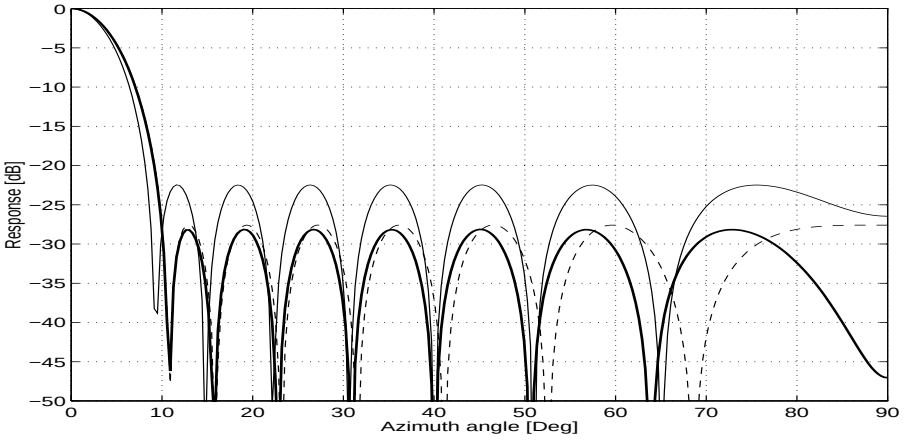
(c) The Sandrin array

Figure 5.3: This figure shows the element locations on the arrays used in the succeeding examples of simulations in this section.

It is worth noting that the uniformly weighted Schjær array has a maximum sidelobe level at -22.5dB. Thus it is substantial better than the uniformly weighted equi-spaced array. This implies that by designing arrays with more elements according to the method in [50] will give even better sidelobe appearance.



(a) Coarrays for the Schjær array

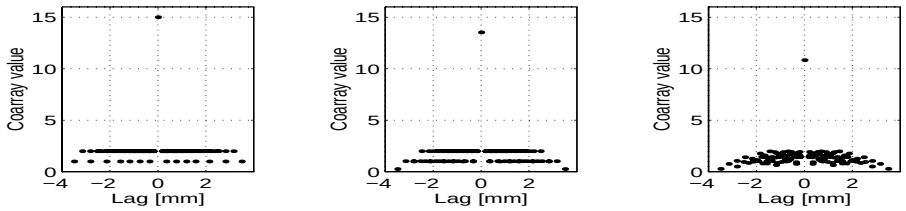


(b) Beampatterns

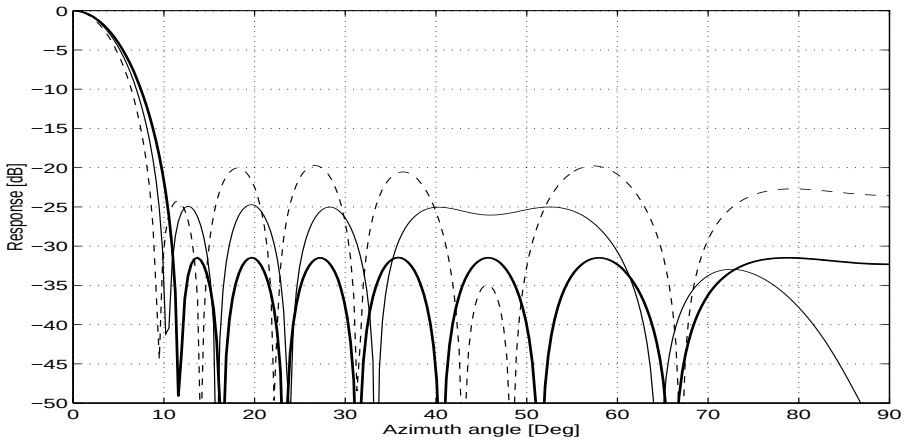
Figure 5.4: Simulations with a 15 element array optimized with respect to the sensor locations by the method of [50]. An equi-spaced and optimal weighted array with equivalent aperture is plotted in the same figure (with dashed line). The irregular

spaced array is optimized with our Remez method and the result is drawn as a boldface curve. All responses are calculated by our coarray routines.

Other authors have adressed the problem by considering nearly-uniform tapering of the array elements, and as a second example we give one of these, namely a 15 element array proposed by Sandrin *et al.*[49]. The arrays in the article of Sandrin *et al.* are very special cases (and a bit odd), where the arrays are designed with a nominal spacing of '2λ'! This is motivated by the fact that the intended application is an antenna whose sub-aperture beamwidths are less than $\pm 15^{\circ}$.
For our applications we therefore multiply the sensor location values by the factor 0.2277 (and not 0.25!) to make an array with mean spacing equal to one-half the wavelength.



(a) Coarrays for the Sandrin array



(b) Beampatterns

Figure 5.5: Simulations with a 15 element array optimized with respect to the sensor locations of [49], with partial uniform weighting (thin line drawing). Results from simulating the same array with no weights is plotted with a dashed curve. Finally, the array is optimized with our Remez method, and the result is drawn as a thick line. All responses are calculated by the coarray approach.

In the article they have divided the location values by four to get what they have called a '$\lambda/2$-spaced' array. However, using the given values suggests a mean spacing of 0.549λ!. Thus there is discrepancy between the nomenclature in this thesis and the one in the article of Sandrin *et al.* [49].
We have simulated the response from the 15 element Sandrin array with a scaling corresponding to our definition of mean spacing and the results are given in figure 5.5. The coarrays are also included, since we use them to calculate the responses.
First we have simulated with an unweighted array and the coarray is shown to the left in figure 5.5. The angular response of this array is seen to outdo the equi-spaced array in the unweighted case.
Next we have used the proposed weighting of [49] i.e $w_n = 0.515$ on the end elements and unity weights for all other elements. The coarray is shown in the middle of figure 5.5.Finally, we have optimized the array with our routine.
Despite using a different down-scaling the results resembles what is reported in the article. We believe this is due to the array having a low sidelobe level continuing into the invisible region.
In table 5.1 the beamwidth and sidelobe level of the Schjær array is compared to an equi-spaced array and the Sandrin array [49],each of equivalent aperture. The Schjær array is seen to give the lowest sidelobe level. Moreover it also have the lowest dynamic range of the weight coefficients among the three tested arrays. This is another indication of it being a well-defined geometry with respect to its approximation to an ideal spatial response.

Array type	**-6 dB beamwidth**	**Max. sidelobe**	Δ_D
Non-weighted arrays			
Equi-spaced	$9.23°$	-13.15dB	1.0
Schjær array	$10.34°$	-22.46dB	1.0
Sandrin array	$10.28°$	-19.7 dB	1.0
Sandrin's nearly uniform weighting proposal			
Sandrin array	$11.18°$	-24.7 dB	1.94
Optimized with our Remez algorithm			
Equi-spaced	$11.24°$	-27.6 dB	2.79
Schjær array	$11.24°$	-28.2 dB	1.71
Sandrin array	$11.24°$	-27.6 dB	1.87

Table 5.1: Key numbers of the responses plotted in figure 5.4 and 5.5. The dynamic factors for the optimized weights given in the lower table.

Another interesting thing can be noted from the coarrays resulting from the uniform weighted and the optimal weighted array, respectively. The main difference between these coarrays, is that the optimal weighted

has values decreasing with increasing lags. Both coarrays have a relatively smooth appearance with a very homogeneous distribution of lags. This might indicate that the array belong to a class where applying low weight values on the ends will result in lowered sidelobes. A similar effect is observed with the Sandrin array, where reducing the weight value on the end elements to 0.515 reduces the maximum sidelobe level with 5dB.
Simulations with steering of the Schjær array has unveiled broadening and distortion of the main beam as well as an increased sidelobe level at the opposite side of the steered beam's mainlobe. This is observed even for steering at narrow angles. Thus the array demands optimization of locations for each steering angle as is a similar effect to what is demanded with optimization done on the weights (i.e. the weights must be optimized for each steering angle).

5.3.2 Simultaneous location and shading optimization

Methods for simultaneous optimalization of the sensor locations and tapering have also been proposed [38] and [30].
Due to the lack of time, these methods have not been considered in this work. It must though be noticed that these arrays probably not are passable solutions for ultrasonic arrays, because of their nonlinear sensor locations. Their results will undoubtable be good, when both locations and weights are optimized at the same time. And for larger apertures than we consider, they will probably offer excellent spatial response patterns.
Often these methods starts with a few sensors and one and one is added in every step until a desired number is reached. One should be aware that some methods referred to as optimal, not optimize the already existing sensor locations when adding a new element to the set. Thus the 'optimality' stated must be considered as a partial optimal solution. Mitrou [38] optimize previous taps as well as the new one added.
We expect that the optimization must be performed for each steering angle for these methods as is the case for the previous discussed optimization methods that deal with one of them at the time.

5.4 Summary

In this chapter a new optimization algorithm has been proposed that can be used to find optimal weights even for arrays with a non-equal intersensor spacing. It is based on a generalization of the Remez exchange method to a general spatial frequency grid. The method as such, has similarities with the optimal linear phase FIR-filter algorithm of Parks

and McClellan [44] and even more with Streit's algorithm [57].
The developed algorithm has been formulated and described in this chapter. The differences between our method and the equi-spaced algorithm in [37] have been mentioned. Streit's routine has not appeared in detail, so we cannot be too specific with respect to similarities and differences with our routine. We use the same terminology with respect to the beampattern and both use the Remez exchange algorithm, so we can say that similarities at least exist. However, our optimal weight values are slightly different from Streit's. Moreover, our routine seems to give slightly better results for the beamwidths considered by Streit. This is reported in section 6.2.1.
Finally in this chapter, some results have been reported that have shown our routine's usefulness. Angular responses from two arrays, one with optimal sensor locations and one with near-optimal sensor locations, are seen to be improved with our method. An interesting point from these two arrays is that both arrays had elements more densed near the array origin than at the array ends. This is similar to the effect observed when thinning an array. Because then the anti-Gaussian thinning was the one performing best in the non-weighted case (section 4.5.3).

