UiO : Department of Physics

# Near-field Characterization of Sonars

Asle Tangen

Master's Thesis, September 8th 2014

# Abstract

Sonars are used in a wide range of marine applications. The advantage of using sound waves in water has made the sonar technique ideal to detect underwater objects, imaging of the seafloor and detection of vessels. Sound waves can propagate over long distances in water unlike other waves. The ability to see underwater using acoustic waves has made the sonar technique important for marine and military industry. Most sonars uses the far-field when imaging. To characterize a sonars ability to depict one can measure the sonars far-field characteristics. This is usually done in big water tanks because the far-field boundary often is several meters away. This thesis investigates the possibility of characterizing the sonar using near-field measurements. This is done with simulations in matlab and measurements on a Simrad SH-90 transducer in a $[l, b, h] = 2x1{,}5x1{,}5\ m^3$ water tank at the University in Oslo. This was the first time measurements like these were done in the water tank so development of hardware and procedures were necessary. A method of simulating the wave field with parameters as close to the conditions in the water tank as possible was developed. The result of simulations showed that the system was sensitive to changes in the sound velocity. We found that it was possible to characterize a transducer if we had a calibrated system were the only unknown parameter is the speed of sound. By using the least square method of errors it was possible to characterize a transducer by finding the elements weights and phase. This gave us a basis for trying to match up the data in the measurement experiment. When trying to match up the simulated and measured data we found that there were too many unknowns in the measuring experiment. Due to time limitations of this thesis we were not able to do an optimal calibration of the measuring system. The work on this thesis showed us that it was possible to characterize a transducer in the simulations but more work has to be done on the measuring system.

# Acknowledgments

This thesis is written as a required part of the Master of Science degree in physics at the Department of physics, University of Oslo, Norway. The work was started in august 2013 and ended in September 2014.

First and foremost I would like to thank my thesis supervisor, Andreas Austeng at the Department of Informatics. You have been an excellent supervisor for my thesis. Your door has always been open for me when I needed help and advice. Your encouragement and interest throughout the period of this work was inspiring to me and a major incentive for finishing the thesis. To be honest; without your help, guidance and patience I would not have finished this work so you have my deepest gratitude.

I would also like to thank my co-supervisor Professor Sverre Holm for help on the hardware and making the measurement experiment possible. Thanks to Svein Bø at the Department of Informatics for guidance and help on the dynamic positioning system and the water tank. A special thanks to my fellow student Mohamed Kidash who worked with me in the water tank. At times we worked through some frustrating hours together trying to get the measuring rig to work. I wish to thank Kongsberg Maritime for lending us the SH-90 transducer. Thanks to Kjell Tangen at Veritas for some good discussions around the matlab programing and geometries in this thesis. I would like to give a special thank the DSB group at the Department of Informatics. To all the helpful, interesting and smart professors, PhD students and students who all contribute to an including and pleasant social environment. This makes the $4^{th}$ floor at the Department of Informatics great place to study.

Finally, I reserve my deepest appreciation for my partner in life, Johanna Westin. You have given me support, encouragement and patience throughout this laborious period. I love you with all of my heart.

# Table of Contents

# Chapter 1

## 1. Introduction

Sonar is a technique that uses sound propagation of acoustic energy from low to high frequencies to extract information from the surrounding environment. Sound waves can propagate over long distances in water unlike other waves, such as electromagnetic or optical that will quickly dissipate in the ocean. This makes the sonar technique ideal to detect underwater objects and has been important for the marine and the military industry for more than a century. The ability to see underwater using acoustic waves is advantageous to many. About 75% of the Earth's surface is covered by ocean and with most of the floor beneath the sea water uncharted; the mapping of the earth's seafloor is an important scientific goal. In addition, many commercial gains can be achieved, from planning underwater communication links or oil and gas pipelines to discovering underwater mining resources. Besides mapping the seafloor, the ability to see underwater allows us to explore. For example monitoring and locating underwater archaeological sites. In a military defense capacity, acoustic waves are useful for locating submarines, mines and guiding of underwater weaponry.

Throughout the $20^{th}$ and the $21^{st}$ century sonar has become an important technology for the fishing industry. Sonars ability to locate object over vast distances gives the fishing vessels ability to detect schools of fish several kilometers away from the vessel. This has rationalized the fishery and has resulted in the need for cheap reliable sonars that is easy to operate. With its long coastline, Norway has a rich fishing industry with different seasons for different fish species and geographical location. This makes Norway ideal for developing sonar technology for the fishing industry.

Simrad is a Norwegian firm and one of the world's leading manufacturers of sonars. The firm has developed sonars since 1953. Simrad develops sonars for both short and long range and use advanced broadband transducer design combined with digital output amplifiers and receivers. A frequency modulated signal with special filters ensures detection and identification of fish even under difficult conditions, and without losing the large distance range. The SH-90 transducer which is used in this thesis is a Simrad transducer. The production of these sonars involves testing of the sonar ability to depict. Since this testing must be done several meters away from the sonar, the industry has to use big water tanks. A problem with testing in big water tanks is that one sonar can be tested at a time. The question

is can this be optimized by doing the same measurements in a smaller environment in the near-field.

## 1.1 Objectives of this thesis

The sonars that are used for imaging usually operate with a center frequency in the range of 10 to 100 kHz. The images are formed in the sonars far-field and the smallest distance is limited and given by Eq. (1.1):

$$R > \frac{2D^2}{\lambda} \tag{1.1}$$

Where $R$ represents the range; $D$ represents the sonars diameter or extent; λ is the wavelength.

To characterize a sonars ability to depict one can measure the sonars far-field characteristics. This is done in a water tank. Because the far-field boundary often is several meters away, a big water tank is used to implement such a measure.

The DSB-group at IFI has a water tank that measure$[l, b, h] = 2x1{,}5x1{,}5\ m^3$. This is a small tank compared to what the industry uses to test sonars. In this Master thesis we investigate the possibility of estimating a sonars far-field characteristic by measuring its near-field characteristics. The water tank also has a dynamic positioning system that is used in this master thesis to carry out the necessary measurements for an acoustic near-field to far-field transformation.

This thesis is divided into the following objectives:

- Simulating the sonar field. This will be done in matlab and in the ultrasound simulation program Field II [13].
- Programing the dynamic position system so measurements can be made in the water tank.
- Preparing the SH-90 transducer. Make necessary hardware.
- Calibrating the system, this is done both with the measured and simulated data.

- Measuring the sonars near-field in the water tank.
- Characterizing the transducer based on these near field measurements.
- Comparing the measured near-field characteristics with the simulated near-field characteristics. This is done by looking at the response, amplitude, phase difference and element weights.
- Try the same with different delays in the elements. This is to see if the same result can be produced with simple signal processing.

## 1.2 Thesis outline

In this thesis the work on the objectives given in Chapter 1.1 is presented. It gives a presentation of the results of the measurement experiments and the simulations. It also contains a review of some of the basic sonar theory, wave theory, underwater acoustic theory and far-field/near-field theory. These underlying principles are important concepts to understand these theses. A presentation of the equipment used is also a part of this text. The water tank and the dynamic position system were used for the first time; therefore some information about this is also included. Including this one the thesis consists of 7 chapters, appendices and a bibliography.

***Chapter 2*** gives the presentation of the theory which is the basis for this thesis. The first part gives a short introduction to sonar history. The second part presents the basic physics and under water acoustic theory. The third part looks at the near/far-field theory. The last part of chapter 2 looks at the signal processing used in sonar.

***Chapter 3*** presents the methods and implementations used in this thesis. The chapter focuses on the simulation programs used for curved and linear arrays. It looks at the geometry's and calculations used to program the simulation programs. All of the simulations were done in matlab and some early tests were done in Field II. The chapter also presents the recording equipment used in the water tank experiment.

***Chapter 4*** looks at the measuring experiment. The first part of the chapter presents some of the equipment used, mainly the SH-90 transducer, the water tank and the dynamic position system. The recording equipment was presented in the previous chapter. The second part presents the measurements and how they were done. This part also looks at the first tests and the measurement of the speed of sound in the water tank. This is an important concept for the simulations and the processing of data.

***Chapter 5*** is the main chapter where the results of the measurement experiment and the simulations are presented. This chapter gives a thorough review of the results of the simulations and measurement data. The chapter gives a small review of the linear array results but the main focus is on the simulations with the curved array and the measurement data. In the first part we look at the optimization of the simulations. The second part looks at the results of the measurement experiment and in the last part of chapter 5 the results are discussed.

***Chapter 6*** summarizes and the conclusion regarding this work is made.

***Chapter 7*** presents suggestions for further work

# Chapter 2

## 2. Background

In this chapter an introduction to SONAR theory will be given. This theory is the basis for this master thesis. The main focus will be the basic underwater acoustic theory and the near-field/far-field theory. This chapter is based on lecture notes from imaging systems [9][16] and corresponding literature in statistical signal processing[11] and signal processing in space and time[14]

### 2.1 A short introduction to Sonar history

The use of sound in water for transmitting and receiving information has been of great interest for humans throughout centuries. Animals like dolphins and bats have used sound to communication and object detection for millions of years. The first recorded use of sound in water by humans was initially by Leonardo da Vinci in 1490. By inserting a tube into the water and placing an ear to the tube, one could detect vessels.

But the use of sound propagation in water wasn't utilized until the $19^{th}$ century when one of the earliest human applications of underwater sound was used to provide warning of hazards. By installing submerged bells on lightships, sound from these bells could be detected at considerable distances by hydrophones mounted in the hull of a ship. If two hydrophones were located on opposite sides of the hull the sounds received by each could be transmitted separately to the right and the left ear of the listener, with this separation the approximate bearing of the lightship could be determined.

The use of sound to locate objects underwater the same way bats use sound for aerial navigation, seems to have been prompted by the Titanic disaster in 1912. The first ever patent for an underwater echo ranging device was filed out at the British Patent Office by the English meteorologist Lewis Richardson one month after the Titanic disaster. The use of submarines during World War I prompted more research into the use of sound. A technique for detecting submarines was needed. The British made use of hydrophones, while the French physicist Paul Langevin and the Russian engineer Constantin Chilowski, worked on the development of an active sound device that could detect submarines by using the crystal quartz. Although piezoelectric and magnetostrictive transducers later superseded the electrostatic transducers that they used, this work influenced future designs.

During the 1930s the Americans developed their own underwater sound detection technology. This was when important discoveries were made and the Americans began to use the term SONAR for their systems. SONAR is coined as the equivalent of RADAR and is an acronym for sound navigation and ranging. The SONAR uses transmitted and received underwater sound waves to detect and locate objects by measuring distance under water. Today SONARs are used for navigation, communication, detection and imaging, both in military and civil applications.

## 2.2 Principals of sonar

The principal of sonar in the simplest terms is an electrical impulse from a transmitter converted into a sound wave by a transducer and sent into water. This can be a short burst of electrical energy converted into a very short burst of high frequency sound energy called a pulse. When this pulse strikes an object, it rebounds. This echo is detected by the transducer, which converts it back into an electric signal. This signal is amplified by a receiver and sent to a display.

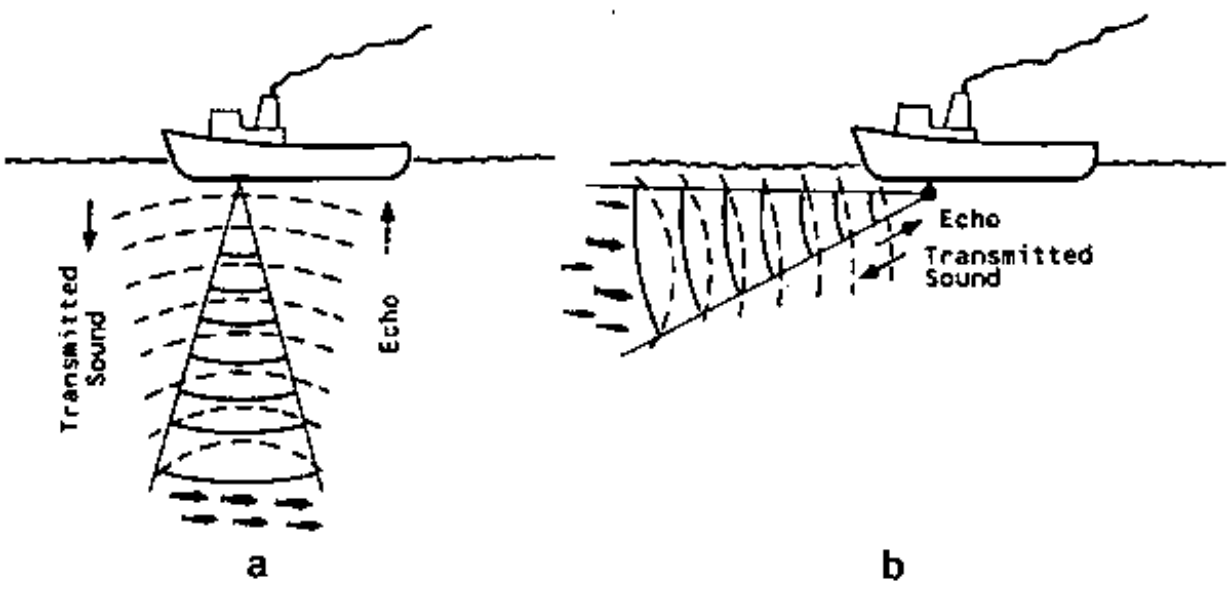


*Figure 2.1: Principal of sonar transmitted and received sound wave, a: vertical b: horizontal [FAO 2014]*

The sonar utilizes the piezoelectric effect[1]. The transducer is made up of flat cut quartz wafers layered with intermediate metal electrodes. If these wafers are exposed with electrical

[1] A piezoelectric substance (usually a crystal) is one that produces an electric charge when a mechanical stress is

impulses they send out sound. When sound waves hit the transducer the piezoelectric effect generates electrical impulses.

### 2.2.1 Basic physics

In this master thesis some basic physics about sound waves is used to produce and describe the results. Sound is a pressure perturbation that travels as a wave. We can also refer to sound as compressional waves, longitude waves and mechanical waves. To characterize these acoustic vibrations we use the following terms:

- Wave period $= \frac{1}{f(s)}$ , the duration of one cycle.
- Frequency $f = \frac{1}{T}$ (Hz), number of cycles per unit time.
- Sound speed c (m/s), the distance travelled per unit time by a sound wave propagating through a medium.
- Wavelength $\lambda = \frac{c}{f}$ (m), the spatial period of the wave, the distance over which the wave's shape repeats.
- Amplitude A, a periodic variable that is a measure of how much the wave change over a single period

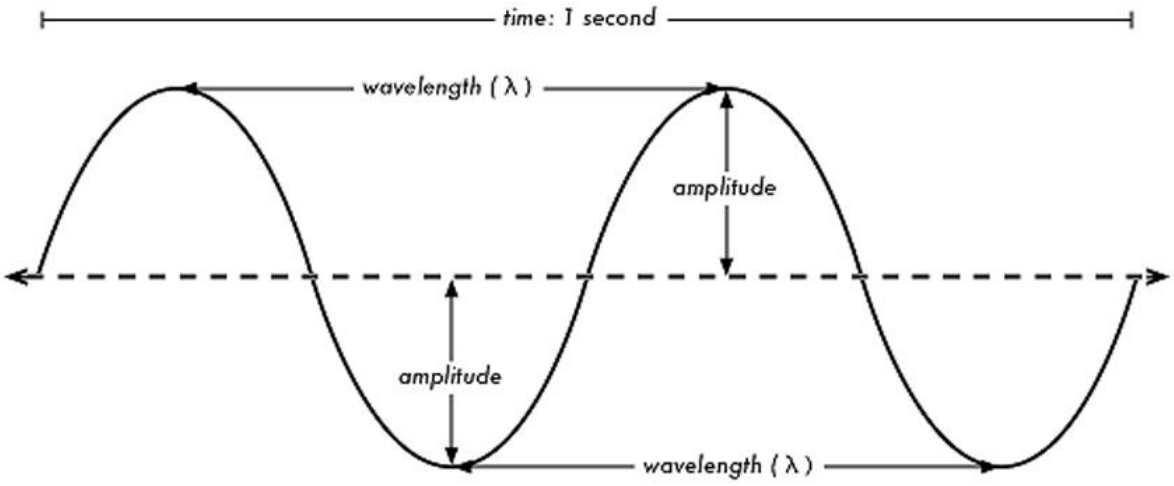


*Figure 2.2: Simple illustrations of basic wave physics showing the amplitude, wavelength and time [iGCSE Physics 2011]*

Sound pressure is often measured as a level on a logarithmic scale, referred to as decibel (dB). The dB scale is defined as:

$$I_{db} = 10log_{10}(I) \quad (2.1)$$

Where $I_{db}$ is the intensity in dB and I is linear intensity.

If we look at the dB scale we see that 10 dB is a factor 10, 20 dB is a factor 100, 30 dB is a factor 1000 etc. The dB scale increases exponentially and the reason for using this scale is that the acoustic signal strength varies several orders of magnitude over a typical distance travelled.

### 2.2.2 Basic underwater acoustic theory

Underwater acoustics describes the propagation of sound in water. The water may be salt water or fresh water. The frequencies typically associated with underwater acoustics are frequencies between 10 Hz and 1 MHz. The propagation of sound in water at frequencies lower than 10 Hz is usually not possible because the low frequencies penetrate deep into the seabed; frequencies above 1 MHz are rarely used because they are absorbed very quickly. Sonar operation is affected by a variation of factors which are important to discuss for the results in this thesis. The most important are Speed of sound, spherical spread, absorption, refraction, scattering and noise.

***Speed of sound*** is slower in fresh water than in sea water, though the difference is small. The speed is determined by the water's temperature, mass density, salinity and pressure (depth). The sound travels slower in cold water than warm water.

***Spherical spread*** is when an acoustic wave expands as a spherical wave in a homogeneous medium[2], as shown in Figure 2.3. The acoustic intensity $I$ decreases with range R as the surface of the sphere expands throughout the medium. The intensity decreases with the factor:

$$I = \frac{1}{R^2} \quad (2.2)$$

[2] A medium is homogenous if the physical properties are unchanged at different points.

We see in Eq. (2.2) that the intensity decrease proportional to the range squared. For the two way propagation the acoustic wave expands as a spherical wave to the reflector and the reflector spreads the signal in all directions as spherical waves. When the wave hits the receiver the two way loss become:

$$I = \frac{1}{R^2}\frac{1}{R^2} = \frac{1}{R^4} \tag{2.3}$$

As we can see in Eq. (2.3) the loss of intensity in the signal proportional to range is significant in two way spherical spread.

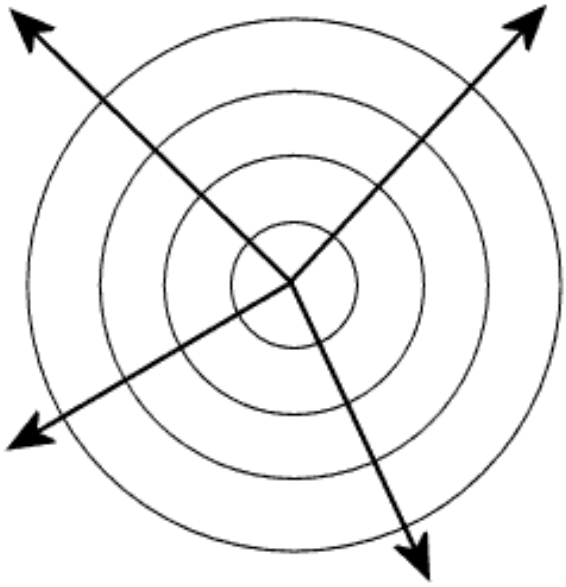

*Figure 2.3: spherical wave spread from a single point in a homogenous medium [Wikipedia 2014]*

***Absorption*** is frequency dependent, the higher the frequency, the greater the absorption. This means that lower frequencies has longer range than higher frequencies. Temperature does also influence on absorption, the higher the temperature, the greater the absorption. This means that the waves goes faster in warm water but has shorter range. The absorption also depends on the salinity of the water. For normal sea water (with salinity of 35ppt) the absorption at 400 kHz is approximately 100dB/km and 50dB/km at 200 kHz. Fresh water has little, if any salinity (0,5ppt), so absorption is considerably less. Approximately 40dB/km at 400 kHz and 10 dB/km at 200 kHz.

***Refraction*** happens when the sound velocity continuously changes with the change in pressure. An acoustic ray will continuously refract into a new direction at different depths. This is because the ray passes through a sound speed gradient from a region of one sound speed to a region of a different sound speed. How much the ray bend is dependent upon the amount of difference between sound speeds, the variation in temperature and the salinity and depth of the water.

***Scattering*** of acoustic waves in water can be divided into the two categories surface scattering and volume scattering. Surface scattering is scattering from the surface or the seafloor and volume scattering is scattering from fluctuations, marine life or objects. Surface scattering from a smooth surface will mainly give specular reflection[3]. With rough surfaces will parts of the radiated acoustic energy be scattered diffusely in random directions. If the surfaces are very rough more acoustic energy will be scattered diffusely. This means that for non-normal[4] incident waves the surface has to be rough for the signals to reach back to the receiver. The scattered field is dependent on the roughness of the surface and the difference between media.

***Noise*** is error or unwanted disturbance in a signal. There are a wide variety of noise sources present in the underwater environment. In sonar there are three important noise sources that effect sonar performance. Background noise also called ambient noise, reverberation and self-noise.

Background noise is sounds that interfere with the sound the sonar is monitoring. Background noise can be thermal, biological, from surface (wind, rain etc.) or human made (Distant ship traffic etc.). Background noise is location and frequency dependent. At the lowest frequencies, from about 0.1 Hz to 10 Hz, ocean turbulence is the primary contributors to the noise background. Distant ship traffic is one of the dominant noise sources in most areas for frequencies of around 100 Hz, while wind-induced surface noise is the main source between 1 kHz and 30 kHz. At very high frequencies, above 100 kHz, thermal noise of water molecules begins to dominate.

[3] Specular reflection is when the reflection angle corresponds to the angle of incidence.
[4] Non-normal incident waves have an incident angle that is not 90 degrees.

Reverberation is mainly caused by multiple reflections of the transmitted signal. It is the echo of the signal off of the environment. Surface and bottom reverberation and scatters in the water also called volume reverberation.

Self-noise is the noise that the sonar and the platform it is mounted on make. The sources for self-noise can be electrical, machinery or flow of water across the transducer. Flow noise is usually the dominant source of self-noise for a moving sonar system.

### 2.2.3 Range and bearing estimation

***Range*** is defined as the distance between the sonar and the reflector. Range estimations are a very important application of sonar. To measure the distance to an object, the time from transmission to reception is measured and converted into a range by the speed of sound. This shows the importance of knowing the speed of sound.

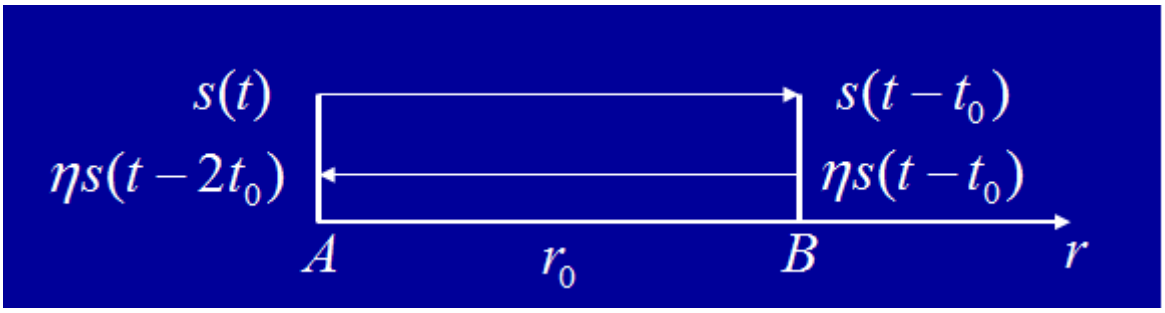


*Figure 2.4: Range estimation in a sonar system, A is transmitter and receiver. B is reflector [Tryggve Sparr 2014]*

In Figure 2.4 we see a pulse $s(t)$ of duration $T_p$ transmitted from a transmitter A in the direction of a reflector B. The receiver records the signal until the echo from the reflector arrives. The pulse arrives at the reflector with a time delay $t_0$ and is reflected back to the receiver with a reflection coefficient η. The pulse is received at the receiver with a time delay $2t_0$. This gives us a received signal$w(t)$:

$$w(t) = \eta s(t - 2t_0) \tag{2.4}$$

Where $2t_0$ is:

$$2t_0 = \frac{2r_0}{c} \tag{2.5}$$

If we say that $2t_0$ is the delay $\tau$ we can find the range:

$$r_0 = \frac{c\tau}{2} \tag{2.6}$$

The sound velocity $c$ has to be known to be able to map the delay into distance. The accuracy of which the range is estimated is related to the length of the pulse $T_p$. A short pulse gives better range resolution. However, short pulses can have less energy, which again gives shorter propagation range.

***Bearing*** is the estimation of direction. There are some key elements in estimation of direction in sonar: Transducer size and the grouping of transducers into arrays. A transducer is directive if the size of the transducer is large compared to the wavelength. The directivity pattern generally contains a main lobe with a beam width. To find the beam width of a transducer we can use the equation:

$$\beta \approx \frac{\lambda}{D} \tag{2.5}$$

where β is the beamwidth, $D$ is the diameter of the transducer and λ is the wavelength .Since $\lambda = \frac{c}{f}$ we can see in Eq. (2.5) that the beam width is frequency dependent:

$$\beta \approx \frac{c}{fD} \tag{2.6}$$

We see in Eq. (2.6) that higher frequencies gives narrower beam for a given transducer size or smaller transducer size for a given angular spread. This is the single most important reason for choosing high frequencies in sonar imaging.

### 2.2.4 Imaging sonar

Imaging sonars transmit sound pulses and convert the returning echoes into digital images, much like a medical ultrasound. The process of imaging in sonar can be described by:

- The aperture is excited by an electrical signal, so that it emits a wave field into the water.
- The wave field propagates through the water, experiencing attenuation, diffraction, scattering and reflection.
- The reflected and scattered wave field propagates back towards the transducer, where upon reception the measured wave field is converted into electrical signals.
- These electrical signals are processed so that they can be used to form an image.

The advantage with sonar is that they can "see" through dark or turbid water in zero visibility conditions. Sound wavelengths in water are about 2,000 times longer than the wavelengths of visible light. Because the wavelengths are so much longer, sound can go around particles that would otherwise block and scatter light waves. This makes optical systems like underwater cameras ineffective and unable to penetrate over long distances. Another disadvantage with optical imaging is that it lacks the range information found in sonar images.

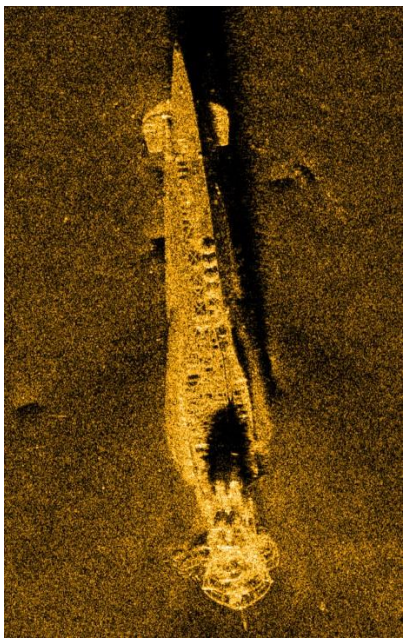

*Figure 2.4: Synthetic aperture sonar image of a sunken WWII submarine done with the HUGIN AUV. [Roy Edgar Hansen 2007]*

The principle of imaging sonar is to estimate the reflectivity for all calculated ranges and in all selected directions. The field of view is given by the angular width of each element in an array. The azimuth resolution is given by the array length measured in wavelengths. The range resolution is given by the bandwidth of the system. To characterize a sonars ability to depict one can measure the sonars far-field characteristics.

## 2.3 Near-field/Far-field

Transducers have both near-field and fare-field boundaries. In sonar theory the near-field/far-field boundary is an important concept. The reason for this is that a targets scattering characteristics is different in the near-field than in the far-field. In long ranges a underwater target is seen as point expressed by a single highlight. But from short ranges an extended underwater target can have several distribution characteristics.

### 2.3.1 Near-field

The near-field region can be divided into 2 regions. Reactive near-field and radiating near-field also called the Fresnel region. The reactive near-field region is very close to the transducer were the fields are out of phase by 90 degrees of each other. This region can be described by:

$$R < 0.62\sqrt{\frac{D^3}{\lambda}} \tag{2.7}$$

The radiating near-field is the region between the reactive near-field and far-field. In this region the reactive fields are not dominant and the radiating fields begin to emerge. In this region the shape of the beam pattern may vary greatly with distance. The radiating near-field region can be described by:

$$0.62\sqrt{\frac{D^3}{\lambda}} < R < \frac{2D^2}{\lambda} \tag{2.8}$$

Within the near-field the wave fronts produced by the transducer is not parallel. The intensity of the wave oscillates with range. This means that the echo levels from targets within the near-field region can vary greatly with small changes in location. If the source is located close to a transducer, the wave front of the propagating wave is curved with respect to the dimensions of the transducer and the wave propagating direction depends on sensor location.

### 2.3.2 Far-field

The far-field, also called the Fraunhofer region is the region far from the transducer. Within the far-field region the direction of propagation is approximately equal at each sensor and the propagating fields within the transducers aperture consists of plane waves. When the sonar transmits, the sound is projected from the whole surface of the transducer. This means that the near/far-field transition occurs when the sonar beam is equal to the size of the transducer. This is illustrated in Figure 2.5.

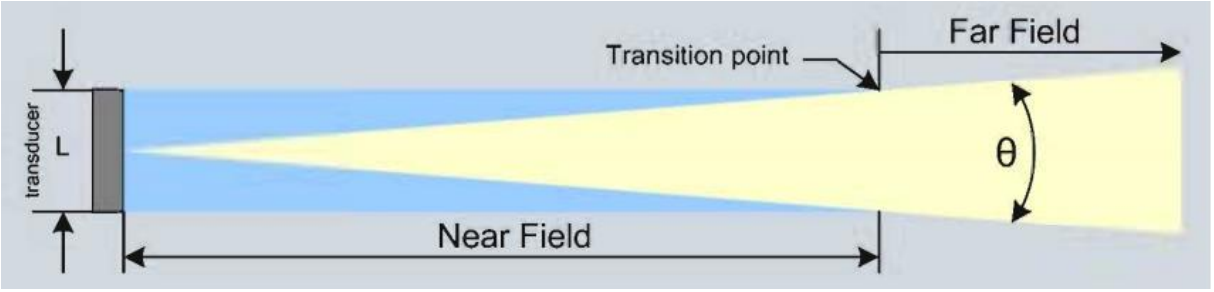


*Figure 2.5: the near-field/far-field transition zone, the transition point is when the beam equal to the size of the transducer L.[Kongsberg-mesotech 2014]*

To find this transition zone we can use the Eq. (1.1) with the conditions that R>>D and R>> $\lambda$. The far-field region is the most important because this is where we determine the sonars beam pattern.

## 2.4 Sonar signal processing

When sonar receives an acoustic signal from a target the information in the signal cannot be utilized and used without signal processing. Signal processing is altering the properties of a signal to achieve some effect. Useful information has to be extracted from the signal. This is done in several steps. In this section we look at array theory, beamforming, beam pattern, real and complex signals, sampling and time frequency signal processing witch is important concepts for this thesis.

### 2.4.1 Arrays

Arrays of elements are used in a wide range of applications where the goal is to gather information from the surrounding space. Arrays consists of a number of sensors that each could be an aperture or Omni-directional transducer. The sensors are combined in a discrete space domain to produce a single output. Arrays have the advantage that a two-dimensional or three-dimensional volume can be scanned by adding phase or time delay to the array elements without the system having to contain moving mechanical parts. Another advantage of sensor arrays is that an array can increase the antenna gain in the direction of the signal and decrease the gain in directions of unwanted noise. This means the sensor array can increase the signal to noise ratio. Sensor arrays can come in many different shapes both regular and irregular.

A commonly used array is the one containing M spaced elements on a flat line referred to as a uniform linear array. Uniform linear arrays have the element placement:

$$x_m = \left(m - \frac{M+1}{2}\right) d, \qquad m = 1, \dots, M, \tag{2.9}$$

where $d$ is the distance between each element. The time delay required for the $m$th element to steer the wave field in the direction $\theta_0$ in the far-field can be described as:

$$\tau_m = \frac{md}{c} \sin\theta_0 , \qquad m = 1, \dots, M, \tag{2.10}$$

where $c$ is the speed of sound.

### 2.4.2 Focusing

All sonar systems suffer in noisy environments which can cause significant degradation to the sonar systems performance. To improve the sonars it is desirable to increase the received signal to noise ratio. Signal to noise ratio is the level of desired signal to the level of background noise. One way of increasing the SNR is through focusing of the wave field. This means focusing the wave field to one focal point. With arrays this can be done by imposing a time delay on each element making the different wave fields arriving at the focus point at the same time. This is referred to as beam forming.

### 2.4.3 Beam forming

Beam forming is the technique of using an array of transducer elements to focus or steer a wave field. Beam forming can be implemented both on transmission and reception. At transmission both the amplitude and time of excitation is controlled at each element so that propagating waves add up constructively in the focal point.

### 2.4.4 Beam pattern

The way to describe the amplitude profile of an array is to plot a beam pattern also called a radiation pattern. The beam pattern is the function of the angle of transmission. A plot of the beam pattern shows the magnitude of the main lobe and the associated side lobes. The side lobes are unwanted radiation in undesired directions. The power density in the side lobes is generally much less than that in the main lobe. It is generally desirable to minimize the side lobe level. Ideally the peak of the first side lobe should be between 10-15 dB below the maximum main lobe power but this depends on the shape of the aperture, rectangular or circular.

The beam center is the average values of the angle of transmission. Ideally it is centered on 0 degrees. The width of the main lobe is specified by the half power beam width, the angle between the points on the side of the lobe where the power has fallen to -3 dB of its maximum value. A beam pattern is illustrated in Figure 2.5.