In the next chapter we will use our routine to optimize arrays with various element configurations. Thus we will try to determine our method's limitations and advantages.

Chapter 6

Computer simulations

This chapter is devoted to descriptions of the simulations and results obtained from the simulations that have been performed with the developed routines. The goals of the computer simulations are threefold. Firstly, we want to verify our algorithms and methods by using test data with known solutions. This implies reproducing results reported in the open literature.
Secondly, we will compare different arrays and different configurations of elements in order to find properties or characteristics that are similar or distinct between different types of arrays.
And last but not least, we want to use the simulations as support for our conclusions.
Different arrays will be investigated and optimized with the newly developed optimizing routine.In this part of the work we had need for a tool that could thin and alter element locations. The tool that was made and implemented will also be described briefly in this chapter.
At the end of this chapter we summarize the results made in these simulations.

6.1 Simulations with equally spaced arrays

We will first report some results from simulations with optimized equispaced arrays. The intention with these simulations is to find tendencies between different characteristics or parameters of the arrays.
We will take a look at the relationship between the maximum sidelobe level and the resolution, the relation between the dynamic range of the weight coefficients, and the resolution or maximum sidelobe level. These relations are visualized in the figures 6.1, 6.2 and 6.3.
Figure 6.1 displays an almost linear relation between the sidelobe level and the resolution. We see that increasing the number of elements is

equivalent to lowering the maximum sidelobe level for a given resolution. The inclination of the curves in fig. 6.1 is increasing with the number of elements as well. Thus we observe the effect that an increased number of sensors, giving a larger aperture, permits a better resolution while it maintains a low sidelobe level.

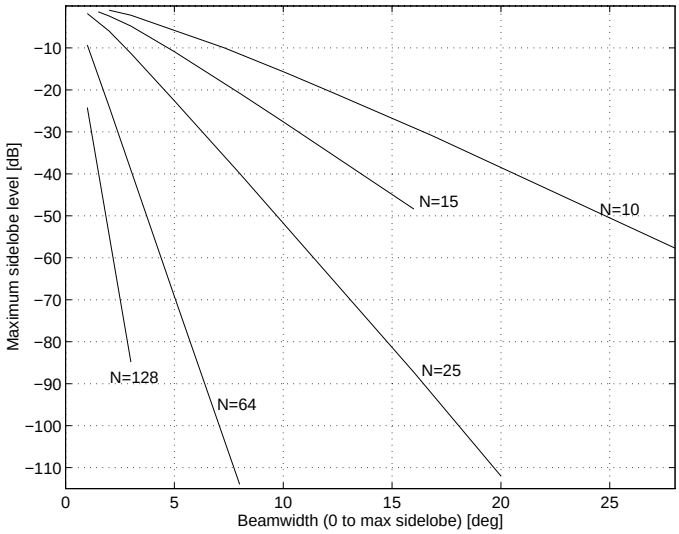


Figure 6.1: The figure show the relation between the beamwidth and the maximum sidelobe level for an equi-spaced array, optimized with our general Remez algorithm. The beamwidth is measured from $0°$ to the angle where the maximum sidelobe level is reached. It is thus not the usual -6 dB beamwidth.

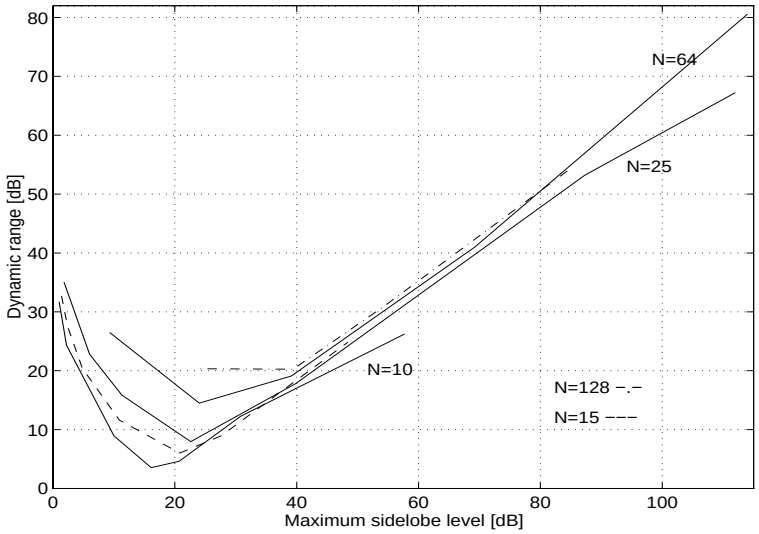


Figure 6.2: The figure shows the relation between the weight dynamic range and the maximum sidelobe level for an equi-spaced array optimized with our general Remez algorithm.

In figure 6.2 the relation between the dynamic range of the weight coefficients, Δ_D, and the maximum sidelobe level is shown. For det most part

the relations are similar for the different number of sensors. The Δ_Ds have minima for given sidelobe levels and these level values increases with the number of sensors. A characteristic point with these curves is that the deviation between them is relatively small. Obviously, this can at least partly be explained by the fact that we have plotted the values with logarithmic axes.

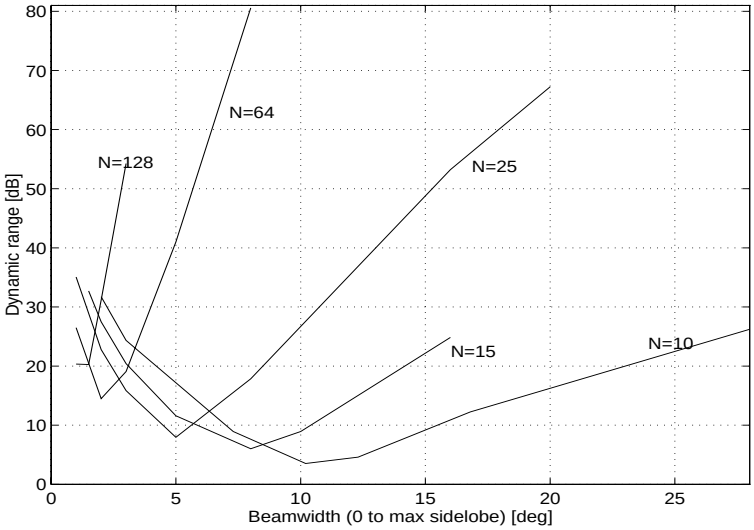


Figure 6.3: The figure shows the relation between the dynamic range and the beamwidth for an equi-spaced array optimized with our general Remez algorithm.

The last figure (Fig. 6.3) shows the relation between the dynamic range of the weight coefficients and the first value of the response where the maximum sidelobe level is reached (i.e. called the beamwidth in this section). To a great extent the curves are similar to the ones in the preceding figure (Fig. 6.2). A larger deviation of the curves are observed, but this is probably because we use a linear scale for the beamwidth. It is observed that the minima reached for the Δ_Ds appears at increasing angles when the number of elements increases.

In this section we have observed that there is an almost linear relation between the beamwidth and the maximum sidelobe level. By accepting this the maximum sidelobe level at a particular beamwidth, of an array can be read from a figure similar to figure 6.1. It is also clear that lowering the sidelobe level will decrease the beamwidth and hence reduce the imaging resolution of that array. Demanding a narrow beamwidth will equivalently result in a higher sidelobe level.
From the preceding three figures it is also clear that the dynamic range of the weight coefficients, Δ_D, are low only in a limited interval of the sidelobe (and beamwidth region). Thus when there is a limit on this value one must assure oneself that the chosen beamwidth and hence the maximum sidelobe level, is within the proper interval. It is observed that a high number of elements generally have higher values on the Δ_Ds.

The largest array that was simulated had 128 elements. Our routine displayed very high condition numbers with this array in general and with a desired cut-angle equal to 4° it quickly diverged. The same were then observed with an array of 64 elements, when a desired cut-angle equal to 9° was input. By trying different angles we observed that the divergence appeared when the δ value decreased below a value around $-220 - 230dB$. A routine more similar to Parks and McClellan's [37] did not diverge at this angle but the responses that was output had one (!) sidelobe with a maximum level of approximately $-20dB$! In other words an ill-conditioned system caused completely unreliable results.
Thus one limitation with our routine[1] were revealed in these simulations, that apparently is caused by a numerically ill-conditioned matrix system.

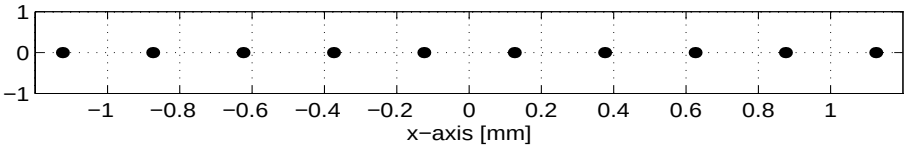
6.2 Simulations with non-equally spaced arrays

We will now proceed from the apparently trivial case of equally spaced arrays and consider the more untrivial case of non-equally spaced arrays. We will be considering three different kinds of such arrays. The use of non-uniformly spaced arrays can lead to a reduced number of elements and thus lower cost, without degrading the array's performance. Arrays optimized by the element locations were seen to have good results in the previous chapter.
First we take a look at two arrays derived from Gaussian quadrature formulas. One of them have 10 element and their location are given in figure 6.4 as well as in table 4.2 and the other have 25 elements and is given in [56].
Next, two arrays originating from geometric series are considered. The first with a cardinal number larger than one and the second with a cardinal number less than one. These two arrays will be seen to have very distinct responses in the unweighted case.
Finally, two perturbed arrays are optimized. The first is a special perturbation, where an array with 129 elements and an inter-sensor spacing of $\approx \lambda/4$ was thinned, with an uniform random distribution, until its mean spacing was $\lambda/2$. The second array of these two, is a 25 element equi-spaced array that is perturbed by displacing its element locations in Gaussian way. Its mean spacing is $\lambda/2$ and its inter-sensor distance can be no more than λ according to the restrictions posed in the routine.