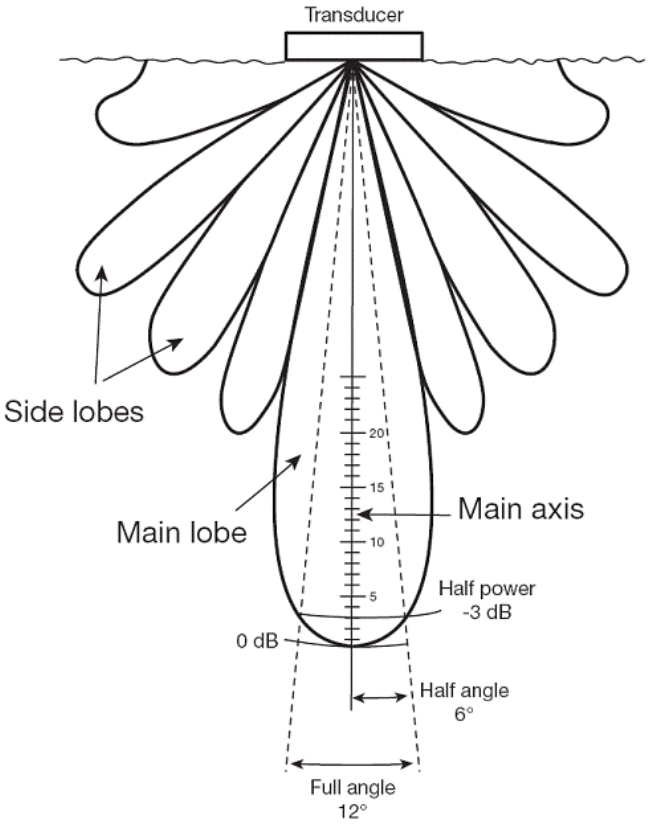


*Figure 2.5: Beam pattern of a transducer, shows the main lobe with side lobes. Half power limit at -3dB [Cornell University 2014]*

The beam pattern is an important concept in this master's thesis because the beam pattern of sonar is defined in the far-field. Like stated in Chapter 1.1 the main objective of this thesis is investigating the possibility of transforming the near-field measurements to the far-field and creating a beam pattern for the SH-90 transducer.

### 2.4.5 Real and complex signals

Acoustic signals are disturbances in the background pressure level in the medium. We can say that it is a disturbance in the normal background environment that conveys information. The information is contained in the way the disturbance changes in time, frequency, direction and space. All signals in the real world are real. However in signal processing applications it is convenient for the purpose of analyze and data reductions by means of band pass sampling to represent a signal as a complex-valued function of space and time. Sonar normally operates with a small bandwidth. For purely mathematical reasons, the concept of complex number representation is closely connected with many of the basics of signal theory, such as frequency response, transfer-function, Fourier etc.

### 2.4.6 Fourier analyze

All signals real or complex that vary in time, can be broken into its spectrum. We can compare this to how light get broken into different colors with a prism. The mathematical expression for this is called Fourier transform. The Fourier transform of a time signal gives the frequency content of the signal. Much signal processing is done on the frequency domain by mathematical operations on the Fourier transform of the signal in interest. These results of the Fourier transform can be converted back to a time signal with the inverse Fourier transform.

### 2.4.7 Sampling

In the field of signal processing sampling is a fundamental link between the analog domain and the digital domain. Sampling is the process of converting a signal of continuous time or space to a numeric sequence of discrete time or space. The sampling rate $f_s$ is defined as the number of samples obtained in one second to represent the signal digitally. Suppose we want to recreate a continuous sin wave, how often do we need to sample it to recreate the wave?

If we sample at one time per cycle we can think it is a constant. If we sample it 1.5 times per cycle we can think it is a lower frequency sine wave. If we sample at twice the sample frequency we see that the sin wave can be closely recreated. This makes an important theorem in the spectral analysis, the sampling theorem. It states that a real, band limited, finite signal can be reconstructed from samples taken at 2 times the Band width. This is called the Nyquist rate. Sampling at less than the Nyquist rate can result in overlap in the frequency domain also called aliasing. The result is distortion of the reconstructed signal.

# Chapter 3

## 3. Methods and implementation

This chapter presents the methods and the implementation used in this master thesis. We look at the theoretical calculation of the field, the simulations in matlab, the experiment and the implementation of the recording equipment used in the water tank. All the results of the simulations and the water tank experiment are presented in more detail in Chapter 5.

### 3.1 Theoretical calculations and principals

This section presents the theoretical calculations and principals that form the basis for the simulation programs and measurement experiment.

#### 3.1.1 Huygens principle

Huygens principle states that every point on a wave front can be regarded as a new source of waves. The wave front is defined as the point which wave motion is in the same phase. The idea is that any point in the medium where the wave motion takes place, can be regarded as the starting point for waves, elementary waves, which propagate uniformly in all directions. The real wave motion is equal to the sum of all elementary waves at a given moment that spreads from a wave surface. By calculating the elementary waves and summing them we can find the shape of the wave surface at a specific time.

#### 3.1.2 Least square method

The least squares method is a mathematical model that describes a set of data in a way that minimizes the difference between the model and the data. In signal processing the least squares method can be used to find the parameters that produce the least difference between the desired and the actual signal. The least squares method looks at the squared sum of the errors between the desired and the actual signal. The squared sum of errors can be described by:

$$S = \sum_{i=1}^{n} |e(n)|^2 \tag{3.1}$$

Where $S$ is the sum of errors, $n$ is the number of data points and $e(n)$ is the error between the desired and the measured data. For an optimal fit between the desired and the actual signal the sum of errors must be as close to zero as possible. This can be used in the simulation program to find the optimal parameters.

## 3.2 Simulations

This part will present the simulation methods used in this master thesis. The simulations have been done in three different matlab scripts. Linear array, curved array and back propagation.

### 3.2.1 Purpose

The purpose with the simulations was to recreate the environment in the water tank and simulate the sonar fields. By simulating the sonar fields it was easier to find the distance and the number of measuring points needed for the water tank experiment. The results of the simulations can then be compared with the sonar fields measured in the water tank. The idea was to see if we could characterize an array element with regard to weight and phase based on near field measurement. If so it will be possible to calculate the far-field response. The simulations were first done with a linear array in matlab and then with a curved array. The reason for doing it with a linear array was to look at the field response with a simple flat array. The simulations with the curved array were done in the ultrasound simulation program field II and a simulation program KjorKode.m m.

### 3.2.2 Field II

Field II is a program for simulating the ultrasound transducer fields and imaging using linear acoustics. It is developed by Jørgen Arendt Jensen [13] at the Department of Electrical Engineering at the Technical University of Denmark. The program is running under Matlab and uses the Tupholme-Stepanishen method [13] for calculating pulsed ultrasound fields. Field II is developed for ultrasound transducers but is capable of calculating the emitted and pulse-echo fields for both the pulsed and the continuous wave case for a large number of different transducers. In this thesis a curved linear array is used as the transducer. The program can simulate any kind of linear imaging as well as realistic images of human tissue. Field II is currently free to use under certain restrictions [13].

### 3.2.3 Linear arrays

The first simulations we did were with a linear array. This was to see if it was possible to produce a result and simulate a sonar field with a simple flat array. Two matlab scripts was made, one for measuring the field and one to back propagate the measured field data. The

experience we got from the linear arrays would help us program simulation programs for a cylindrical array in Field II and matlab.

An important aspect with simulating any array shape is to look at the geometry in the simulation sequence. It is important that the pulse is sent at the right time and is measured at the right time and place. The shape of the array and the behavior of radiating waves from each element are important to describe the geometry of the measuring sequence. In this section we look at a flat linear array where the geometry can be described as:

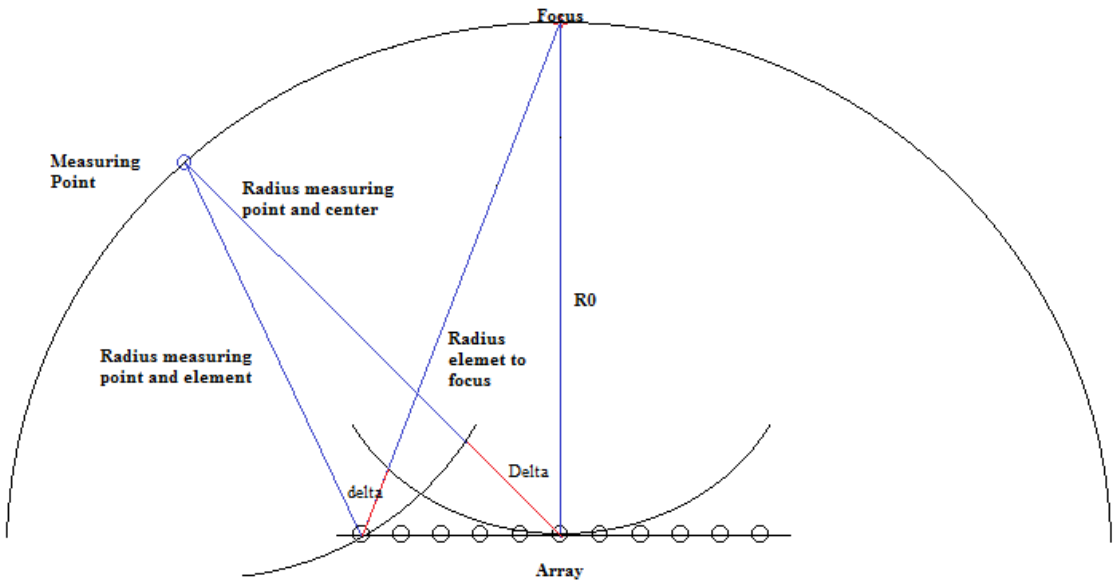


*Figure 3.1: Geometry for an 11 element linear array. The distance between measuring points and array is set to extreme near-field to illustrate the different distances.*

Figure 3.1 illustrates the geometry of the simulation sequence. In this example the array consists of 11 elements but the same geometry applies for arrays with more elements. The measuring points are set with a distance R0 from the array. The focus point is a distance of R0 from the array center. The focus point is where all the elements are focused. This means the field has to be recorded when the radiating waves from all the elements reaches the focus point at the same time. In Figure 3.1 this is illustrated with the element one on the left side. If we want to see what the pulse is in the measuring point in Figure 3.1 we have to look at the distance from the element to the measuring point, element to focus and measuring point to

center. By focusing from the center of the array the difference in the pulse traveling distance from each element to focus point is delta. This means that to get a focused array the elements have to start sending a time equivalent of delta before the center element for all the pulses to reach the focus point at the same time. In the measuring point the pulse for element 1 has then traveled a distance of Delta+delta past the measuring point this is illustrated in figure 3.2.

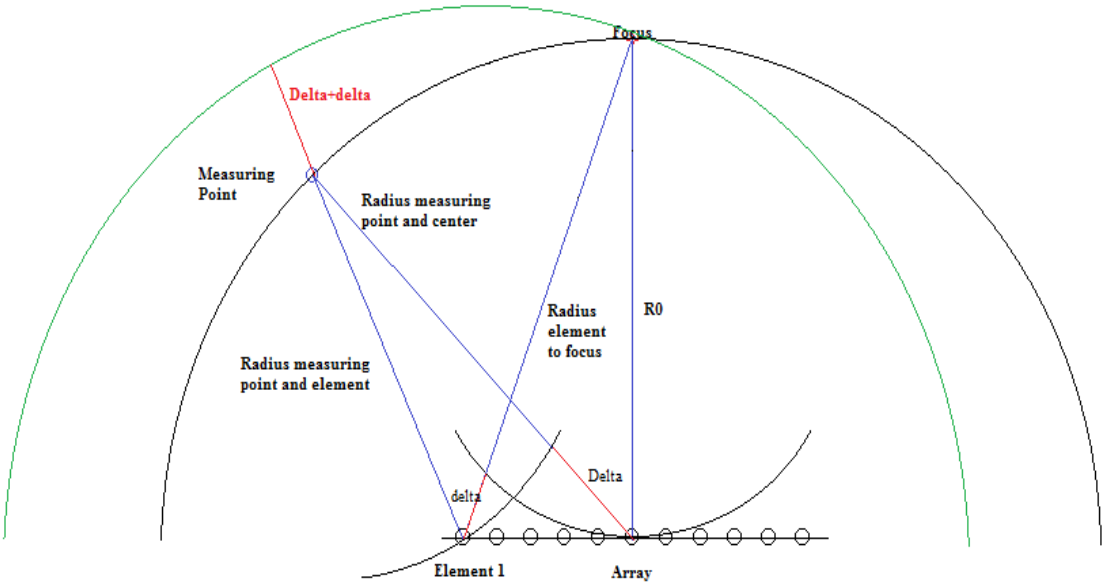


*Figure 3.2: The pulse traveled from element 1. The green half circle illustrates the radiated field from element 1 at a distance Delta+delta from the measuring point.*

Since element 1 is closer to the measuring point than the focus point we get an additional Delta from the measuring point to the center of the array. The pulse has traveled a distance of Delta +delta past the measuring point when all the pulses have reached the focus point and Delta+delta shorter at the measuring points furthest from the element. This means we have to make a measuring window greater than the pulse length on each side of the measuring points to see the whole field. With this geometry we could start looking at the simulation program.

To see what the field looks like at a given distance from the array we have to look at the response in every measuring point at a time window. The field can be described as:

$$R_n(t) = \sum_m P(t) \tag{3.2}$$

Where $n$ is the measuring position and $m$ is the element position. $R_n(t)$ is the response in the measuring points and $P(t)$ is the pulse. This is in the time domain. In our case we have to put in a time delay equivalent to our deltas. We call this $\tau_{mn}$. This gives us the expression:

$$R_n(t) = \sum_m P(t - \tau_{mn}) \tag{3.3}$$

The time delay has to be equivalent to the deltas in our geometry in Figure 3.1 and 3.2. This means that

$$\tau_{mn} = \tau_{mn}^{T.d} + \tau_m^{focus} \tag{3.4}$$

Where $\tau_{mn}^{T.d}$ is the traveling distance between measuring position and element position and $\tau_m^{focus}$ is the distance between element position and focus. This gives us:

$$\tau_{mn}^{T.d} = \frac{\|element\ Position - measuring\ position\|}{c} \tag{3.5}$$

$$\tau_m^{focus} = \frac{\|element\ Position - focus\|}{c} \tag{3.6}$$

By dividing on the speed of sound c we get the distance over in the time domain. By creating a big enough time windows for the measuring points these formulas will give us the field at a given distance. In the matlab program ForoverPropagering.m a simple linear array transmits a pulse, and a given number of points at a given radius measure the sonar field.

In the matlab program BakoverPropagering.m the measurements are made at the array and the response measured in ForoverPropagering.m is transmitted from the measuring field. By doing it this way the field is back propagated to see what the responses are over all elements. In this program the geometry is a little different since the measuring points are now the elements and vice versa. This gives us

$$R_m(t) = \sum_n R_n(t + \tau_{mn}) \tag{3.7}$$

In this case $R_m(t)$ is the response in the elements and $R_n(t - \tau_{mn})$ is the response in the measuring points with the time delay $\tau_{mn}$ witch in this case is

$$\tau_{mn} = \tau_{mn}^{T.d} = \frac{\|element\ Position - measuring\ position\|}{c} \tag{3.8}$$

This gives us a delay on the traveling distance.

### 3.2.4 Simulating with linear arrays

The simulation is done in the far-field. The distance between the measuring points and the array is set to 1000 meters. This gives us a sonar field close to infinity. The pulse is set to 3 cycles the same as in the water tank experiment. The number of measuring points is set to 121. This is the minimum number of measuring points that can be used and still get a good resolution.

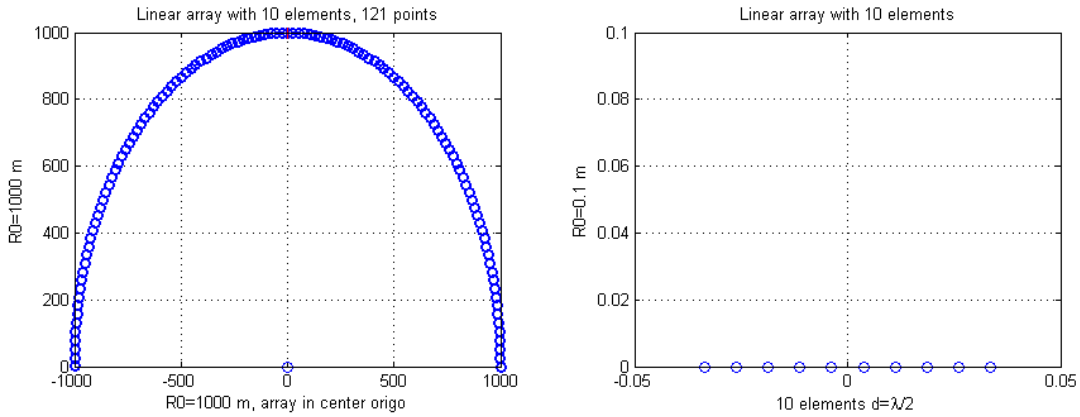


*Figure 3.3: The measuring points and the linear array. The number of measuring points is 121 and the number of elements in the array is 10.*

### 3.2.5 Curved linear array

This simulation was done with two different simulation programs and a matlab program created by the author. The first simulations with the curved array were done in Field II. The reason for using Field II was to create a sonar field and see how it looked and then create a program that did the same thing and could be use to back propagate the measured data from the water tank.

***Field II:*** To create a transducer in Field II with the same dimensions as the SH-90 transducer used in the water tank, a curved linear array is used in the simulations. This array is created in the matlabfile xdc_curved_array.m. This matlab file is created by Jan Egil Kirkebø[15]. By using this matlab file we can create a cylindrical array with the same specifications as the SH-90 transducer. Like the SH-90 transducer we create an array with 48 elements where each element is divided into 10 sub elements. This gives us a full cylinder with the same dimensions as the SH 90 transducer (figure 3.3).

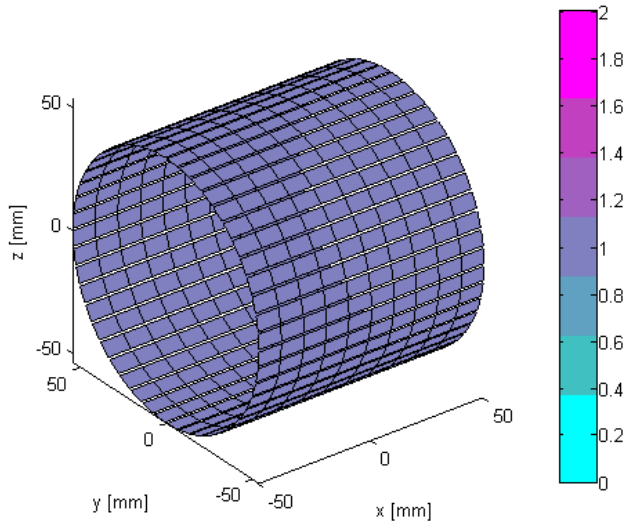


*Figure 3.3: Curved linear array, full cylinder with 48 elements divided into 10 sub elements*

In the water tank experiment we only use one element with 10 sub elements. The measurements in the water tank are made from +90 degrees to -90 degrees in front of the transducer in a 180 degrees part of the full cylinder. This means that only 24 elements are needed to create the same sonar field as in the water tank (Figure 3.2).

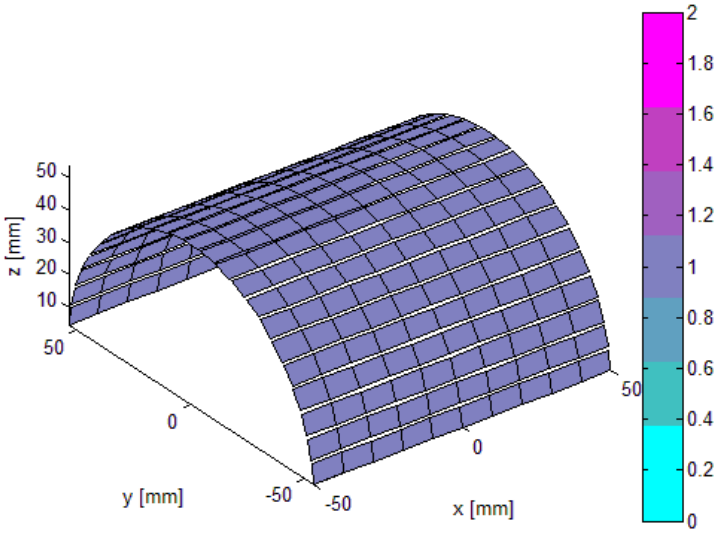


*Figure 3.2: Curved linear array with 24 active elements*

***The matlab program:*** This program was created in the matlab script KjorKode.m. The only difference in the geometry is the curved array, were the elements are placed with a radius $r_0$ from center.

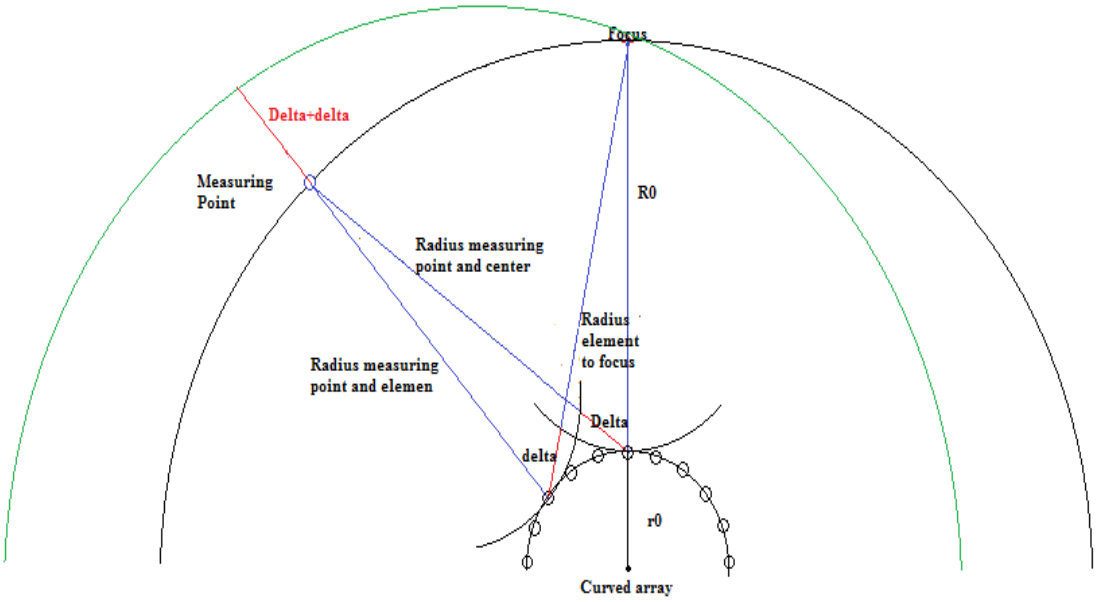


*Figure 3.6: Geometry of a 11 element curved array with a distance $r_0$ from center*

We see in Figure 3.6 that the geometry is similar to the linear array except from the radius $r_0$ in the array. This gives us a new delay time in $\tau_m^{focus}$:

$$\tau_m^{focus} = \frac{\|element\ Position - focus\| - r_0}{c} \tag{3.9}$$

Where $r_0$is:

$$r_0 = \min(\|element\ position - focus\|) \tag{3.10}$$

By putting this into the program KjorKode.m we get a simulation program that works for the curved array. It is not necessary to look at the field from +90 to -90 degrees in the simulations. To look at a sector of 120 degrees in front of the transducer should be sufficient to characterize a transducer.

### 3.2.6 The simulations in Field II

In the matlab file LagSimulering.m, a measuring sequence is set up. In this program the center frequency is set to 100 kHz and the sampling frequency is set to 100 MHz. In LagSimulering.m it is possible to set the parameters of the transducer and the number of measurement points in the simulation. One can also control the distance between the transducer and the measuring points. This provides the opportunity to recreate the conditions in the tank and get measurements as close to the SONAR field of the SH-90 transducer as possible. The pulse is set to 3 cycles like in the water tank experiment.

## 3.3 The experiment and recording equipment

In this part we will present the experiment and recording equipment used. This part only looks at the recording equipment and the SH-90 transducer, the water tank and the dynamic position system will be presented in more detail in chapter 4.

### 3.3.1 Purpose

The purpose with the experiment and recording equipment was to record the measurements done with the SH-90 transducer in the near-field in order to fully characterize the transducer and then calculate the far-field response. The measurements can be done at different distances from the transducer by using the dynamic positioning system. It was also a objective to test and learn how use the existing equipment at the sonar lab at Ifi since it had never been used in this purpose.

### 3.3.2 Experiments

The experiments were done in the water tank at Ifi with a SH-90 transducer, a hydrophone and a dynamic position system. A matlab script was made to control the measurements and process the results. The transducer and the hydrophone were mounted on the two carriers on the dynamic positioning system. To do the experiments the tank had to be filled with water. That was done with a water hose over a period of 2 hours. The water would remain in the tank until the experiment was over. That also meant that the transducer had to remain in the water until the experiment was over. That could be several days or weeks. We were unsure if the transducer could stay in the water for several weeks. We filled the tank with a water level of 1.20 m, which meant there was a problem getting the transducer out of the water due to limitations in the ceiling height, the water level and limitations in carrier movement. Filling and emptying the tank between each measurement would be a tidies and time consuming task. After each filling we had to wait up to a week to get the right temperature in the water. When testing the system we saw that the results got better if we waited a longer period. This was due to the water bubbles in the water. The water got more saturated if we left it undisturbed for a period of time. This meant we had to find a method of getting the transducer out of the water. The solution was pulling a boat fender over the transducer and removing the water inside the fender. The transducer could then stay dry between each measurement. This is illustrated in Figure 3.3

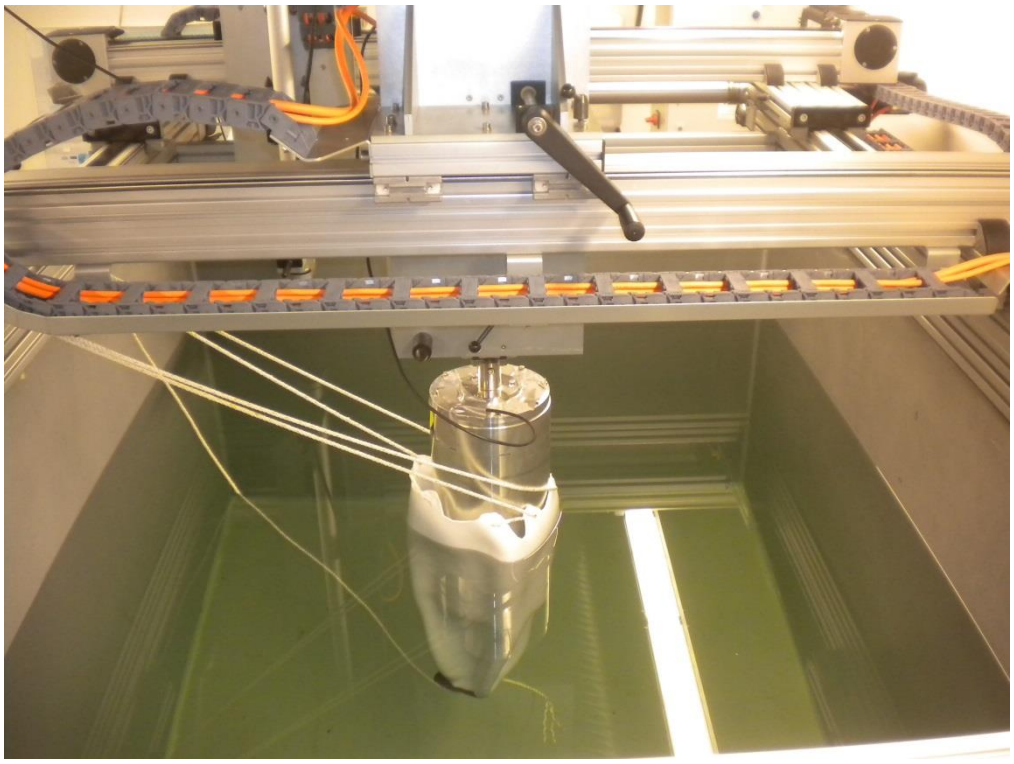

*Figure 3.3: The transducer in the water tank with a boat fender pulled over it. The water is removed and the transducer stays dry between measurements.*

The hydrophone is mounted on the carrier were the movement in x, y and z direction can be controlled by the computer. This means that we can steer the carrier between the pipes in the ceiling and emerge the hydrophone between each measurement. This is illustrated in Figure 3.4.

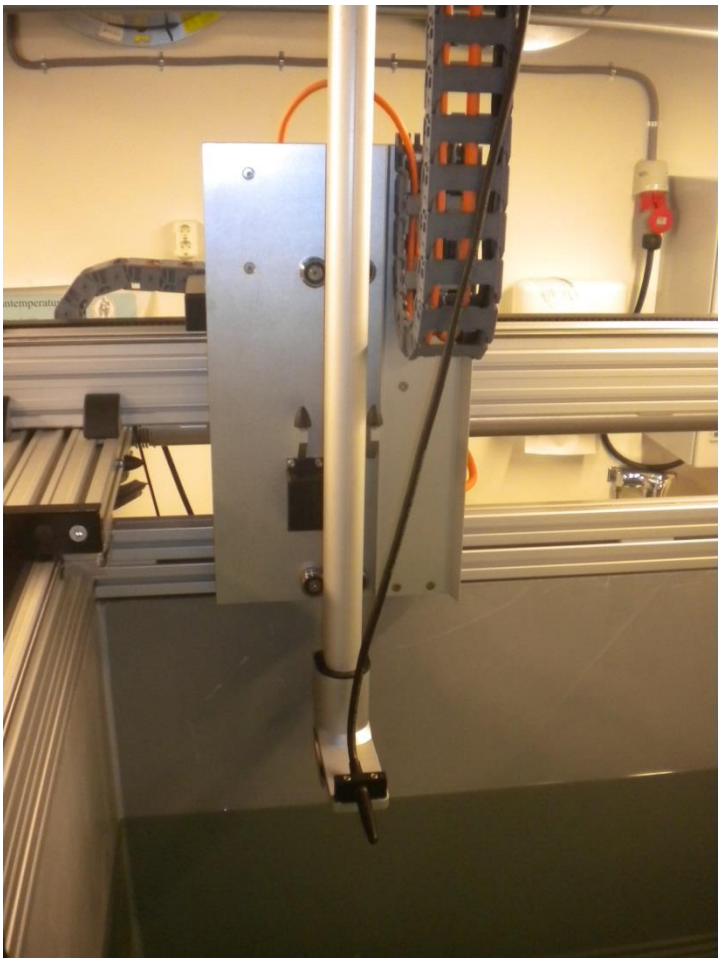

*Figure 3.4: The hydrophone emerged and steered between the pipes in the ceiling.*

Since there is no purification method we try to keep the water tank filled for short periods. This means filling the tank, waiting a week for the water and temperature to settle and then doing the experiments. All measurements in this thesis are done that way.

### 3.3.3 Recording equipment

To do the measuring experiment we had to set up a measuring system that could send and record a pulse. Everything had to be controlled using a computer and the result had to be processed in matlab. The transducer is a fishing sonar from Kongsberg Maritime [24], the rest of the equipment belongs to the sonar lab at Ifi. The equipment used in the recording rig is

*Table 3.1: Equipment used in the recording system*

| Equipment recording rig |
|---|
| Oscilloscope |
| Signal generator |
| JBL amplifier |
| Ultrasonic preamp |
| Hydrophone |
| SH-90 transducer |

The oscilloscope and the signal generator are connected to a computer running matlab. The dynamic position system is also connected to this computer. The pulse is sent from the signal generator to a transducer via a JBL amplifier. The transducer sends the pulse through the water and the pulse is captured by the hydrophone. The pulse is then sent through an ultrasonic preamp and into the oscilloscope. The matlab script then captures the pulse through the oscilloscope. This is illustrated in Figure 3.5

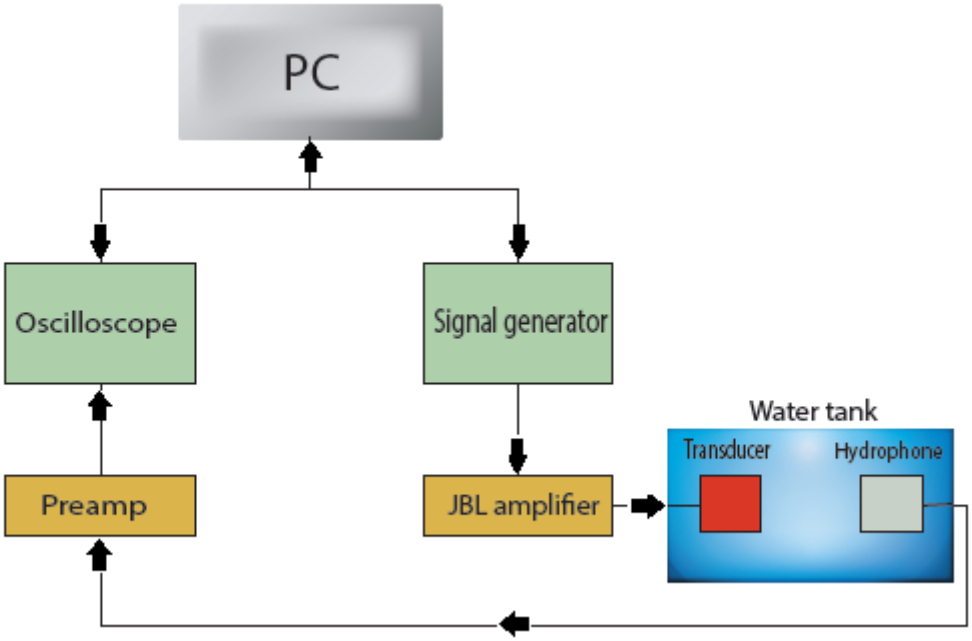


*Figure 3.5: The Flowchart of the recording system*

To connect the transducer to the JBL amplifier we made a simple circuit board. The transducer has 8 connectors which connect all elements with the possibility to connect to 60 elements. In this experiment we wanted to use 10 elements which meant we had to connect one connector with one circuit board. With the information we got from Kongsberg maritime we were able to connect the 10 elements stacked in front of the transducer close to 0 degrees. To find out how the elements were stacked we had to do a simple measurement experiment. This experiment is discussed in Chapter 4.2.

### 3.3.4 Recording of data

The recording of data is done with a matlab script programed by the author with help from a fellow student, Mohamed Kidash. In this matlab script it is possible to set the parameters for:

- Number of measuring points in x,y and z direction
- Rotation in degrees of transducer
- Parameters for oscilloscope and signal generator
- Delay time for recording in automatic mode

The program runs automatically or manually. In manually mode a button has to be pressed before each capture. All data captured are saved in mat files.

# Chapter 4

## 4. Measuring experiment

### 4.1 Equipment

The equipment used in the water tank experiment was a sonar transducer SH-90 [24], a dynamic position system and a water tank.

#### 4.1.1 Sonar SH-90

The Simrad SH90 is omnidirectional high frequency sonar (Figure 4.1). The operational frequency is 114 kHz. The SH90 is a cylindrical multi-element transducer that allows the omnidirectional sonar beam to be tilted electronically from +10 to -60 degrees.