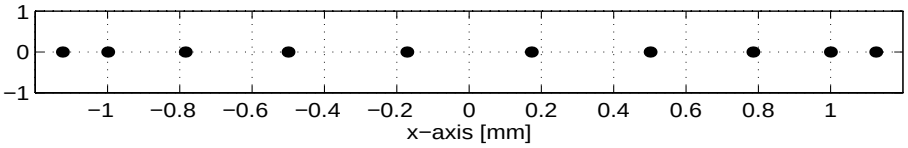
[1] as well as a routine closely related to the one in [37]. This routine is mentioned in the user's manual in Appendix B.

6.2.1 Locations based on Gaussian quadrature formulas

These arrays have both been proposed by Streit [57, 56] and the origin is described in section 4.5.1. They seems to be well-behaving arrays, with respect to resolution and sidelobes. However, their dynamic range of weight coefficients , Δ_D, is generally higher for these arrays than was the case for the equi-spaced arrays. Generally they also have a slightly higher sidelobe level than the equi-spaced arrays.
Now turning to the figures, one should first note the differences between the element configurations of the $\lambda/2$-spaced and the 10 element Gaussian quadrature spaced array. In the upper panel of fig. 6.4 we observe the regular spacing.



(a) Equi-spaced array



(b) 10 element Gaussian quadrature array

Figure 6.4: An equi-spaced array and the array proposed by Streit [57].

An 'inconvenient' excitation frequency, $f = 3.08MHz$, has been chosen. Because the propagation velocity of the media is $1540m/s$, the wavelength is $0.5mm$ and thus the inter-element spacing can be observed to be $d = \lambda/2$ in the figure[2].
The middle panel of fig. 6.4 shows that the non-equally spaced array is a displaced version of the first with equivalent aperture, but with element spacing decreasing towards the ends.
The unequally spaced array is optimized with our general Remez algorithm and the results are plotted in figure 6.4. One should observe that an additional sidelobe develops towards $\phi = 90°$, when the cut angle increases towards $20°$. Increasing the angle beyond $30°$ will result in the development of a grating lobe at farther angles. We expect this is due

[2] Note that the first array elements are spaced $\lambda/4 = 0.125mm$ from the array origin.

to the array being of even length as the similar is observed with several other even length arrays.

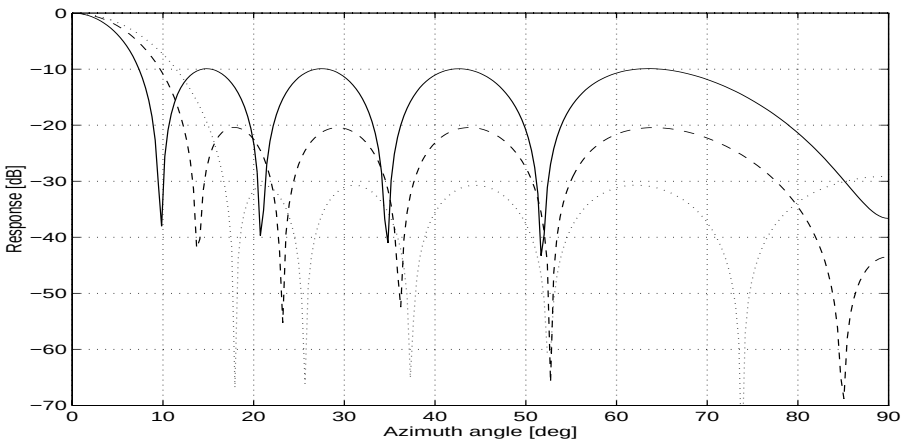


Figure 6.5: Simulations with Streit's proposed array. This is an output from the general Remez algorithm of UltraSim. The results resemble the figures in [57].

An equi-spaced Dolph-Chebyshev array is also tested for the same initial cut-angles. The performance of the two arrays is compared and the results for some chosen angles are given in table 6.1.

u_0	ϕ_i	**Dolph-Chebyshev**			**Streit array**		
		ϕ_{BW}	SL	# I	ϕ_{BW}	SL	# I
0.40138	7.34°	9.51°	−10.01dB	5	9.51°	−9.934dB	5
0.5563	10.20°	11.66°	−16.08dB	4	11.66°	−15.90dB	5
0.6693	12.30°	13.77°	−20.67dB	4	13.77°	−20.44dB	4
0.9080	16.80°	17.55°	−30.78dB	5	17.55°	−30.73dB	3

Table 6.1: Optimization of two 10 element arrays with aperture $A = 5\lambda$ and for three chosen initial cut-angles, ϕ_i. One Dolph-Chebyshev array and one non-equally spaced array given by Streit. ϕ_{BW} is the beamwidth, measured from 0° to the angle where we find the first zero crossing. SL is the sidelobe level of this beamwidth. And the number of iterations before the routine converged, $\#I$, are indicated as well.

An interesting point is that the performance of the two arrays to a great extent resembles. The equi-spaced is only a few decimals better and these results are similar to the ones made by Streit[57]. It is noteworthy that the ϕ_{BW} reported here, is not the usual −6dB beamwidth, but the angle where the response has its first zero.

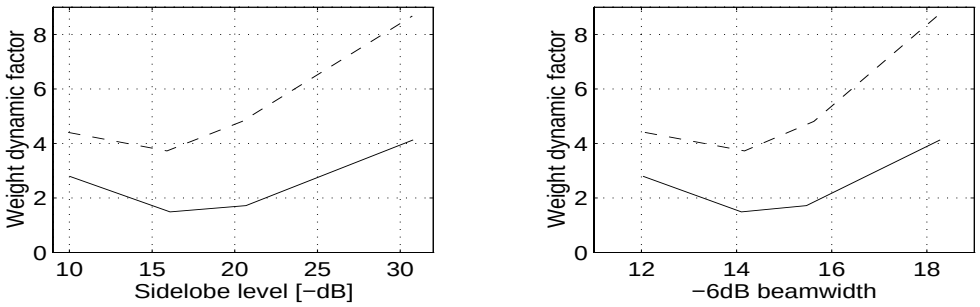
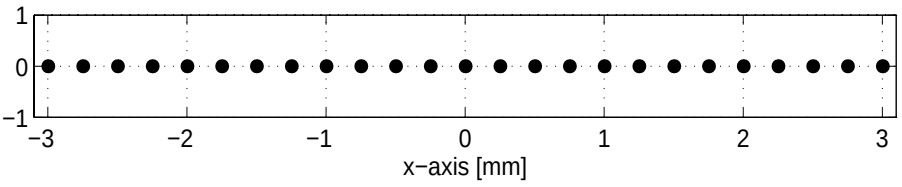


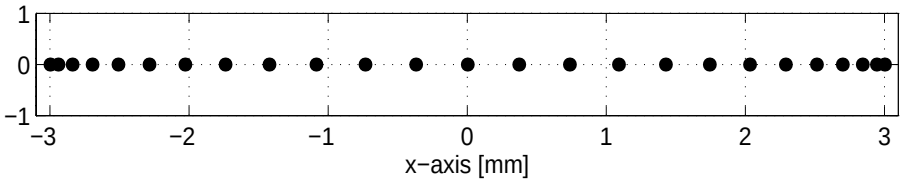
Figure 6.6: The dynamic factor versus the maximum sidelobe level and the $-6dB$ beamwidth is plotted in this figure. Note that in this case the values of the Δ_Ds are plotted with their actual values, not their logarithmic values. The non-equally spaced array is drawn with the dashed line.

From the latter figure, we observe that the relations between the dynamic range of the weight coefficients and the sidelobe level and beamwidth, display relations similar to the curves for the equi-spaced arrays. The difference consist in that the minimum values for the unequally spaced arrays are generally larger.

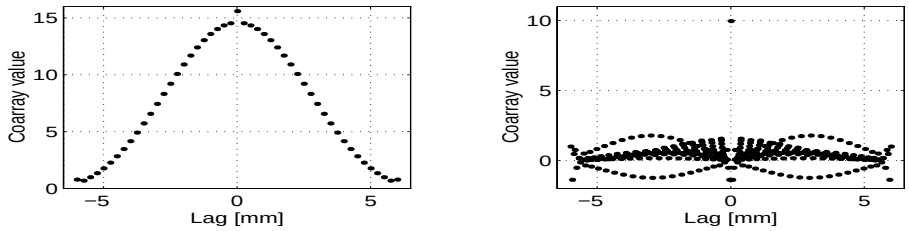
The next array has 25 elements and they are also located according to the nodes of a Gaussian quadrature formula [56]. We observe the difference between the location of elements in the two arrays in figure 6.7. For some choices of cut-angles this array needed a more densed grid to converge. However, the dynamic range of the weights was still relatively small. The unweighted arrays made from Gaussian quadrature nodes have been observed to have a regular angular response, with sidelobes slowly decreasing with higher angles from a maximum sidelobe level of about -10dB.



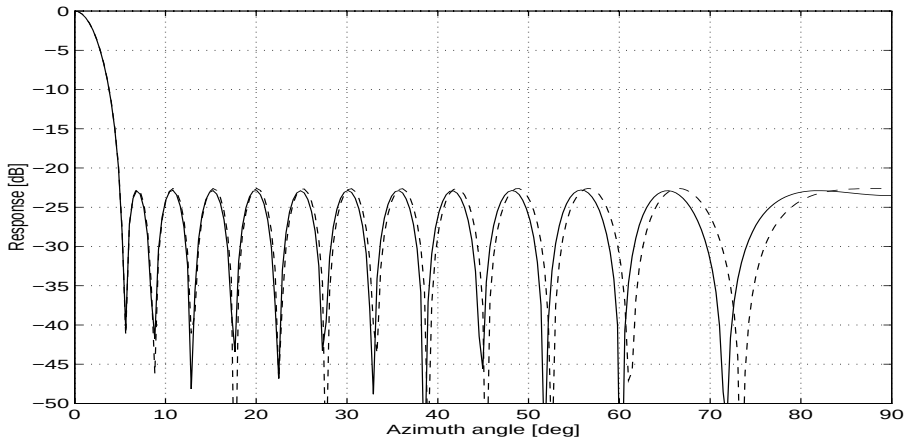
(a) Equi-spaced array



(b) 25 element Gaussian quadrature array



(c) Coarrays, equi-spaced to the left



(d) Beampatterns for the unequally spaced array

Figure 6.7: Simulations with a 25 element Gaussian quadrature based array (proposed by Streit in [56] and an equi-spaced array with the same aperture and number of elements. The upper panels show the elements in the arrays, the middle panel their coarrays and the lower panel displays the angular responses of the arrays.