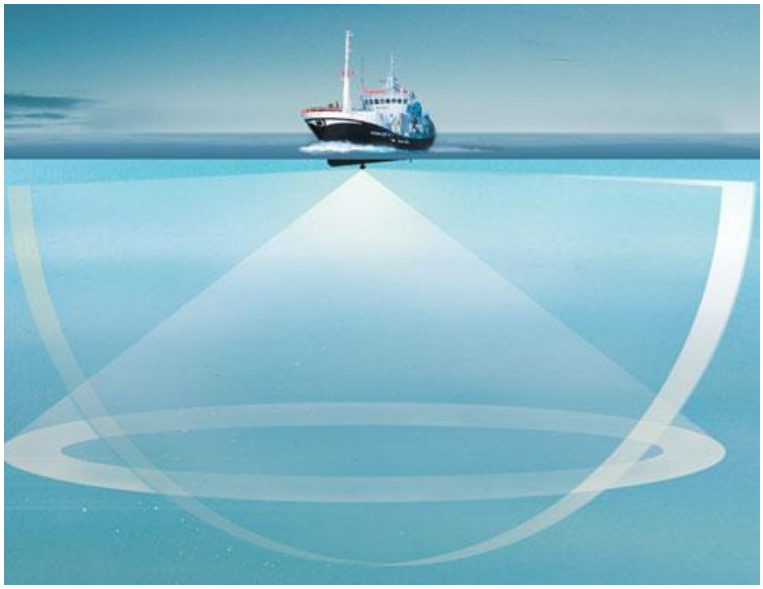

*Figure 4.1 Omni beam principle of the SH-90 sonar [Simrad 2014]*

It has 480 individual elements spread around the cylindrical transducer. The SH-90 transducer beam has a vertical opening angle of 7,5 degrees and a horizontal opening angel of 360 degrees [25]. The SH-90 transducer has a range up to 2000 meters.

#### 4.1.2 Water tank

The tank used in this thesis is a $[l, b, h] = 2x1{,}5x1{,}5\ m^3$ big water tank. It is located at the sonar laboratory at the Ifi building at Blindern. A picture of the water tank can be seen in Figure 4.2.


*Figure 4.2: Water tank at the sonar laboratory with the dynamic position system mounted on top*

### 4.1.3 Dynamic positioning system

The dynamic position system is mounted on the top of the water tank. The system is custom made by Fosstech AS [7] and is a semiautomatic system that can position a hydrophone and a transducer in x-, y- and z- directions and rotate about the z- axis. The positioning system communicates via a computer and is controlled with a matlab script. There are two positioning carriers on the system. The one farthest from the laboratory door, in the southern end of the room, can be operated programmatically along four axes. The four axes are three for spatial positioning and one for rotation angle. We call this carrier the 4D carrier. The one in the northern end only has a single computer operated axis (rotation angle). The other axes must be handled manually. We call this carrier the 1D carrier. A drawing of the system can be seen in figure 4.3

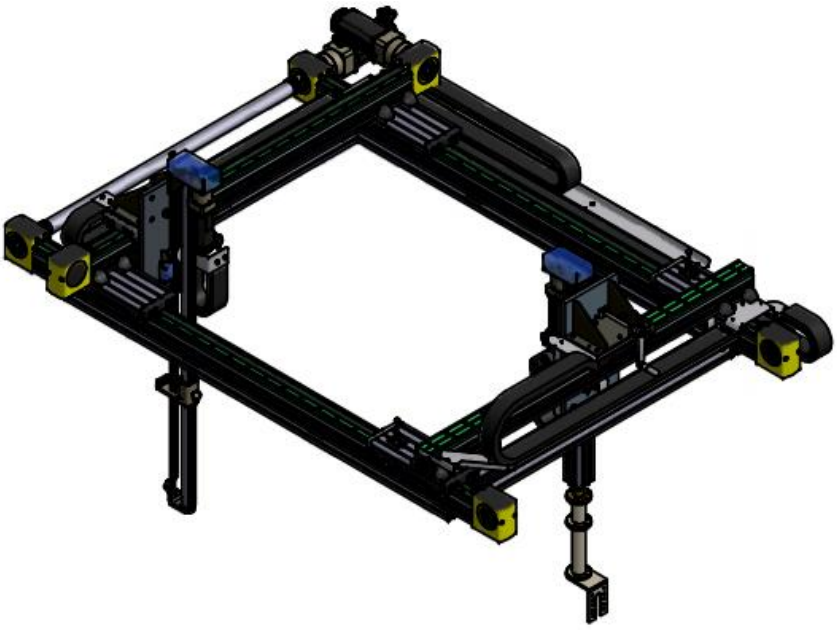

*Figure 4.3 Schematic drawing of Fosstech dynamic position system [Fosstech 2014]*

The carriers can hold a transducer and both of them can work as a transmitter or receive. One can hold a transducer while the other holds a passive object (phantom). The 4D carrier has a weight limitation of 0.5 kg and the 1D carrier has a weight limitation of 8 kg.

The 4D carrier can move in X, Y and Z directions at a speed of 0-100 mm/s with an accuracy of ± 0.2 mm when it is stationary. Rotational movement has a speed of 0-10 r/min with a precision of ± 0.1 °.

The 1D carrier can manually move in X, Y and Z direction. By entrenched position accuracy of ±0.5 mm. Rotational movement is automatic and has a speed of 0-10 r / min with a precision of ± 0.1 °.

Due to the limited headroom in the sonar laboratory there are some limitations to transducer and hydrophone movement. This is mainly in the z direction due to ventilation pipes in the ceiling. This is something you have to take into account when programing the position system.

## 4.2 Measurements and data

### 4.2.1 Determining the speed of sound in the water tank

The speed of the sound is dependent on several different factors. Sound travels faster in liquids and solids than it does in air. In water the sound travels about 4 times faster than in air.

Determining the exact speed of sound in the water tank is therefore important for the results of the measurements. The speed of sound in water is dependent on temperature, salinity and depth. In the water tank we use freshwater and the depth is so small that the speed of sound is only dependent on the temperature. This means we can use the *Lubbers and Graaff's equation* [22]

$$c = 1404.3 + 4.7T - 0.04T^2 \tag{4.1}$$

where $c$ is the speed of sound and $T$ is the temperature in Celsius. The range of validity on this equation is 15-35$C^o$ and it has a maximum error of 0.18 $ms^{-1}$.

By using Eq. (4.1) on different temperatures we get:

*Table 4.1 Calculated speed of sound in fresh water at different temperatures*

| **Temperatures ($C^o$)** | **Speed of sound (m/s)** |
|---|---|
| 16 | 1469.3 |
| 17 | 1472.6 |
| 18 | 1475.9 |
| 19 | 1479.2 |
| 20 | 1482.3 |
| 21 | 1485.4 |
| 22 | 1488.3 |
| 23 | 1491.2 |
| 24 | 1494.1 |
| 25 | 1496.8 |
| 26 | 1499.5 |
| 27 | 1502.0 |

To confirm these results we did a simple experiment in the water tank. By measuring the temperature in the water tank and the time of arrival a pulse at two different ranges i.e. different points in the y direction, we can look at the time difference between the two points. Since we know the distance between the two points we can use the equation for speed:

$$v = \frac{s}{t} \tag{4.2}$$

where $v$ is the speed, $s$ is the distance and $t$ is the time, in our case the time difference.

By using Eq. (4.2) we can find the speed the pulse travels with and thereby the sound speeds in the water tank. We did a measurement at 20 degrees Celsius. We measured the temperature in the water tank and measured the received pulse at 17.5 cm and 37.5 cm i.e. a range difference 20 cm. By doing this we compared the result with the calculated results in Table 4.1

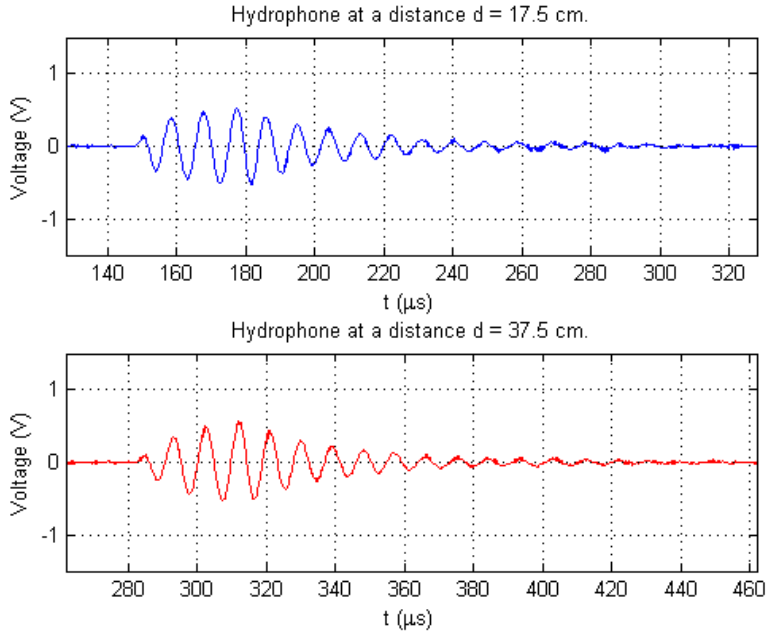


*Figure 4.4: The received pulses at 17.5 cm and 37.5 cm*

We see in Figure 4.4 that the two pulses arrive at two different times. The first pulse arrive sometime around 150 µs the second pulse arrive sometime around 185 µs. If we zoom in on the plot we can get a more exact time.

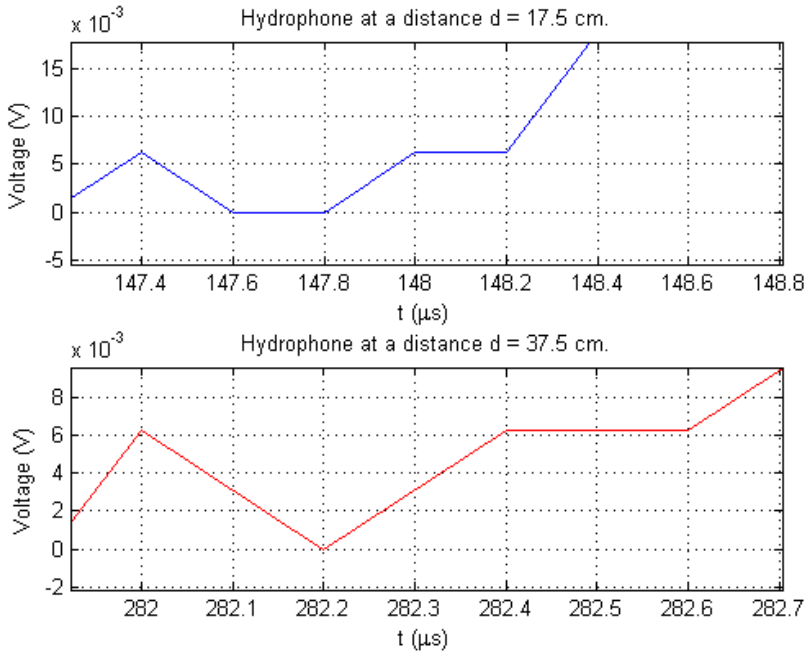


*Figure 4.5: times when the pulses is received at hydrophone*

We see in Figure 4.5 that the first pulse is received at 147.8 µs, this is the beginning of the oscillation. The second pulse is received at 282.2 µs. The time difference is 282.2 µs – 147.8 µs = 134.4 µs. If we use equation 4.6 the speed of sound in the water tank becomes:

$$v = \frac{0.2m}{134.4\mu s} = 1488.1\ m/s \tag{4.3}$$

We see in Eq. (4.3) that the speed of sound in the water tank was measured to 1488.1 m/s. If we compare this result with those calculated in Table 4.1, we see that this speed corresponds to a temperature between 21 and 22 degrees Celsius. For 20 degrees Celsius the speed should be somewhere between 1482.3 m/s and 1485.4 m/s. The measured speed is a little faster but there can be many sources of error. The measured temperature is not exact. Measuring the exact temperature is difficult. The temperature may differ +/- 1 degree. The distance of 0.2 m is not exact due to transducer placement. The reading of time values is not exact. We see that for a 0.1 or 0.2 µs difference in time value the speed of sound changes significantly. An example: if the second pulse arrives at 282.4 the speed of sound is 1485 m/s. That gives us a

temperature between 20 and 21 degrees. With a 0.3 difference the speed of sound is 1483 m/s around 20 degrees.

### 4.2.2 Testing the measuring system and determining element position

This was done as one of the first things to test the measuring system and to determine how the different elements are stacked in the transducer and where the maximum values in the main lobe are. It also gave us an idea on what the sonar field looked like. By measuring at 11 different positions in the x direction and 27 different positions in z direction, we got 297 measuring points in a grid 0.33 meters from the transducer. The reason for 27 points in the z is to get a good resolution in z direction. By using equation for the wave length $\lambda = \frac{c}{f}$ and Eq. (2.5) we can find the transducers opening angle in the vertical z direction.

$$\lambda = \frac{1480m/s}{100kHz} = 0.01480m = 14.80mm \tag{4.4}$$

In Eq. (4.4) we used 1480 m/s which are the speed of sound at 19 degrees Celsius. This gives us a wavelength of 14.80 mm. The outer dimension of the transducer is slightly more than 11 cm in vertical z direction. If we assume that the transducer has 10 elements stacked in the z direction and choose each element height to be 10 mm with a minimum spacing of 1 mm we get an array length of 110 mm which gives us an opening angle of:

$$\beta = \frac{14.80mm}{110mm} = 0.1345rad \tag{4.5}$$

By using the equation for conversion between radians and degrees we get an opening angle.

$$angle\ in\ radians * \frac{180}{\pi} = 0.1345rad * \frac{180}{\pi} = 7.7089\ degrees \tag{4.6}$$

In the specs for the SH-90 transducer [25] it is said that the transducer has an opening angle of 7.5 degrees. Our calculation is not far from that, an error around 0.2 degrees. The causes for

this error can be the temperature in the water is not exact and the spacing between the elements is different. The size of the elements can also differ. This result confirms that the transducer has an opening angel on 7.5 degrees in the vertical direction. To get a good resolution in the vertical z direction we have to have more than one point inside the opening angle at range 0.33 m. To find how big the opening is at 0.33 m we used simple trigonometry. A simple drawing of the aperture is illustrated in Figure 4.6.

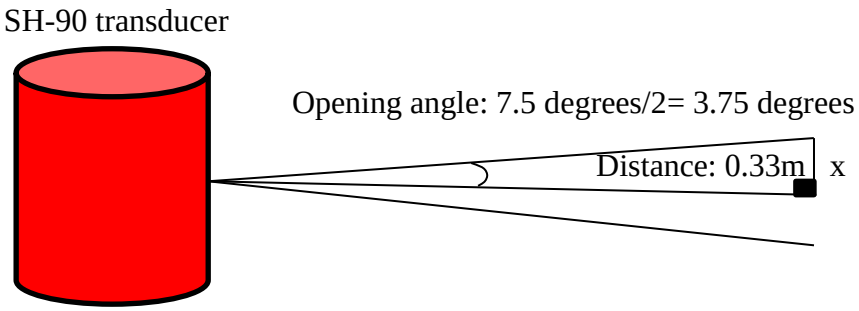


*Figure 4.6: the aperture angle in the vertical z direction on the SH-90 transducer*

By dividing the aperture angel on two we get an orthogonal triangle with an adjacent leg 0.33m and half an opening angle 3.75 degrees. We can then use the tangent function to find the opposite leg.

$$\tan v = \frac{Opposite}{Adjacent} \tag{4.7}$$

We are interested in finding the opposite leg, so by turning the tangent function around we get

$$\tan(3.75\ degrees) * 0.33m = 0.229m \tag{4.8}$$

This gives us an opening at 0.33 m which is$0.2299m * 2 = 0.4598m$. This means 27 points in vertical z directions we can have 19 points inside the opening angle and 8 points outside. By moving the hydrophone $\frac{0.4598m}{19points} = 0.0242m$ steps for each measuring point, gives us a total of 0.634 m of total movement in vertical z direction.

Instead of moving the hydrophone in 297 different positions around the grid we placed the hydrophone in front of the transducer, only moving it in 27 different positions in vertical z direction. By rotating the transducer 7.2 degrees for each measurement we got a grid of points around the transducer 0.33 meters in y direction. This gave us a sonar field with a 72 degrees opening in x direction. The main goal with this experiment was to find the main lobe of the transducer and the maximum point in vertical z direction. The experiences from these measurements were use in later experiments were the measurements were done in one plane. By finding the maximum in vertical z direction, we were able to do rest of the measurements in this thesis in this z point. This meant that we only had to move in x and y direction in the tank.

The measurements were done automatically with a series of matlab scripts. The dynamic position system is controlled by matlab scripts programed by engineer Svein Bø [2] at the DSB group. The recording of the data were done with matlab scripts programed by the author.

The measurements in this experiment were done over 2 hours in November 2013 in automatic mode. The signal generator was set to send a sinus pulse with 3 cycles and center frequency at 100 kHz. The data were processed in the matlab.

By looking at the plots of the data we can see that this experiment answers some of the basic questions, does the system work? What does the field look like and where is the vertical maximum point?

To answer the first question does the system work, we can look at the received pulse at one of the measuring points.