The most interesting thing to note in the latter simulations is the striking differences between the two coarrays that nearly have an equivalent angular response. Thus we have seen an example of the difficulties that arise when trying to use the coarrays in the design procedure. Because the angular responses have such similarities, while the coarrays are very different, the differences in response must appear in the phase. This is normally not shown for angular responses.

Another observation made is that the inter-sensor spacing decreases towards the array ends and this is the opposite of how the spacing of the optimally located sensors developes.
It can thus be stated that the Gaussian quadrature arrays have a less good unweighted appearnce than the equi-spaced array, because its inter-sensor spacing is decreasing and not increasing as it does for the location optimized arrays. The equi-spaced are inbetween these two types of element locations.
The same effect is shown by the incresing level of the dynamic range of the weight coefficients of these arrays.

6.2.2 Locations determined from a geometric series

In this section we consider two arrays with a geometric series location of the elements. One should note that these arrays are similarly referred to as logarithmic arrays. By choosing a cardinal number for the geometric series the inter-sensor spacing and hence the sensor locations, are easily calculated.
The two 25 elements arrays are shown in figure 6.8. The upper panel shows the array with cardinal number greater than one i.e. $a = 1.05$, and the lower with $a = 0.97$.

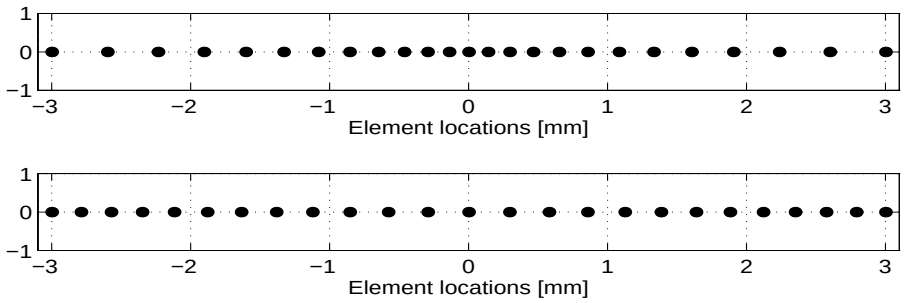


Figure 6.8: This is two examples of arrays with element locations determined from a geometric series.

The response of the unweighted array in figure 6.9 displays a relatively low sidelobe level. Using the optimization routine on this array has not improved the sidelobe level by more than $6 - 8$ dB at the nearest sidelobes. For sidelobes at higher azimuth angles, the optimized array has a higher sidelobe level than the unweighted array. We expect this to due to the unweighted array's initial flatness in the sidelobe region. One could view[3] the equiripple appearance of the optimized array's as the effect from lowering the high sidelobes while lifting the low one, and adjust to a level where we have equality. From this point of view it is clear that

[3] This is, to a certain degree, a simplification of what is happening.

the initial response pattern does not give much possibility to compensate for decreasing the highest sidelobes.

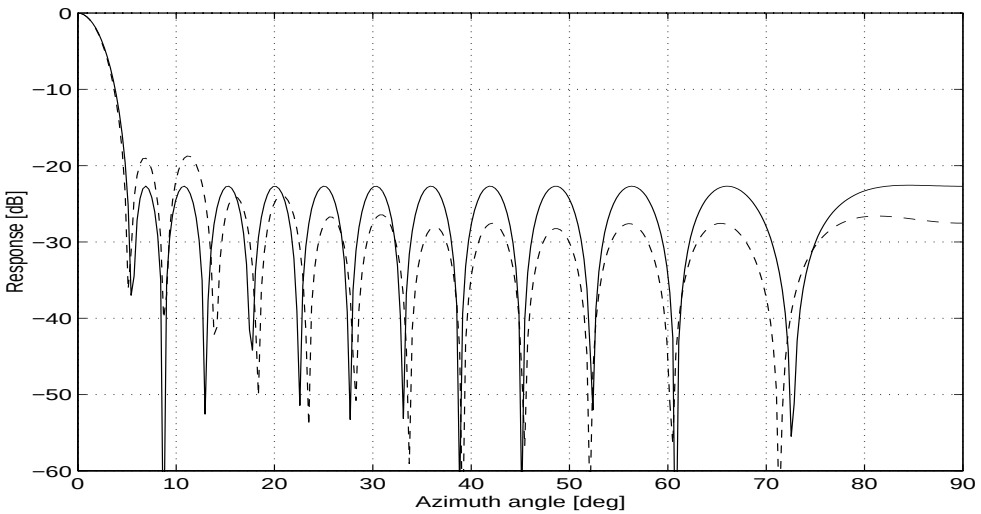


Figure 6.9: A 25 element array from a geometric series with $a = 1.05$ has been optimized. Its angular response is shown in this figure, with the response of the unweighted array as a dashed curve and the optimal weighted with solid curve. The corresponding array is shown in the upper panel of figure 6.8.

We note that this array has an increasing spacing between its elements as was also the case for the arrays optimized with respect to element locations. Thus this array's low sidelobe level in the unweighted case, may be due to its initial well-defined element locations.

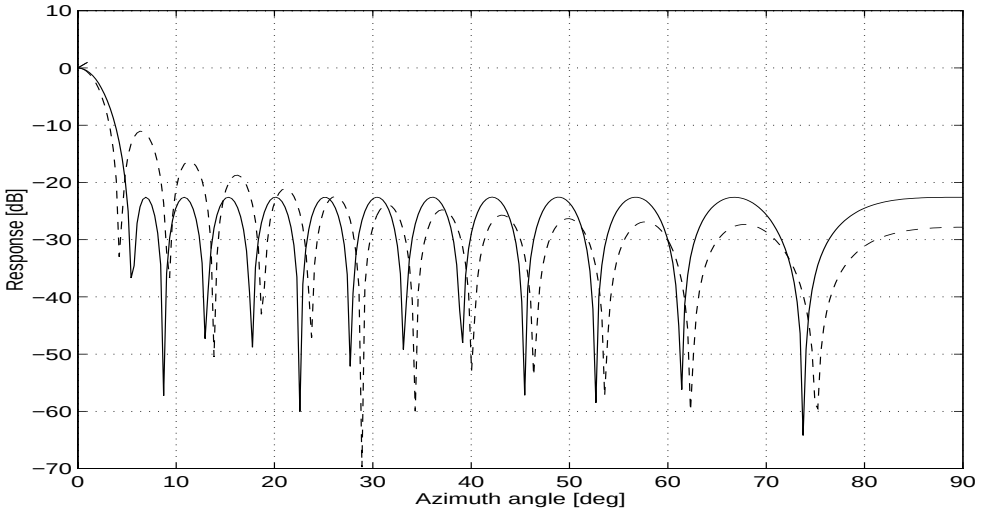


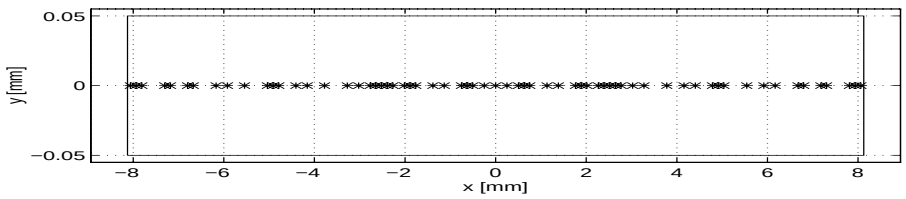
Figure 6.10: A 25 element array from a geometric series with $a = 0.97$ has been optimized and its angular response is shown in this figure. The corresponding array is shown in the lower panel of figure 6.8.

In figure 6.10 an array with inter-sensor locations decreasing towards the ends is shown. In the unweighted case the angular response has similarities with the response from an equi-spaced array. However, the array simulated here has a higher sidelobe level than the equi-spaced would have. One should also note the resemblance with the 25 element array in figure 6.7. In simulations we have observed that the angular responses of these two arrays are quite similar at narrow angles when the cardinal number is chosen to be less than one e.g 0.95. The differences are larger at higher azimuth angles.

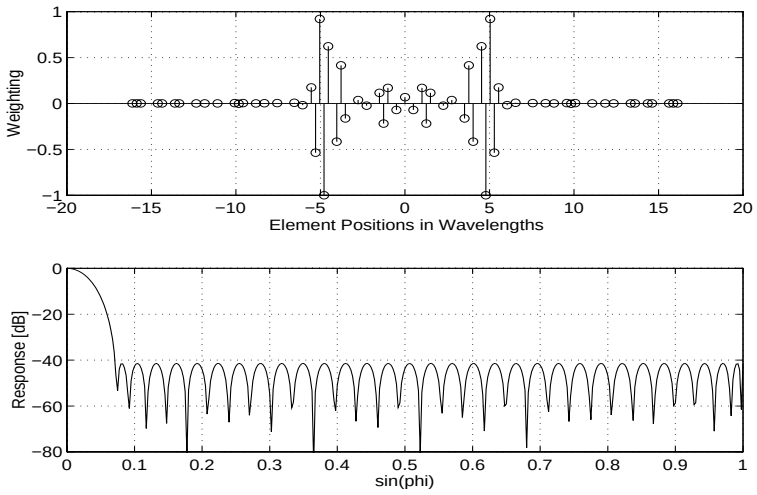
The arrays determined from geometric series that have been simulating in this work had convergence problems for some choices of cut-angles when a cardinal number too far from unity was chosen. In some cases it then was enough to increase the oversampling rate above the default value of 16, to produce convergence. In spite of these problems, their optimized responses obtained were similar to the other 25 element arrays being simulated and the dynamic ranges of the weight values were relatively low.
An interesting observation was done for the dynamic range of the weight values that coincides with the former observations with optimally located and Gaussian quadrature located arrays. The weight values for the arrays with decreasing element spacing was ***always higher*** than the arrays with increasing inter-element spacing.

6.2.3 Optimizing perturbed arrays

These arrays are very special cases of element configurations and will only briefly be discussed in this text. The first array considered in this section have been thinned from an originally 129 element array. Key numbers of the simulations with this particular array are given in table 6.2.

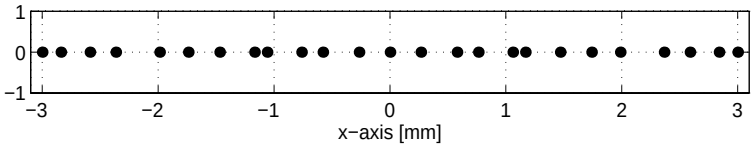


(a) A perturbed 65 element equi-spaced array

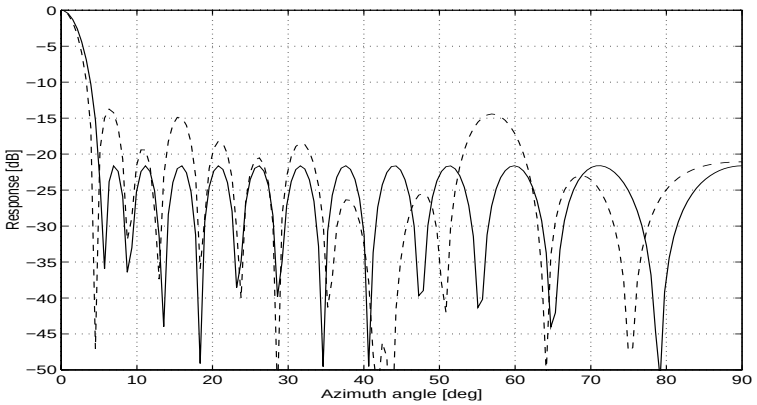


(b) A perturbed equi-spaced array

Figure 6.11: A linear array perturbed by thinning an array with a large number of elements down to a mean spacing of $\lambda/2$.



(a) A perturbed 25 element equi-spaced array



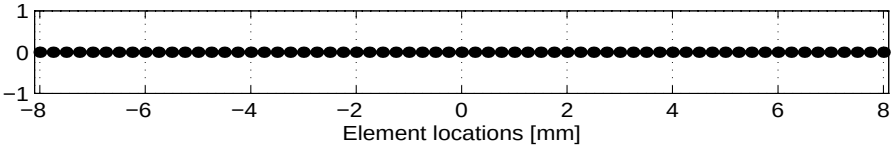
(b) Beampatterns

Figure 6.12: A linear array with element locations perturbed by a Gaussian random displacement routine. It has a mean spacing of $\lambda/2$.

Our method allows a lower sidelobe level to appear with a resolution as good as the comparable arrays have. However, this is done at the expense of an enormous dynamic range of the weights. In the first figure (fig. 6.11) the element configuration is shown in the upper panel, the optimal weights in the middle panel and the response in the lower panel. The large dynamic range unveils the unweighted array's a very poor response. This array would not be used in ultrasound equipment, due to its tiny minimum inter-sensor spacing and its large dynamic range of weights.
The next array (figure 6.12) is produced by a routine, that has altered the locations of the individual elements within $\pm\lambda/4$ of their original uniform inter-sensor spacing, according to a Gaussian distribution. Thus the 'dislocation' of the elements of this array, compared to the equi-spaced, is limited. The limit of this displacement routine must be seen in contrast to making a perturbed array with by thinning elements, who often will give a irregular configuration of elements. This is revealed by the dynamic range, that is generally lower with the arrays perturbed by limited displacement.

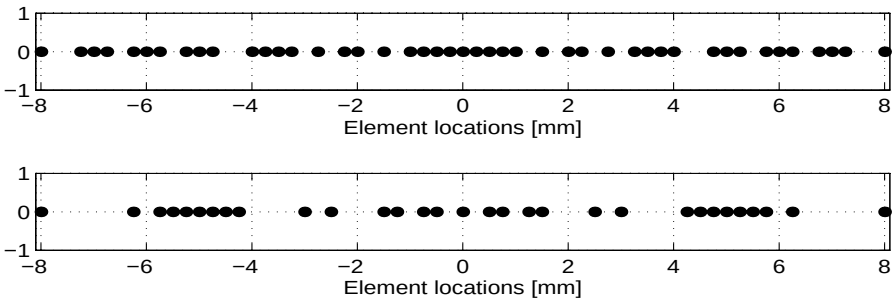
6.3 Optimization of sparse arrays

As mentioned earlier in this thesis sparse arrays have some very useful applications. In this section we have done simulations with thinning of arrays, from 30% to more than 50% with respect to their initial number of sensors. Our initial array is an equi-spaced 65 element array with an aperture of 32.5λ. The thinning was done by removing the elements with locations equal to the values drawn from a random distribution. This was done such that the remaining elements were symmetrically placed around the array origin.
We have used an 'anti-Gaussian'[4] as well as an uniform thinning procedure. Various thinning procedures are implemented in a developed thinning tool case (see section 6.4). Most of the results for sparse arrays in this section have also been published [13].
The following arrays were used in this section:

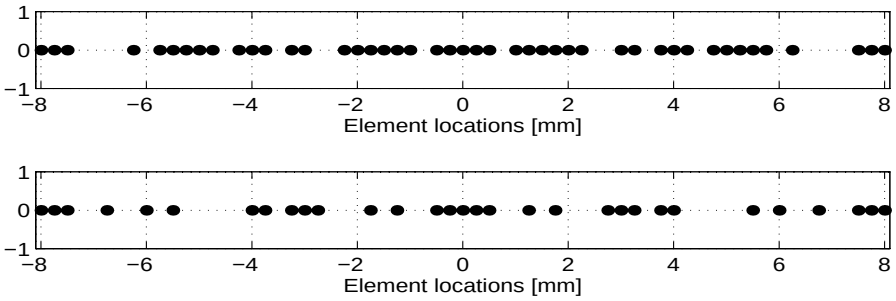


(a) The equi-spaced array

[4] The farther from the array origin an element is, the larger the probalility for being removed



(b) Two sparse arrays anti-Gaussianly thinned



(c) Two sparse arrays thinned uniformly

Figure 6.13: The upper panel arrays in (b) and (c) have 45 elements each, the lower panel arrays have 31 elements each. These arrays have been optimized with our method and key numbers from the simulations are reported in table 6.2.

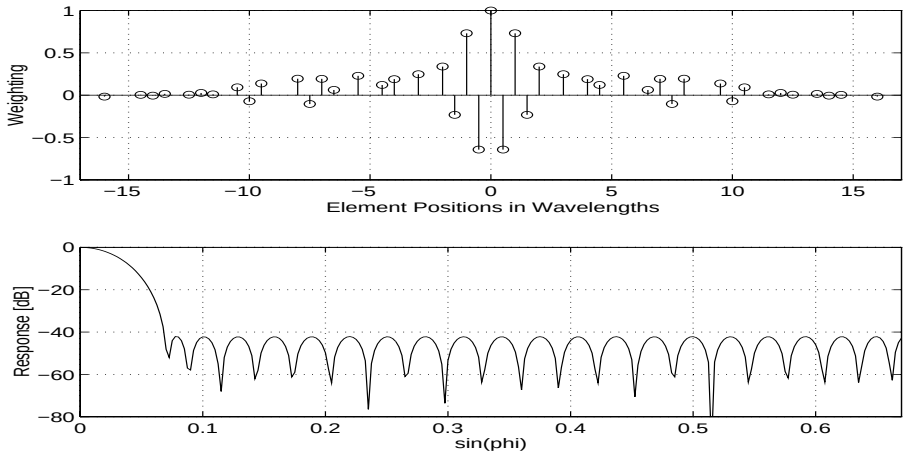


Figure 6.14: Weighting and response of a 65 element equispaced array with aperture 32.5λ thinned in a Gaussian way to 45 elements (shown in the upper panel of figure 6.13b.

In figure 6.14 the optimal weighting function and the angular response

within the optimization region, is given. We observe the equiripple appearance of the sidelobes that is such a characteristic property of Remez algorithms and that a low sidelobe level has been obtained. The results attained for the sparse arrays are reported in table 6.2. An immediate observation from this table is that the sidelobe levels for the various arrays are almost completely matched. In fact the maximal discrepancy is only 1.4dB and is between the 45- and the 31 element anti-Gaussianly thinned array. In contrast to this similarity, the variations of the dynamic weight ranges between the arrays, Δ_D, are extremely large. This number indicates how much the unweighted response deviates from an ideal pattern. From this number we observe that the array perturbed with a thinning routine are definitely 'bad case'.

Type	Beamwidth $-6dB$	Dynamic Range	Sidelobe Level
65 Filled	$3.0°$	20.6 dB	-41.3 dB
65 Perturbed	$4.2°$	84.9 dB	-41.5 dB
45 Gauss	$3.9°$	47.5 dB	-42.2 dB
45 Uniform	$4.3°$	53.8 dB	-41.3 dB
31 Gauss	$4.2°$	52.0 dB	-40.8 dB
31 Uniform	$5.8°$	65.7 dB	-41.5 dB

Table 6.2: Results for the arrays in figure 6.13 with respect to -6dB beamwidth, maximum sidelobe level and dynamic range of the weight coefficients.

Moreover, the beamwidths are observed to increase with stronger thinning. The beamwidth is almost doubled from the equi-spaced array to the 31 element uniformly thinned array. This signifies problems with obtaining high resolution with heavily thinned arrays (when a very low sidelobe level is still wanted).