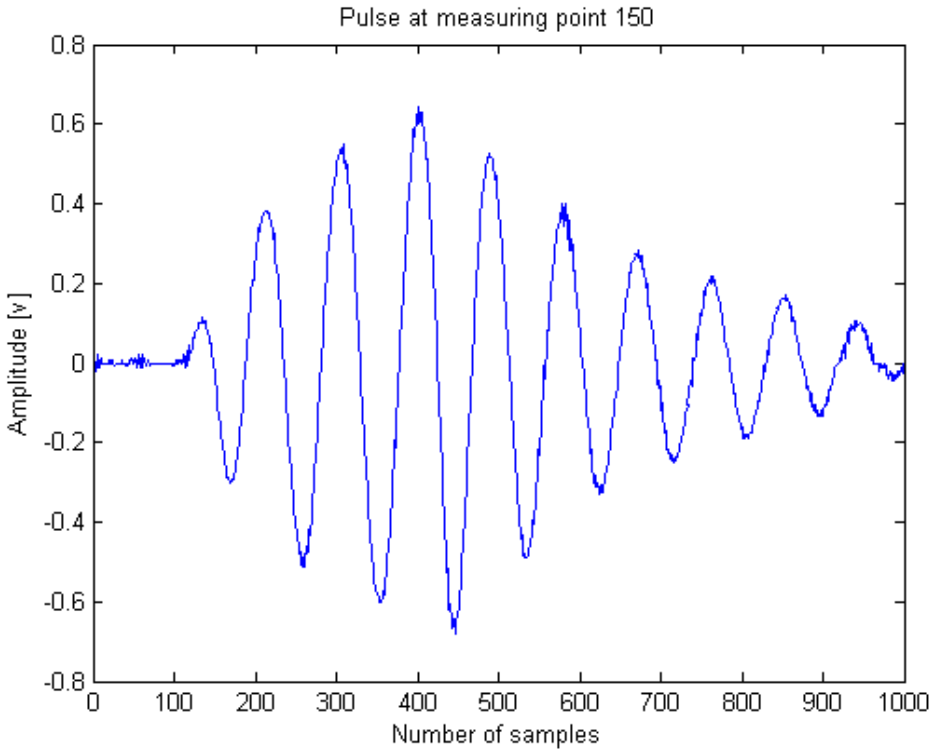


*Figure 4.7: Received pulse at measuring point 150*

We can see in Figure 4.7 that the received pulse oscillates more than the transmitted pulse. We transmit a pulse with 3 cycles and receive a pulse with 8 cycles. This suggests that there is certain inertia in the transducer. The elements use time to get the pulse to oscillate. The pulse also uses time to calm down i.e. we get ringing in the tail of the pulse. This means the transducer is narrow band. The system works and the transducer transmits and the hydrophone receives.

What does the sonar field look like? After doing the measurements in the water tank we process the data and plot a dB plot of the sonar field:

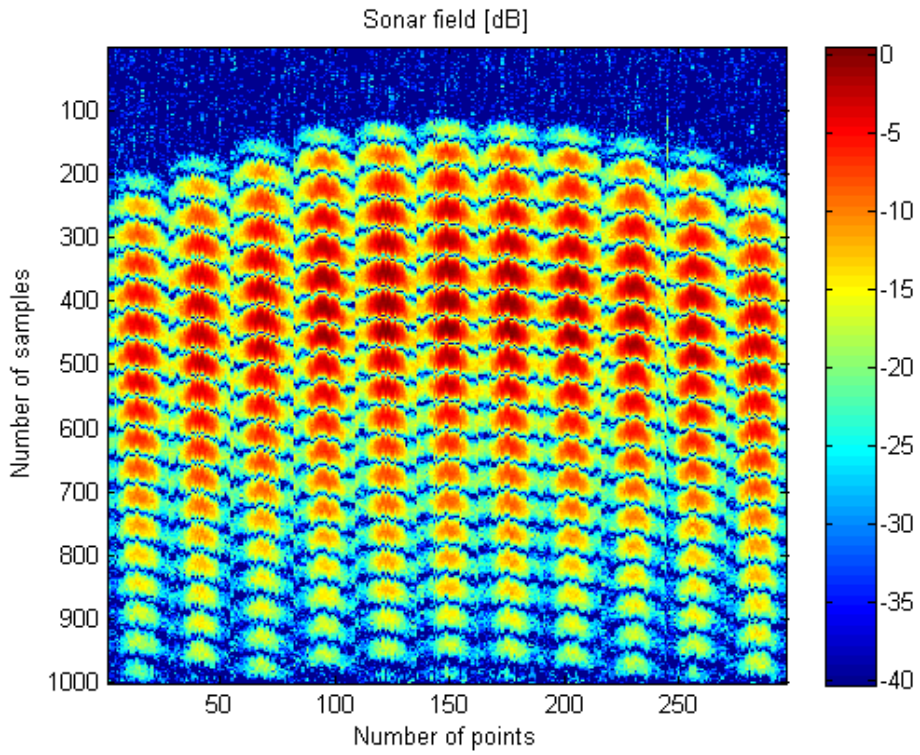


*Figure 4.8: Sonar field in dB with 11 points in the x direction and 27 points in the z direction a grid of 297 measuring points at a distance of 0.33 m from the transducer.*

We see in Figure 4.8 that when we plot the sonar field in dB the resolution in x direction is not very good with only 11 measuring points. But we get a good idea of what the sonar field looks like. For future measurements it is necessary with more measuring points in x direction to get an acceptable resolution. In figure 4.8 we can see that the maximum values lie some were between point 100 and 200. If we look at the field in a three dimensional plot it is easier to see that the maximum point lies in the middle of the transducer.

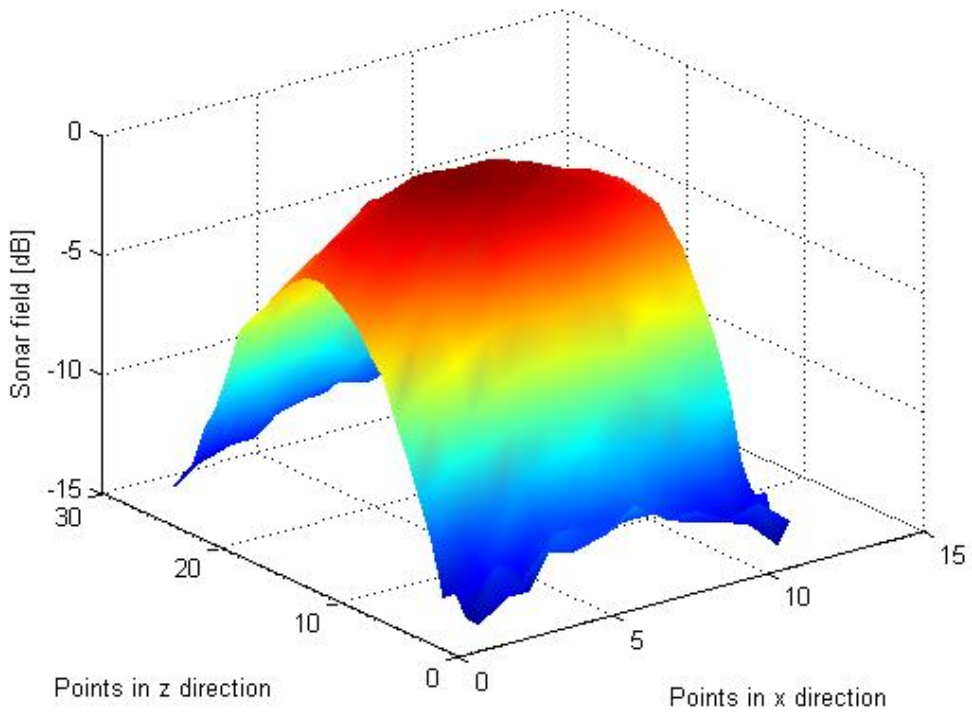


*Figure 4.9: Three dimensional plot of the sonar field. The x, y and z axis is placed like the coordinate system in the water tank.*

In Figure 4.9 we see that the maximum values are around measuring point 15 in z direction and point 6 in x direction. This means that the transducer has its maximum in front with the main lobe pointing outward from the transducer. By closely looking at the values in the data files we find the vertical z positions for the maximum values. This z positions are used in later measurements in one plane. In the x direction we see that the main lobe do not meet the -3dB limit. This suggests that we need a bigger opening angle in x direction than 72 degrees.

### 4.2.3 Testing the sonar field in one plane

The idea with this experiment was to test the sonar field in one plane. With the experience we made in the last experiment the main focus in this measurement was increasing the resolution in x direction and the opening angle. We also wanted to test the manual mode and see if this had an impact on the measurements. The questions for this experiment were how the sonar fields in one plane are and how many measuring points are needed for a good measurement.

In this measurement we choose to use 180 measuring points in x direction. The reason for this was that we wanted to do the measurement from +90 to -90 degrees which would give us a 180 degree opening around the transducer. With 180 points the transducer would have to rotate 1 degree for each measurement. This would give us an idea on the sonar field with better resolution in a wider angle.

We did two measurements in this experiment with two different ranges. One measurement was done close to the transducer 17.5 cm and one at 37.5 cm from the transducer. By doing two measurements with different range we could see if the sonar field changed with distance.

The measurements in this experiment were done over 8 hours in February 2014 in manual mode. The signal generator was set to transmit a sinus wave in burst mode with 3 cycles and center frequency at 100 kHz. The data were processed in the matlab script maling_10_4.

By looking at the plot from these measurements we see that the resolution was increased significantly with 180 measuring points.

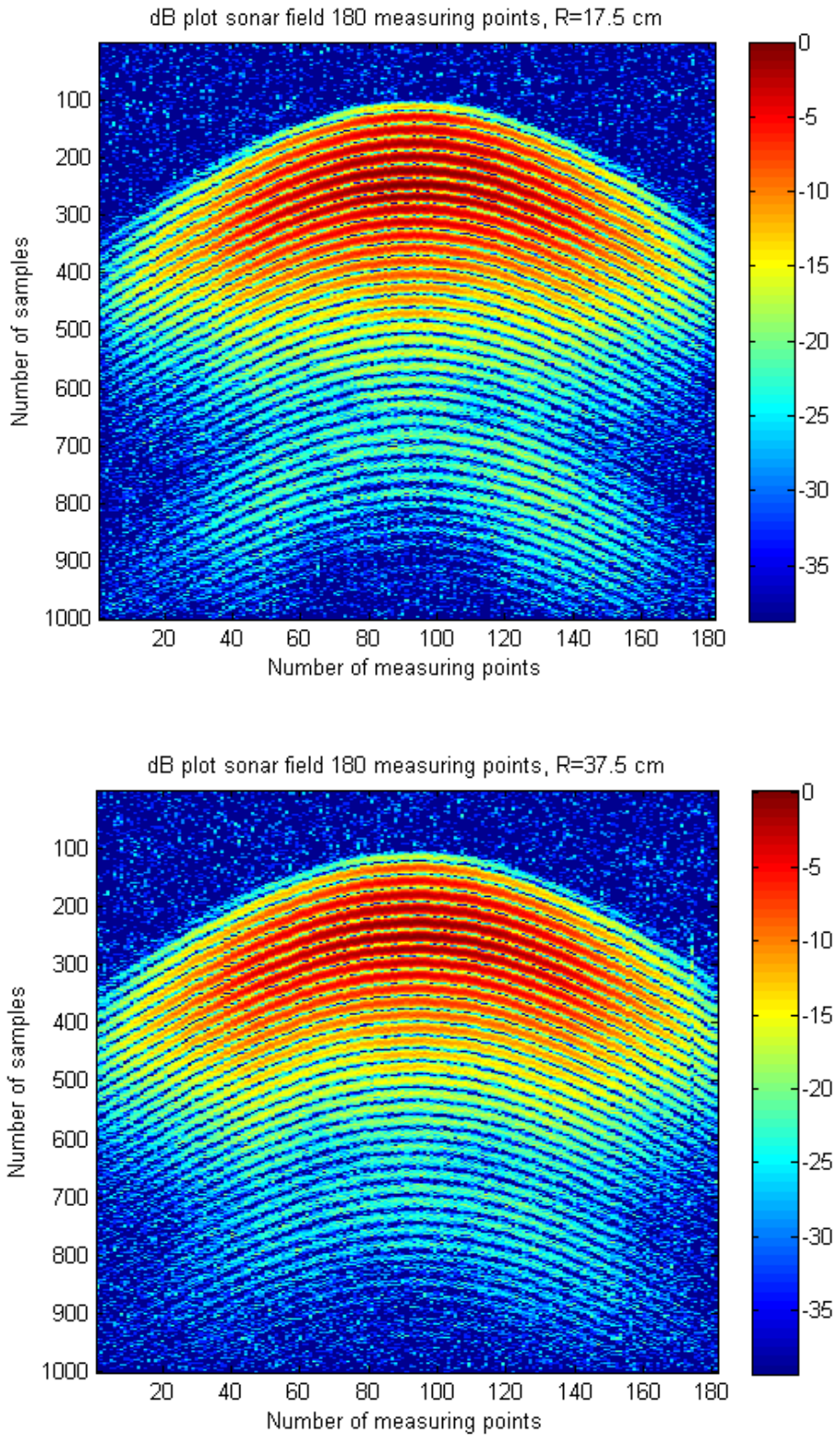


*Figure 4.10: Sonar fields at R= 17.5 cm and R= 37.5 cm. Both measurements were done with 180 measurements.*

We see in figure 4.10 that the resolution in x direction is good with 180 measuring points. There is some quantization noise in both measurements; it is evident in the areas with low dB

value. There are more ringing in the received pulse in the 17.5 cm measurement than in the 37.5 cm measurement. We can also see that since the measurement at 17.5 cm is closer to the array there is slightly more curvature in the field. The problem with these measurements is that with 180 measuring points there is not an even number of measuring points for each element. This can be a problem in the summation of different points and back propagation of the field. This will be addressed in the next experiments.

### 4.2.4 Measuring the sonar field

The measurements done in the tank up to this point has been to find the speed of sound in the water tank, test and calibrate the system, test number of measuring points and determine how the sonar field looks. To measure the sonar field in the tank we have to determine how many measuring points we need to get a good resolution. In the simulations we saw that if I used under 100 measuring points in x direction the resolution was not very good. This was confirmed by the early measurements in the tank. The resolution is dependent on the number of measuring points for each element. This can be found by doing a simple calculation. We have a 360 degree cylinder with 48 elements stacked around the cylinder. Since the measurements are dependent on the rotation of the transducer we have to find how many degrees each element represents.

$$\frac{360^o}{48\ elements} = 7.5^o \tag{4.9}$$

We see in Eq. (4.9) that each element is representing 7.5 degrees. This means that we have to use an even number of measuring points inside 7.5 degrees. We do the measurements from +90 to -90 degrees which means we use 24 elements in front of the transducer. To get a good resolution we need more than four measuring point for each element, $24 * 4 = 96$ points. This means that the minimum number of points for each element is 24 elements*5 points=120 points. This will give us a good resolution and with 5 points for each element the transducer has to rotate $\frac{7.5degrees}{5points} = 1.5\ degrees$ for each measurement. This gives us a field of view of 180 degrees around the transducer.

Two measurements with 120 points were made with two different ranges. The first measurement was done with a radius 0.5 m from the transducer. The second measurement was done as close to the far-field as possible with a radius 1 m from the transducer. These measurements and results are presented in Chapter 5.2.

# Chapter 5

## 5. Results and discussion

In this part the results of the measurements and simulations are presented. We mainly look at the results from the measurements presented in Chapter 4.2.4 and the simulations in Field II with a curved array presented in Chapter 3.1.5. The first measurements leading up to the measurements presented in this Chapter were for testing and calibration and the result from these are presented in Chapter 4.

## 5.1 Simulation data

### 5.1.1 Linear array

The results from the simulations with the linear array provide the foundation for the simulations with the curved array. The objective with these simulations was:

- Measure the sonar field in the far-field.
- Back propagate the field and look at the response over the elements.

By confirming these statements with a linear flat array it is easier to try to recreate the same data with a curved array.

The simulations with the flat array were done with the matlab scripts ForwardProp.m and BackwardProp.m. The first simulation was done with a 10 element flat array with 120 measuring points at a distance 1000 m from the array. The measuring points were placed in a 180 degree half circle around the array and the pulse period was set to 3.5 periods. The sound speed was set to the measured sound speed of 1488 m/s. When we plot dB plot of the field we get:

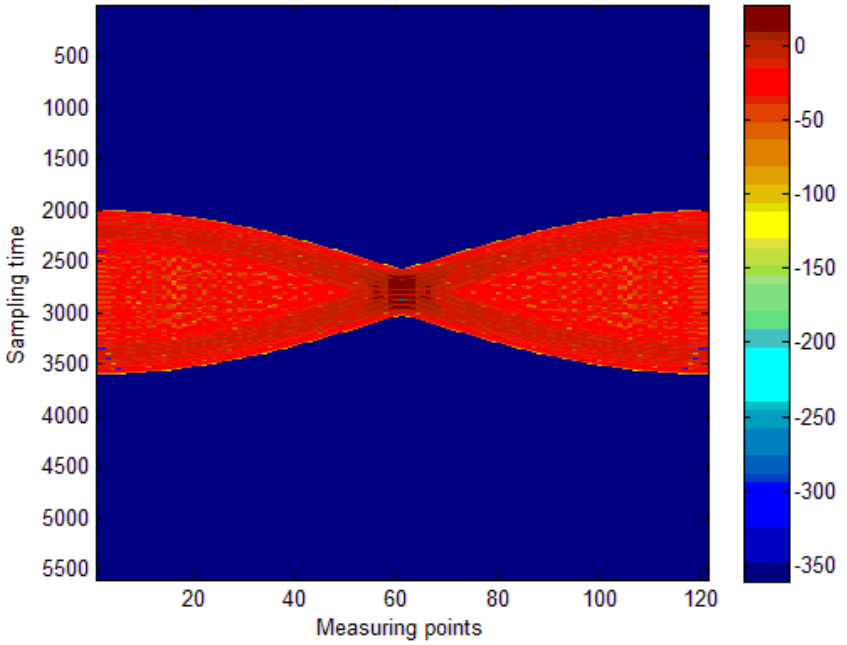


*Figure 5.1: dB plot of the wave field with a 10 element linear array with 120 measuring points*

In Figure 5.1 we see the dB plot of the wave field. In Chapter 3 we discussed the geometry of the measuring sequence (Figure 3.2). If we compare the field in Figure 5.1 to what we expected in Figure 3.2 we can clearly see the difference in arrival time of the different pulses. When we back propagate the wave field back to the array we get:

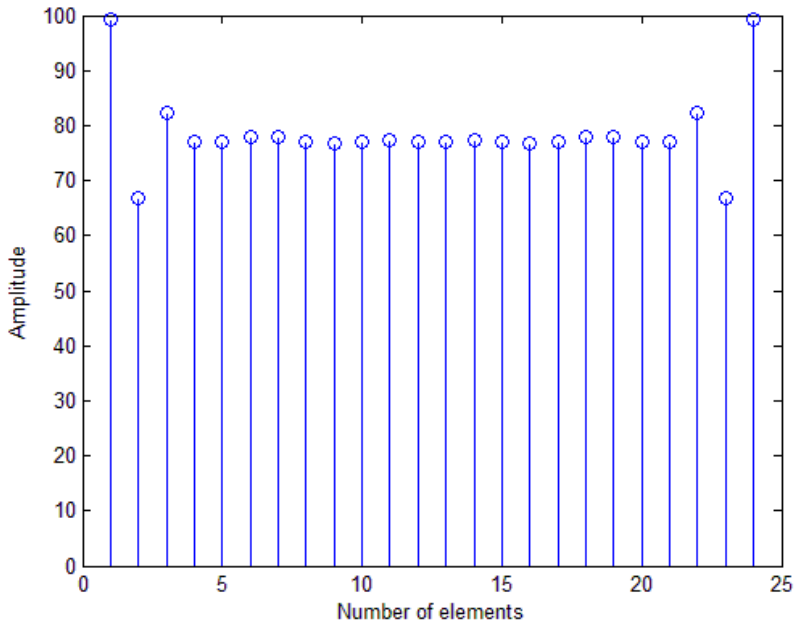


*Figure 5.2: Response over all 24 elements with 120 measuring points distance 1000m*

We observe in figure 5.2 that we have some fluctuations in the amplitude values in the outer elements but the response is smoother in the inner elements. We tried to improve the result by placing the measuring points in a 120 degree sector in front of the array. When back propagating these data we got:

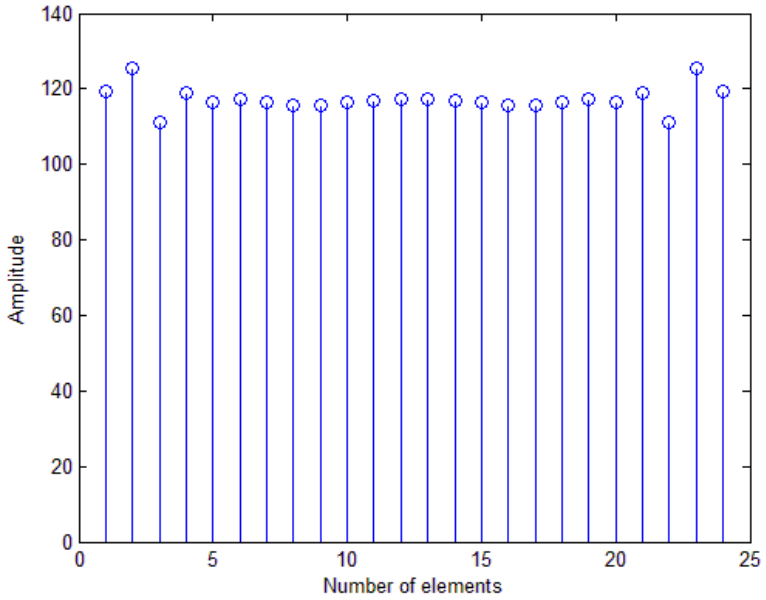


*Figure 5.3: Response over all 24 elements with 120 measuring points at distance 1000m in a 120 degree sector in front of the array.*

We observe in figure 5.3 that by measuring in a sector of 120 degrees in front of the array we get a slightly better result. The amplitude values are a little higher than in the first simulation and the fluctuations in the outer elements are a little less. This gives us an idea of how to do the simulations with a curved array

### 5.1.2 Curved array

The simulation with the curved array was done with the matlab script KjorKode.m. The main purpose with these simulations was too see if the measurements in the near-field were possible and if we could characterize a transducer in the near-field. By looking at the response over all the elements we could see how sensitive the system was. By testing the system with different sound speeds, number of elements and pulse lengths, the idea was to find the setup that gave the minimum least square error.

In the ideal situation the response over all elements are the same when we back propagate. In the ideal situation we set the parameters to:

- 10 elements in the array
- Input signal frequency 100 KHz
- Sampling frequency 100* Input signal frequency
- Speed of sound was measured to 1488 in the water tank so c is set to 1488
- 120 measuring points
- Distance of 1 m, this gives us measurements in the near-field
- Measuring points in a 120 degree sector in front of the array

These parameters are set in the matlab script Parameters.m and by running KjorKode.m we get the response over all elements:

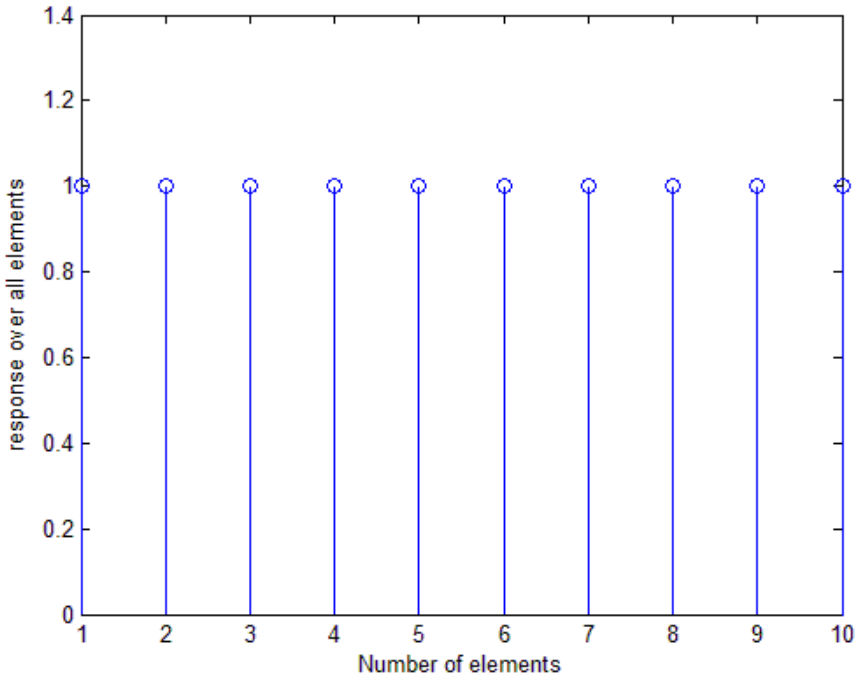


*Figure 5.4: Response over all 10 elements in ideal situation with distance 1m and speed of sound 1488 m/s.*

We see in Figure 5.4 that the response over all elements is the same, centered on 1. This gives us a least square error of 2.98e-23 which is neglectable. This means the program and approach works when the different parameters are known.

To examine how sensitive the system is and if the least square error is a good indication on the response we can change some of the parameters to see what happens to the response and the least square error. In this sensitivity analyze we examined three different parameter changes:

- The change in sound speed
- Number of elements in the array
- Pulse length, number of periods in the pulse

## Speed of sound

To see how the response changes with the sound speed we did simulations with three different sound speeds, 1487 m/s, 1486 m/s and 1483 m/s. The other parameters were set to the same as in previous simulation. By changing the sound speeds the result was:

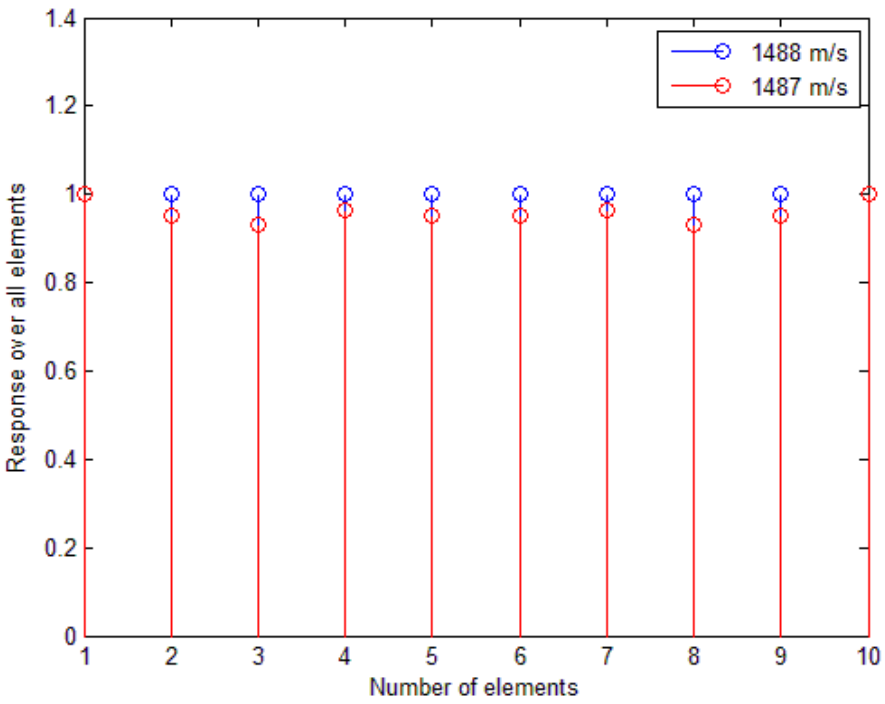


*Figure 5.5: response over all 10 elements with distance 1m and speed of sound1488 m/s and 1487 m/s*

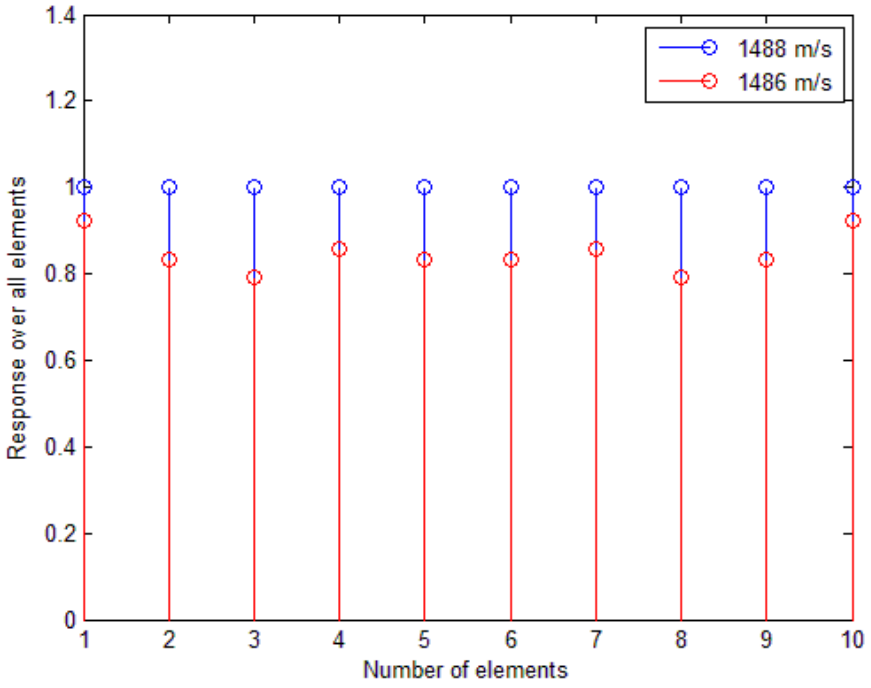


*Figure 5.6: response over all 10 elements distance 1 m and speed of sound 1488 m/s and 1486 m/s.*

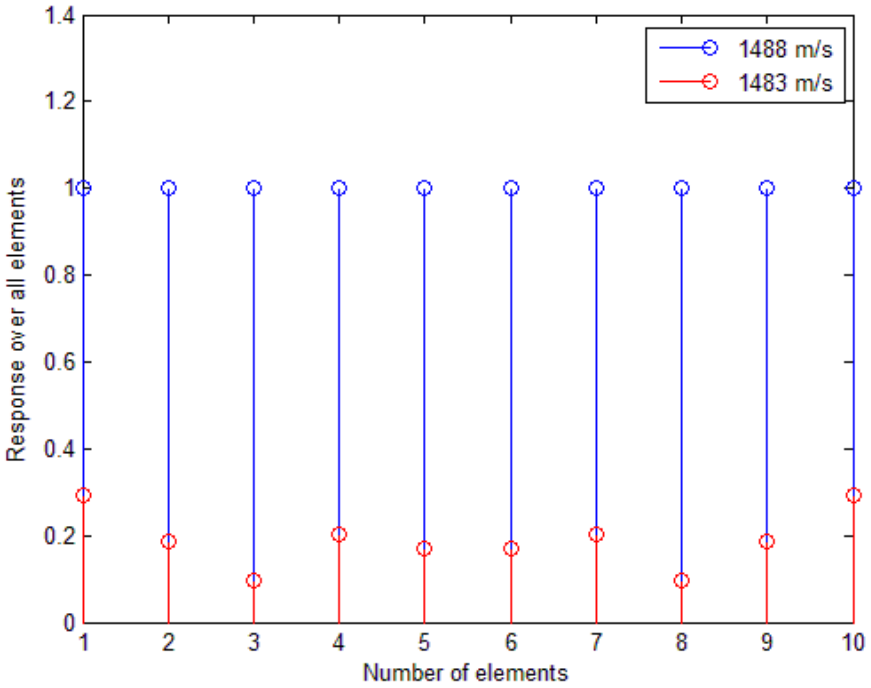


*Figure 5.7: response over all 10 elements distance 1 m and speed of sound 1488 m/s and 1483 m/s.*

We see in figure 5.5 that the response over all elements changes with 1 m/s difference in the sound speed. The least square error becomes 4.67e+3. This proves that the system is sensitive for small changes in sound speed. This becomes more visible when we change the sound speed by 2 m/s to 1486 m/s. We see in figure 5.6 that the response of the elements varies more and more with small changes in the speed of sound. The least square error becomes 1.72e+4. This gives us a large deviation from the ideal situation where the least square error was close to zero. We see in Figure 5.7 that with a difference in the sound speed of 5 m/s the response has a significant drop in value and start to vary for each element. The least square error is 5.94e+4. This shows that the system is sensitive for small changes in sound speed and thereby also the temperature in the water.

## Number of elements

In the first example we used 10 elements in the array. Will the response and the least square error improve by increasing the number of elements to 20 and 40 elements? We use 1486 m/s when we try to improve the result:

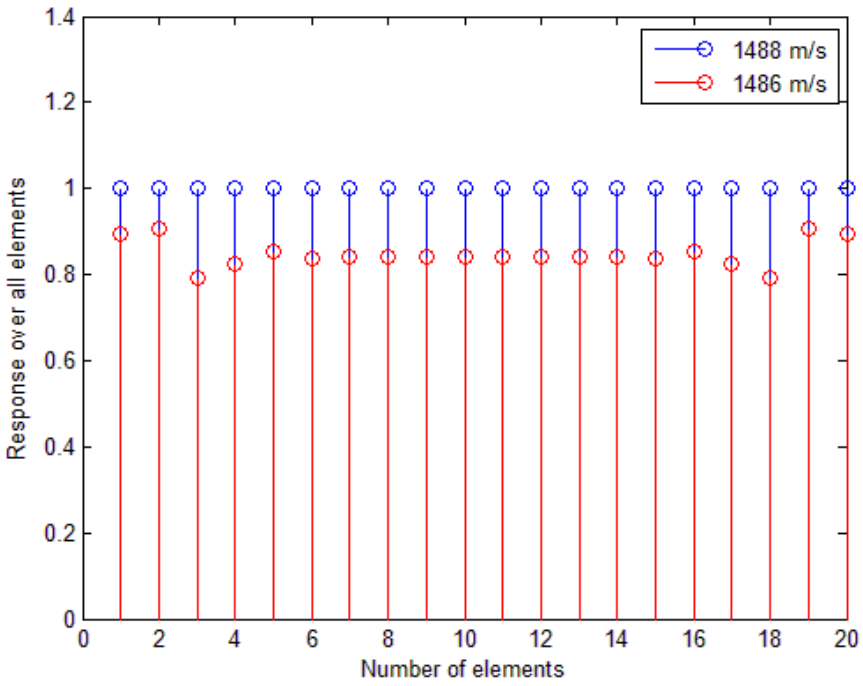


*Figure 5.8: response over all 20 elements distance 1m speed of sound 1488 m/s and 1486*

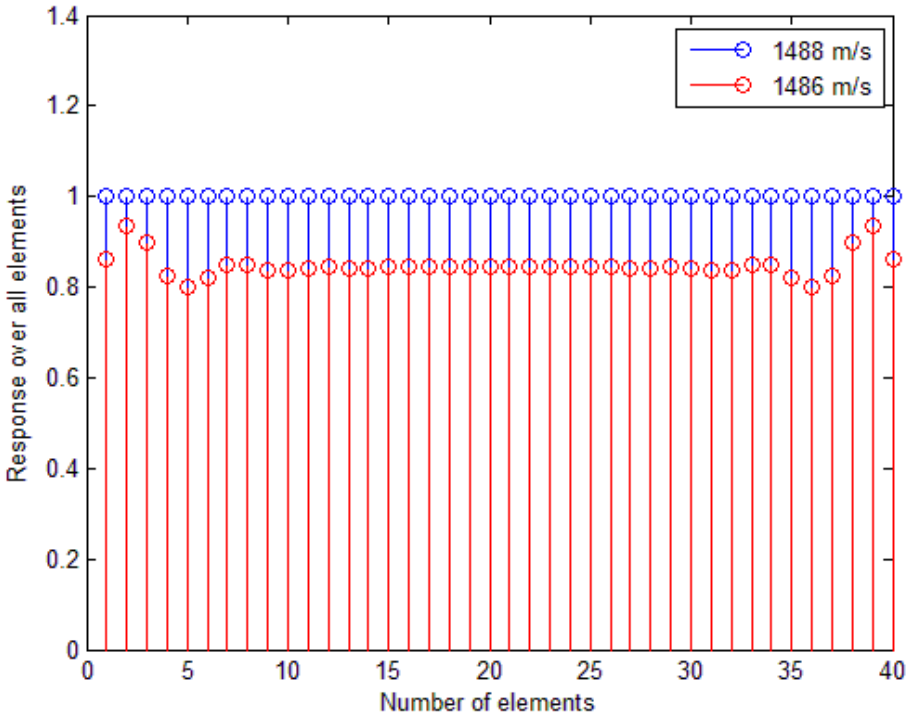


*Figure 5.9: response over all 40 elements distance 1m speed of sound 1488 m/s and 1486*

With 20 elements in the array the least square error in Figure 5.8 is 3.17e+4. The error with 10 elements was 1.72e+4 which suggest no improvement. It becomes clearer when we increase the number of elements to 40 elements: The least square error in Figure 5.9 is 5.53e+4. The error increases with the number of elements. We observe in Figure 5.9 that there are some fluctuations in the outer elements. This is also the case in the previous simulations with 10 elements but becomes more apparent when we increase the number of elements to 20 and 40. It is symmetric on both sides and becomes more apparent when we increase the number of elements.