The main result of our work in this section is that the sidelobe level of a sparse array can be optimized to give responses that resemble filled equi-spaced Dolph-Chebyshev shaded arrays. However, when using a Remez exchange algorithm based routine the optimization region has to be reduced in order to maintain the superior control over the sidelobes. In other words, these methods give the opportunity to subdue a particular number of sidelobe peaks and this number equals the number of remaining elements in the array.
In figure 6.15, an array thinned about 52% (i.e. 34/65) is given. From the thinning rate (i.e. the rate between the number of thinned elements and the number of elements in the filled grid), we know that the optimization now will be limited to the region from $0°$ to about $30° \Leftrightarrow \sin(31/65) \approx 0.46$ (Eq. 5.6) and the response in this figure is shown in this region.

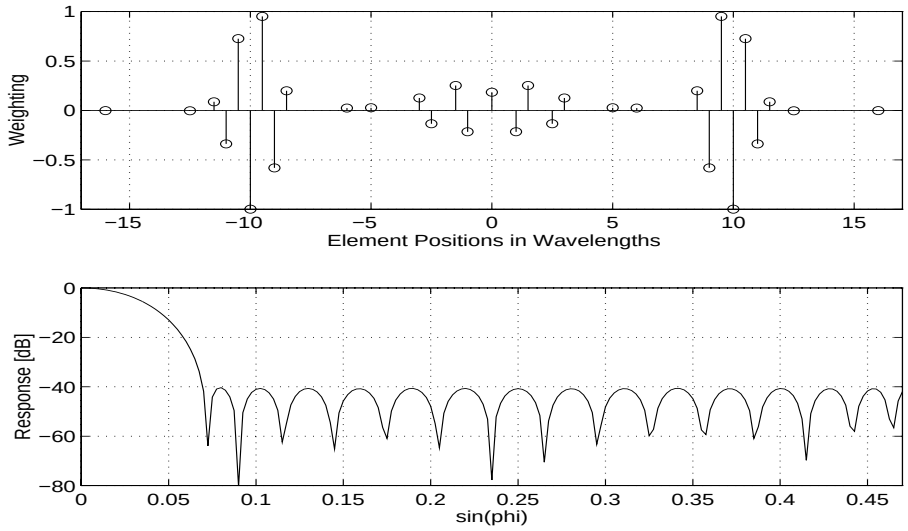


Figure 6.15: Weighting and response of a 65 element equispaced array with aperture 32.5λ thinned to 31 elements in a Gaussian manner.

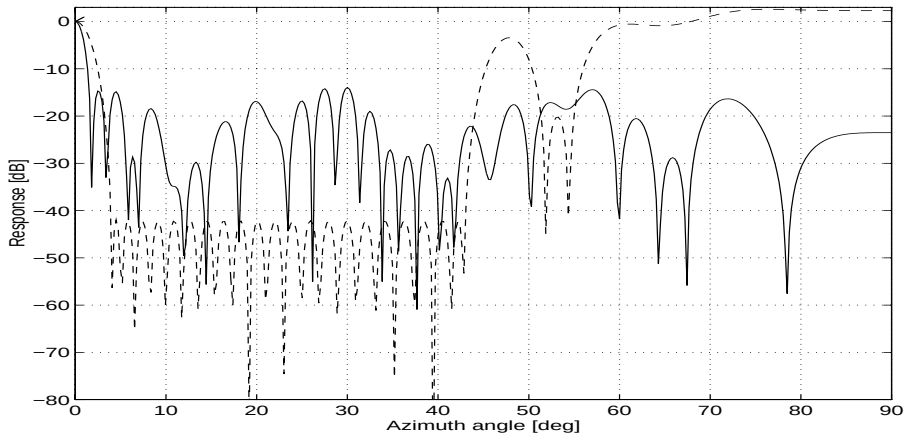


Figure 6.16: Comparison between unweighted (solid line) and weighted (dashed line) response for 65 element array thinned to 45 elements according to a Gaussian distribution. Shown over the entire positive azimuth domain.

The limitation of the optimization region is clearly observed in figure 6.16, where an array that is thinned 30% (i.e. it has 45 elements left out of an initial number of 65) has been simulated and accordingly is optimized only at azimuth angles below $\arcsin(45/65) \approx 44°$.
The optimal weights are seen to have a strong impact on the sidelobe level, which is lowered almost 30dB. A grating-lobe like response is observed in the region $50° - 90°$, that also displays the formerly mentioned limitation of the optimization region.

6.4 Thinning and relocating array elements

During the work on this thesis, a program that could thin arrays in various ways quickly was needed. Later the need for altering sensor locations and placing sensors with spacing determined by a geometric series arose. In this way an initially simple thinning program increased in both number of functions and size. The basic philosophy with these routines has been to make a simple and interactive user interface. Short commands are therefore given interactively, while parameters (e.g. random seed numbers and angular values) are given in the MATLAB command window.
The available functions will now be presented briefly.
Firstly, interactive removal of individual elements from linear and two-dimensional arrays are allowed. Additionally, the user can remove rows, columns and rectangular areas from a two-dimensional array, by just 'clicking' mouse buttons.
The part of this program that has been most widely used in this work, is the *random thinning* part. There *uniform*, *Gaussian* and what we have called *anti-Gaussian* distributions have been implemented. The latter distribution is an upside down Gaussian distribution. In other words the probability of picking a value is largest at the ends.
Another function is to place sensors in a pattern according to a geometric series. The resulting array will have aperture and number of elements equal to the input array. Thus the mean inter-sensor spacing will not be altered by this routine.
Additionally, a routine for displacing the elements within one-half the inter-sensor spacing is also implemented, and the displacement values are picked from a Gaussian distribution.
Finally, functions for restoring the arrays, inserting thinned elements and undoing the last command, is of course also implemented.

6.5 Summary and discussions

A necessary question arises: *Can these arrays be used in a practical system?*

As already stated the unequally arrays are (by the author) not considered to find application in ultrasonic imaging. Thus these arrays will be used with very large arrays (VLAs) and these arrays will when optimized with our routine exhibit excellent results. From the simulations with non-equally spaced arrays we may conclude that an array of this kind, should have an increasing inter-sensor separation towards its ends.
Being optimized with our method the angular responses from various non-equally spaced arrays are very similar. The main difference lies

in the dynamic range of the weight coefficients, Δ_D. Arrays that have such an increasing spacing have the lowest Δ_D among all the simulated unequally spaced arrays, when being compared at sufficient conditions (e.g. the same beamwidth, aperture and number of sensors). The arrays that have a decreasing separation between elements towards the ends always have the highest Δ_D. Thus the equi-spaced array's dynamic ranges have values that lie in between the two formerly mentioned non-equi spaced array types.
Therefore, when the construction of an array with increasing inter-element separation can be carried out in practice, it should be chosen before both equi-spaced and non-equally spaced arrays.

However, for ultrasonic applications we are left with the sparse arrays. First, it should be noted that it may be advantageous to thin an array in such a way that the elements are more densed near the array origin than on the ends. This follows from the conclusion made for the unequally spaced arrays. Moreover, this has also been observed by performing a massive number of simulations of responses, from various randomly thinned arrays. During these simulations we noted that arrays thinned with the anti-Gaussian routine generally had a better sidelobe level in the unweighted case. However, any function for the optimal location of elements has not yet been found[5].
This leave us with the results made from our random thinned arrays reported in section 6.3. Unfortunately, we could only weight down a limited number of the sidelobe peaks originating from applying a sparse array. And hence we could not optimize over the entire azimuth domain. Thus the practical use of such an array could be regarded as limited from this point of view.

The transducer array has been excited by a pulse signal (PW), three periods long.

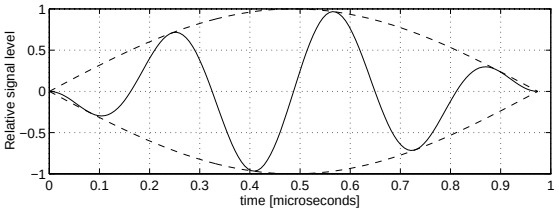


Figure 6.17: The excitation signal used with the arrays when calculating the energy beampattern. It displays a relative signal level along the time axis.

As seen in figure 6.17 the pulse has a cosine envelope. Thus it is cosine weighted and the amplitude of the pulse therefore decreases towards its ends. Because of this pulse weighting the excitation has a broader frequency spectrum than in the unweighted case.

[5] as far as the author knows nor have anyone else

We have simulated the energy beampattern from some of the sparse arrays with this excitation, and one of them is shown in figure 6.18.

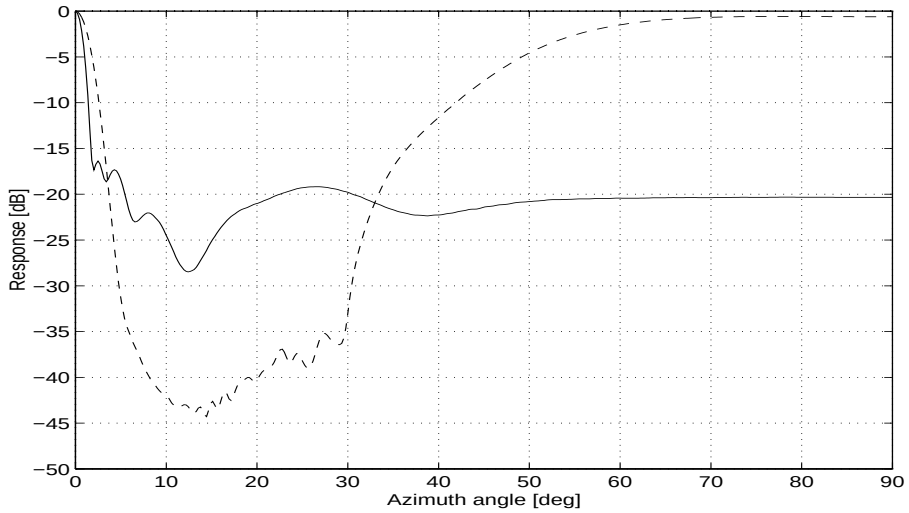


Figure 6.18: A 45 element sparse array with pulse excitation (PW). The pulse have three periods and are cosine weighted i.e. the emitted pulse has a cosine envelope and corresponds to pulses emitted from real transducers.

The cosine weighted response from the optimally weighted array is seen to reach a sidelobe level of about -42 to -43dB as a mimimum, while for the unweighted case the mimimum sidelobe level is about -28dB. Thus for a limited region the optimally weighted array is significantly better than the unweighted. Though, for azimuth angles beyond 33° the array can collect an amount of energy, from larger angles, that far exceeds the unweighted array. In a practical system the outer angles may be suppressed, but it is believed that will not be enough for the array considered here, due to the ascent of the responses starting at azimuth angles as low as 30°.

The next we wanted to check for the weighted array was its performance when being focused. We simulated the (CW) response from the transducer array with a focal point, $F \approx 24.4mm = 1.5A = 1.5 \cdot 32.5\lambda$ and the result is given in figure 6.19. It is observed that the response have some discrepancies with the unfocused. The first sidelobe is increased with $6 - 8$dB and the grating-lobe like structure is drawn a bit closer ($\approx 5^\circ$). Howvever, we observe that the response is by no means destroyed. The beamwidth seems to be kept and the mean sidelobe level within the optimization region has at least closely, been retained. Thus its seems as though the weighted array 'endures' the focusing and as known it is compelled to for having a practical application. In some special applications it might well do.
As a closing comment on this discussion is must strongly be emphasiz

that only a limited number of arrays have been used for these latter simulations.

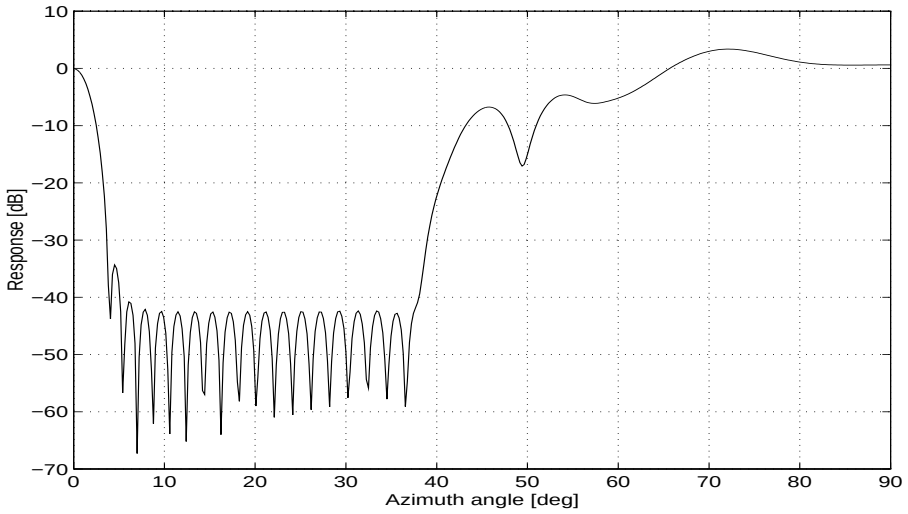


Figure 6.19: The same 45 element sparse array is now focused at $F \approx 1.5A = 24.4mm$.

Finally, we should summarize this the work reported in this chapter. We have used our general Remez-algorithm to optimize the apodization of a large number of arrays being both regular and irregular. Moreover, we have observed that our routine can improve the sidelobe performance of a variety of arrays. This also includes sparse arrays and to the authors knowledge, this has not been reported before.
The main limitation with our routine is the possibility of getting an ill-conditioned system matrix in Eq. 5.11 as well as the possibility of having a too small approximation error, as was observed in section 6.1.

Chapter 7

Conclusion

Different arrays have been proposed for various applications. In ultrasound special limitations on arrays exist due to the small wavelengths used, and to the strict spatial limits on the size of ultrasound probes for some applications. In this work we have mainly considered sparse and non-equally spaced arrays, despite the fact that the latter array generally cannot be used in an ultrasound imaging system. In this text some properties of these arrays and how to design them in an optimal way have been discussed.
An alternative approach to calculation of the beampattern of an array are given through a Fourier transformation of the coarray lag values. A program doing this has been written and demonstrated for linear arrays.
This chapter reveals the results and achievements from this research and we also propose some further work.

7.1 Results from the research

An optimization routine based on the Remez algorithm has been written. This routine optimizes uniform arrays as well as sparse and non-equally spaced arrays. The routine involves matrix inversion and is therefore dependent on well-chosen basis functions. It is the individual elements that determines the basis functions, and thinning of an array may therefore produce an ill-conditioned matrix. A careful choice of frequency grid spacing is also necessary in special cases, in order for converge to an equi-ripple sidelobe level. The routine is user interruptible so that a near optimal solution can be chosen if desired.
Its main advantage over other known routines are that it can be used with all types of element spacings. Additionally, it also is observed to give better results than a similar algorithm reported by Streit [57] have

demonstrated.

Our main result with unequally-spaced arrays is that we have determined what kind of inter-sensor spacing that should be preferred for large arrays, and that this can be seen from the dynamic range of the optimized weight coefficients. Thus not being proved analytically, our simulations have revealed a clear tendency towards the advantage of arrays having an increasing inter-sensor spacing towards the array ends. However, this array design will not give any saving in the number of sensors.

We have seen that our generalized Remez routine can optimize sparse arrays within a limited region. Judged from the 1994 Ultrasonic Symphosium[1], the optimization of weights approach, for sparse arrays, are quite new. And we have demonstrated that this approach might work in practice for some special applications.
However, a new optimization criterion must be used for optimization over a larger region of the azimuth domain and also for optimization of weights for two-dimensional arrays.

7.2 Further work

The propositions for further work could be several and only the ones considered as most interesting by the author will be mentioned. Further work on methods, developed routines and topics related to the work of this thesis could seemingly be further extended to include several new applications.

On the designed optimization method an extention that seemingly easy could be accomplished, would be to include optimization of *steered* sparse and non-equally spaced arrays. This would require changes in the error weighting function and the desired response on the dense grid. Additionally, the forming of the system matrix would have to be redefined to include a steering angle in the argument of the basis functions.
From our results published in [13] and from the discussion above, another method should be used when extending the optimization to two dimensions. This is because we only have control over a limited number of sidelobes. Thus the demand for a restricted number of sidelobe peaks is soon violated when thinning is applied to the arrays. Therefore optimization methods for two-dimensional arrays will probably be realized with non-linear optimization methods, linear approximation programming or a contrained optimization formulation e.g. the one in [30].

The coarray methods could also be extended. Maybe the most inter-

[1] arranged in Cannes, France, in November this year

esting extension would be to implement calculation of coarrays between distinct transmit and receive apertures. A recent article [34] uses an 'effective aperture concept' for optimizing two-dimensional transducer arrays, where the effective aperture is the convolution between the transmit and the receive aperture functions. Clearly this concept is equivalent to the sum coarray formulation and will as well be equivalent to the difference coarray formulation in case of arrays located symmetrically around their origin.
What is attempted is to place the off-axis energy of the transmit response at different locations from that of the receive energy in order to improve the dynamic range of the overall response. This could be utilized for example in 3D-visualization as reported in [51].

Appendix A

Equipment and programs

In this chapter we will present the hardware and software equipment used in this research.

A.1 Hardware equipment

The hardware equipment used in this research are UNIX workstations and for graphical output a Hewlett Packard *Laser Jet 4Si* is used.
The developed programs are written and executet on DEC (Digital Equipment Corporation) -system 500/240 stations at the Department of Computer Science, University of Oslo. These DEC-stations use ULTRIX operating system and ULTRIX Work-System Software (DEC's ULTRIX based window workstation system). The speed of these computers is 24 MIPS.

A.2 Software equipment

The programs used in this study are all written in **MATLAB** versions 4.1 and 4.2a for UNIX Workstations. MATLAB stands for (*matrix laboratory*) and is written in C by The Math Works Inc. It is an interactive program for numeric computation and data visualization. Fundamentally, MATLAB is built upon a foundation of sophisticated matrix software for analyzing linear systems of equations.
The developed programs are implemented in the **UltraSim** library [4, 59]. This is a simulator program developed by Vingmed Sound AS and the Department of Biomedical Engineering at the University of Trondheim.

Appendix B

A user's manual for the new tools

This chapter is an user's manual for the tools developed in the work with this thesis. The tools are being implemented in UltraSim under the *Calculation menu*. A manual is written for user's of Ultrasim that will utilize these tools in order to design arrays or simulate special effects. These routines are started as options under the **coarray** submenu.

The first option is **Array thinning**. This option contains functions for thinning and perturbing an array as well as placing elements with spacing determined by a geometric series. User operation in the thinning command menu is done by pushing the proper key while pointing on the UltraSim configuration window. When asked to input numbers, this must be done in the MATLAB command window.
A local symmetry flag is indicated by ON/OFF and is altered by the pushing the '**x**' button. When ON the thinning will remove elements symmetrically around the array origin.

The next option in the coarray submenu, **Optimize weights**, will give the optimal apodization for the current array. These apodization weights, are put into the global matrix variable **amp-ud** (*User defined apodization*). User operation is performed by pushing mouse buttons or a key on the keyboard, while pointing inside the UltraSim configuration window. The first function, uses the formulation of the Parks-McClellan Remez algorithm and thus can only optimize equi-spaced arrays. For equi-spaced arrays it will be the most efficient to use. Input value is the angle where the sidelobe level is desired to be reached (given in degrees).
A second routine uses our generalized Remez algorithm formulation and can thus bu used to optimize weight values for sparse, perturbed and non-equally spaced arrays as well as equi-spaced arrays. The current version is user interruptable in every iteration. By pushing the '**x**'-key

when asked to strik a key. This has been useful with arrays that slowly converges, because a more densed sampling[1] then might help.
The input format is :

$$[\phi_p^\circ \quad \phi_s^\circ \quad \phi_O^\circ \quad K]$$

Here ϕ_p° is the cut angle for the mainlobe in degrees, ϕ_s° is the angle where we want the equiripple sidelobe level to be reached, ϕ_O° is the upper limit of the optimization region and K is the approximation error weight value (which in most cases can be chosen to 10).

The third option is the **Coarray tools**. Here the user can calculate the difference and sum coarrays, plot the coarray in a figure (along with its error coarray, if it has been calculated) and some optional plotting functions like e.g. the beampattern from the coarray.
User operation is done by striking a key while pointing at the UltraSim **plot tool** window.

[1] This is determined by the *Grid spacing value* that has 16 as default value. We have also observed that in some cases a less densed sampling is most efficient

Bibliography

[1] B. A. J. Angelsen. *Waves, Signals and Signal Processing in Medical Ultrasonics*. Department of Biomedical Engineering, University of Trondheim, 1991.

[2] A. Antoniou. Accelerated procedure for the design of equiripple nonrecursive digital filters. *IEE Proc.*, 129(1-G):1–10, 1982.

[3] S. D. Bedrosian. Nonuniform linear arrays: Graph-theoretic approach to minimum redundancy. *Proceedings of the IEEE*, 74(7):1040–2, 1986.

[4] V. Berre. *ULTRASIM 4.0, program-documentation*. IBT, University of Trondheim, May 1993.

[5] K. A. Blanton and J. H. McClellan. A new search method for minimum redundancy arrays. In *Proc. IEEE ASSP*, 1991.

[6] H. P. Bucker. Adaptive cross-sensor beam forming with planar arrays. *Journal of Acoustical Society of America*, 62(5):1222–4, 1977.

[7] H. P. Bucker. High resolution cross-sensor beamforming for a billboard array. *Journal of Acoustical Society of America*, 65(1):145–7, 1979.

[8] E. W. Cheney. *Introduction to Approximation Theory*. McGraw-Hill Book Company, New York, 1966.

[9] W. Cheney and D. Kincaid. *Numerical Mathematics and Computing*. Brooks/Cole Publishing Co., Pacific Grove, 2nd edition, 1985.

[10] B. Delannoy, H. Lasota, C. Bruneel, R. Torguet, and E. Bridoux. The infinite planar baffles problem in acoustic radiation and its experimental verification. *Journal of Applied Physics*, 50(8):5189–95, 1979.

[11] C. L. Dolph. A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level. *Proc. of the IRE*, 34:335–48, 1946.

[12] C. H. Edwards and D. E. Penney. *Calculus and Analytic geometry*. Prentice-Hall International,Englewood Cliffs, 2nd edition, 1986.

[13] J. O. Erstad and S. Holm. An approach to the design of sparse array systems. In *Proceedings 1994 IEEE Ultrasonic Symposium*, Nov. 1994.

[14] J. W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill Book Company , San Francisco, 1968.

[15] C. R. Greene and R. C. Wood. Sparse array performance. *Journal of Acoustical Society of America*, 63(6):1866–72, 1978.

[16] F. J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[17] R. A. Haubrich. Array design. *Bulletin of Seismological Society of America*, 58(3):977–91, 1968.

[18] R. T. Hoctor and S. A. Kassam. The unifying role of the coarray in aperture synthesis for coherent and incoherent imaging. *Proceedings of the IEEE*, 78(4):735–52, 1990.

[19] S. Holm. Acoustic field simulation. Vingmed Sound, Sept 1993.

[20] S. Holm and K. Kristoffersen. Analysis of worst-case phase quantization sidelobes in focused beamforming. *IEEE Transactions on Ultrasonics,Ferroelectrics, and Frequency Control*, 39(5):593–99, 1992.

[21] T. S. Huang. Two-dimensional windows. *IEEE Transactions on Audio and Electroacoustics*, 20:88–9, 1972.

[22] J. A. Jensen and N. B. Svendsen. Calculation of pressure fields from arbitralily shaped, apodized, and excited ultrasound transducers. *IEEE Transactions on Ultrasonics,Ferroelectrics, and Frequency Control*, 39(2):262–7, 1992.

[23] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing: Concepts and techniques*. Prentice-Hall, Englewood Cliffs,NJ, 1993.

[24] Y. Kamp and J. P. Thiran. Chebyshev approximation fot two-dimensional nonrecursive digital filters. *IEEE Transactions on Circuits and Systems*, 22(3):208–18, 1975.

[25] M. Karaman, P. C. Li, and M. O'Donnell. Array processing for hand-held scanners. In *Proceedings 1994 IEEE Ultrasonic Symposium*, Nov. 1994.

[26] S. M. Kay. *Modern Spectral Estimation*. Prentice-Hall, Englewood Cliffs, 1988.

[27] G. S. Kino. *Acoustic waves. Devices, Imaging and Analog Signal Processing*. Prentice-Hall, Englewood Cliffs, 1987.

[28] L. E. Kinsler, A. B. Coppens, A. R. Frey, and J. V. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons , New York, 3rd edition, 1982.

[29] R. J. Kozick and S. A. Kassam. Coarray synthesis with circular and elliptical boundary arrays. *IEEE Transactions on Image Processing*, 1(3):391–405, 1992.

[30] R. M. Leahy and B. D. Jeffs. On the design of maximally sparse beamforming arrays. *IEEE Transactions on Antennas and Propagation*, 39(8):1178–87, 1991.

[31] J. Leech. On the representation of 1,2,...,n by differences. *Journal of London Mathematical Society*, 31:160–9, 1956.

[32] D. A. Linebarger. A fast method for computing the coarray of sparse linear arrays. *IEEE Transactions on Antennas and Propagation*, 40(9):1109–12, 1992.

[33] D. A. Linebarger, I. H. Sudborough, and I. G. Tollis. Difference bases and sparse sensor arrays. *IEEE Transactions on Information Theory*, 39(2):716–21, 1993.

[34] G. R. Lockwood and F. S. Foster. Optimizing sparse two-dimensional transducer arrays using an effective aperture approach. In *Proceedings 1994 IEEE Ultrasonic Symposium*, Nov. 1994.

[35] J. Y. Lu, H. Zou, and J. F. Greenleaf. Biomedical ultrasound beam forming. *Ultrasound in Medicine and Biology*, 20(5):403–28, 1994.

[36] T. Lyche. *Numerical Linear Algebra*. University of Oslo, Norway, 1992.

[37] J. H. McClellan, T. W. Parks, and L. R. Rabiner. FIR linear filter design program. *IEEE Programs for Digital Signal Processing*, pages 5.1.1–13, 1979.

[38] N. M. Mitrou. Results on nonrecursive digital filters with nonequidistant taps. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(6):1621–4, 1985.

[39] A. T. Moffet. Minimum-redundancy linear arrays. *IEEE Transactions on Antennas and Propagation*, 16(2):172–175, 1968.

[40] P. M. Morse and K. U. Ingard. *Theoretical acoustics*. McGraw-Hill Book Company , New York, 1968.

[41] R. A. Mucci. A comparison of efficient beamforming algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(3):548–58, 1984.

[42] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, 1989.

[43] D. P. Orofino and P. C. Pedersen. Efficient angular spectrum decomposition of acoustic sources - part I: Theory. *IEEE Transactions on Ultrasonics,Ferroelectrics, and Frequency Control*, 40(3):238–49, 1993.

[44] T. W. Parks and J. H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Transactions on Circuit Theory*, 19(2):189–94, 1972.

[45] A. Penttinen and M. Luukala. The impulse respons and pressure nearfield of a curved ultrasonic radiator. *Journal of Physics D.*, 9:1547–57, 1976.

[46] A. D. Pierce. *Acoustics: An introduction to its physical principles and applications*. McGraw-Hill Book Company , New York, 1981.

[47] W. K. Pratt. *Digital Image Processing*. John Wiley & Sons , New York, 2nd edition, 1991.

[48] L. R. Rabiner, J. H. McClellan, and T. W. Parks. FIR digital filter design techniques using weighted chebyshev approximation. *Proceedings of the IEEE*, 63(4):595–610, 1975.

[49] W. A. Sandrin, C. R. Glatt, and D. S. Hague. Design of arrays with unequal spacing and partially uniform amplitude taper. *IEEE Transactions on Antennas and Propagation*, 17:642–4, 1969.

[50] H. Schjær-Jacobsen and K. Madsen. Synthesis of nonuniformly spaced arrays using a general nonlinear minimax optimization method. *IEEE Transactions on Antennas and Propagation*, 24:501–6, 1976.

[51] S. W. Smith, H. G. Pavy, and O. T. von Ramm. High-speed ultrasound volumetric imaging system- part I: Transducer design and beam steering. *IEEE Transactions on Ultrasonics,Ferroelectrics, and Frequency Control*, 38(2):100–8, 1991.

[52] S. W. Smith, G. E. Trahey, and O. T. von Ramm. Two-dimensional arrays for medical ultrasound. *Ultrasonic Imaging*, 14:213–33, 1992.

[53] P. R. Stepanishen. The time-dependent force and radiation impedance on a piston in a rigid infinite planar baffle. *Journal of the Acoustical Society of America*, 49(3):841–9, 1971.

[54] P. R. Stepanishen. Transient radiation from pistons in an infinite planar baffle. *Journal of the Acoustical Society of America*, 49(5):1629–38, 1971.

[55] W. A. Strauss. *Partial Differential Equations: An introduction*. John Wiley & Sons, New York, 1992.

[56] R. Streit. Optimized symmetric line arrays. *IEEE Transactions on Antennas and Propagation*, 23:860–2, 1975.

[57] R. Streit. Sufficient conditions for the existence of optimum beam patterns for unequally spaced linear arrays with an example. *IEEE Transactions on Antennas and Propagation*, 23:112–5, 1975.

[58] A. H. Stroud. *Numerical Quadrature and solution of Ordinary Differential Equations*, chapter : Appendix C. Springer Verlag, New York, 1974.

[59] F. Teigen. *UltraSim Users Manual*. Department of Biomedical Engineering, University of Trondheim, Oct. 1994.

[60] D. H. Turnbull and F. S. Foster. Beam steering with pulsed two-dimensional transducer arrays. *IEEE Transactions on Ultrasonics,Ferroelectrics, and Frequency Control*, 38(4):320–33, 1991.