## Pulse length

We have seen in the previous simulations that increasing the number of elements does not decrease the least square error or improve the response. If we look at the pulse that is transmitted in the first simulations the pulse periods is set to 3.5. Will the least square error decrease if we increase or decrease the pulse periods? The pulse used in the previous simulations is illustrated in Figure 5.7.

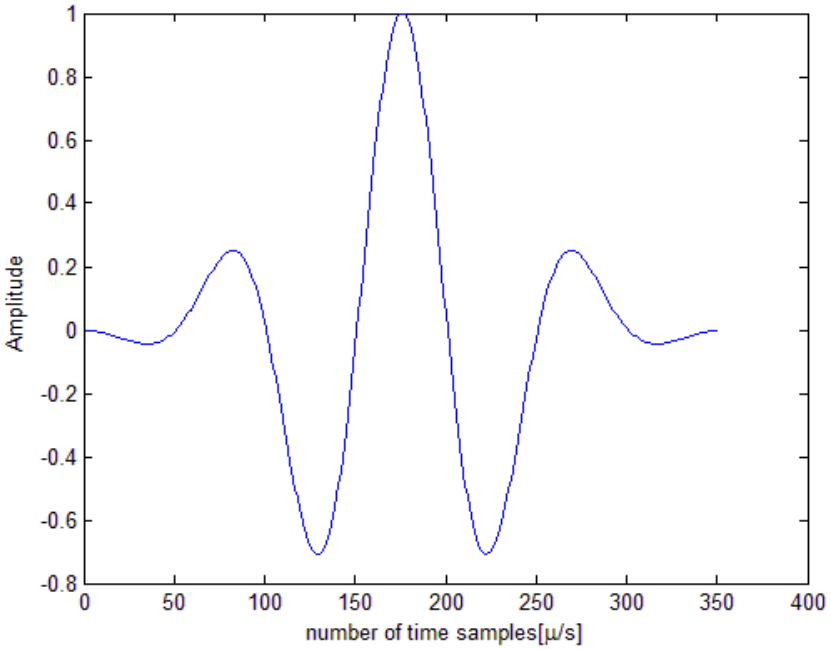


*Figure 5.10: Transmitted pulse with pulse periods 3.5*

If we change the  pulse periods to 8 and 1 we get the transmitted pulses:

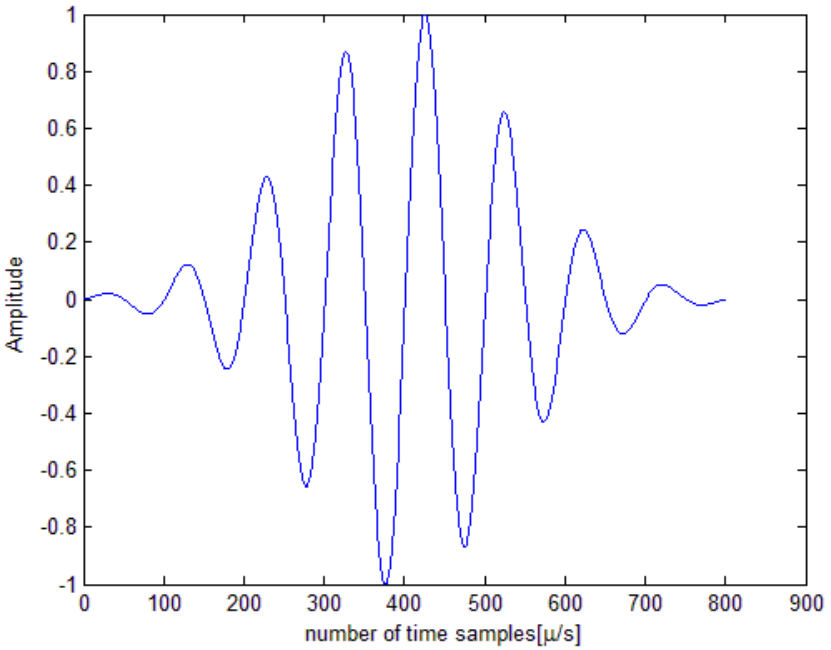


*Figure 5.11: Transmitted pulse with pulse periods 8*

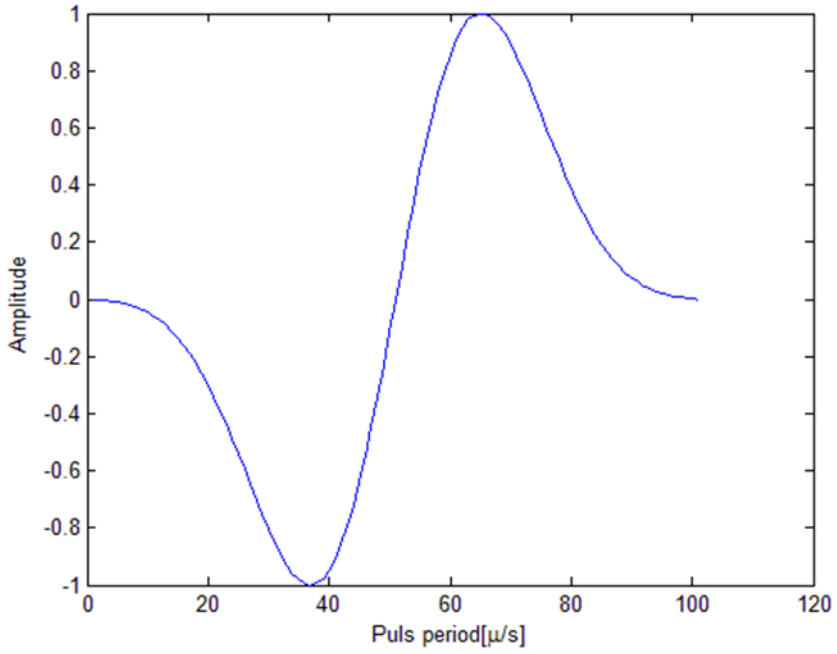


*Figure 5.12: Transmitted pulse with pulse period 1*

This gives us the responses:

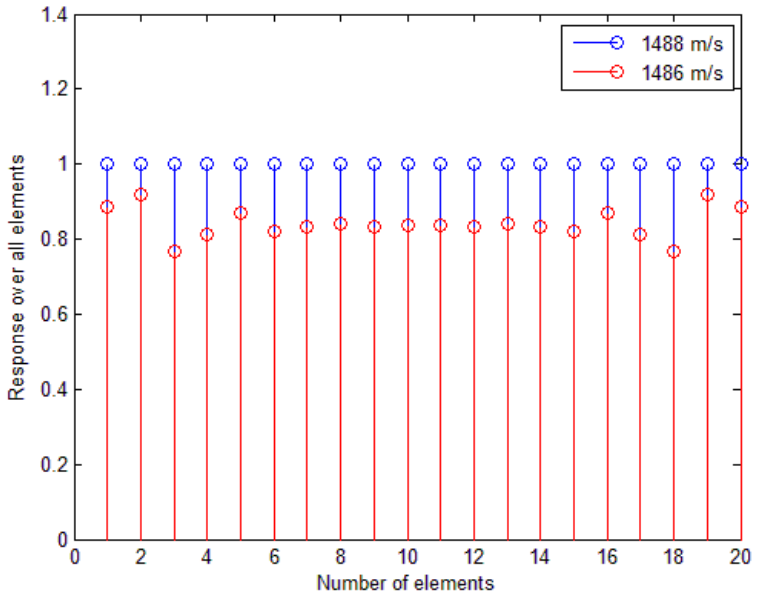


*Figure 5.13: response over all 20 elements distance 1m speed of sound 1488 m/s and 1486. Pulse period 8*

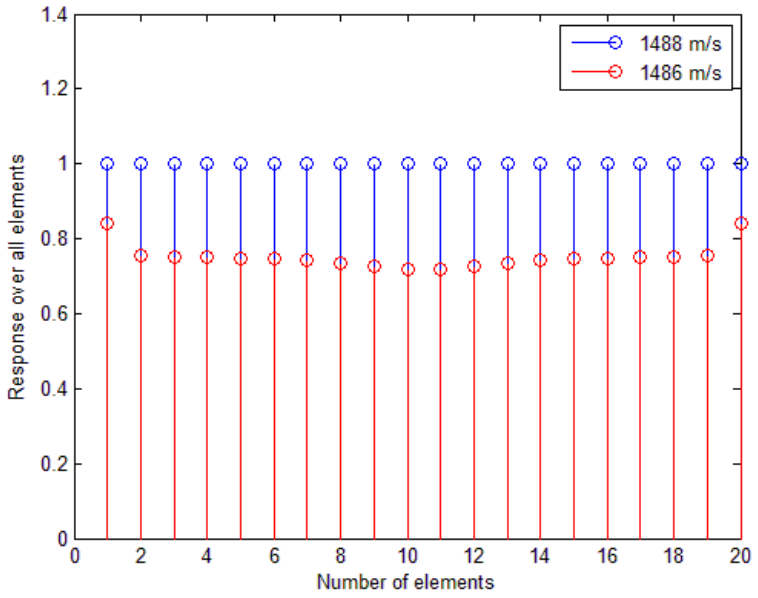


*Figure 5.14: response over all 20 elements distance 1m speed of sound 1488 m/s and 1486. Pulse period 1*

We see in Figure 5.13 that by increasing the length of the pulse to 8 periods does not give a uniform response in the elements. The least square error is 7.48e+4 so by increasing the pulse periods we increase the least square error. This is a clear indication that the result is wrong. If we decrease the length of the pulse from 3.5 periods to 1 period we get a sinusoid with one period. We see in Figure 5.14 that by decreasing the length we reduce the fluctuations in the outer elements but the response over all the elements is slightly smaller than with pulse period 3.5 and 8. The least squares error is 3.27e+4 which is still very high.

To sum up the simulations done we see that the system is sensitive to small changes in the sound speed. By examining the changes in different parameters we have seen how the response changes over the elements. By adding more elements in the array we saw fluctuations in the outer elements and an increased in the least square error. The length of the pulse did not improve the result. In these simulations the lowest least square error was by a factor of ten raised to the power of 3. This suggests that the system is very sensitive for small changes in the sound speed.  This works if we have an array where we know the pulse length and the response to the elements. This means that the least square error can be used if the only unknown parameter is the speed of sound. This means that the least square error can be used to find the right speed of sound and therefore give us a calibrated system.

We can take some of the measurements done in this simulation as an example. We know that the speed of sound is 1488 m/s but if we were to find it we could make a matlab plot of the least square errors. If we make a matlab plot of the least square errors from 1486 to 1490 m/s we get:

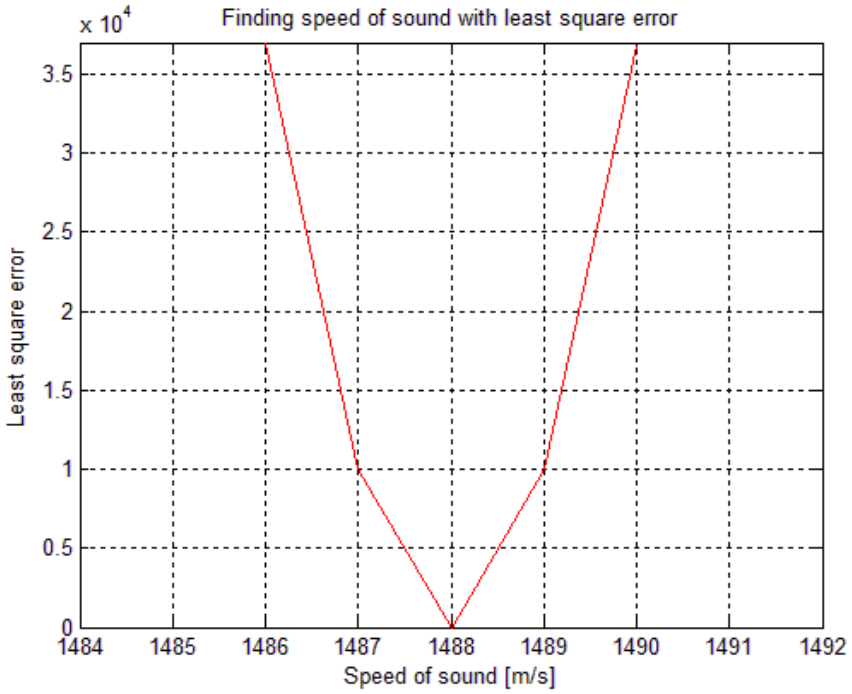


*Figure 5.15: Plot of the least square errors from 1486 to 1490 m/s*

We see in Figure 5.15 that we have a minimum on 1488 m/s which means that the speed of sound is 1488 m/s. We see in the plot that +/- 1 m/s difference in sound speed has a significant increase in least square error.

## Characterize a transducer

To characterize a transducer we have to know the element weights and phase. If we know the elements weights and phase it is possible to calculate the transducers far- field response. Given a calibrated system where the speed of sound is known, can we then determine the elements weights? Can we also find out if any of the elements have a delay?

## Element weights

In the previous simulations the elements were weighted one over all elements. If we use the previous simulated result as the measured signal we can try to find the element weights. By trying different weights we can compare weighted signal to the window function and find the max amplitude difference between the window function and the response. The array has 24 elements and the speed of sound is known and set to 1488 m/s. The test is done with two different weights first using a set window function and then a random window function. A window is shaped so that it is exactly zero at the beginning and end of the data block and has some special shape in between. The random window function is random values set around one. In this example we used a hanning window which has a bell shape.

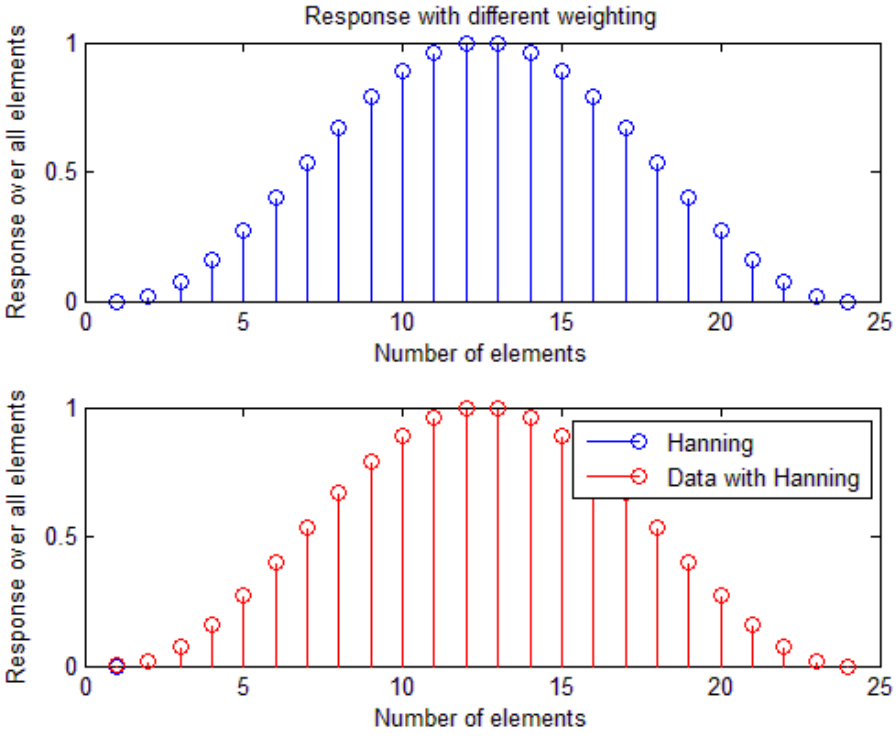


*Figure 5.16: Response on all elements with a hanning window*

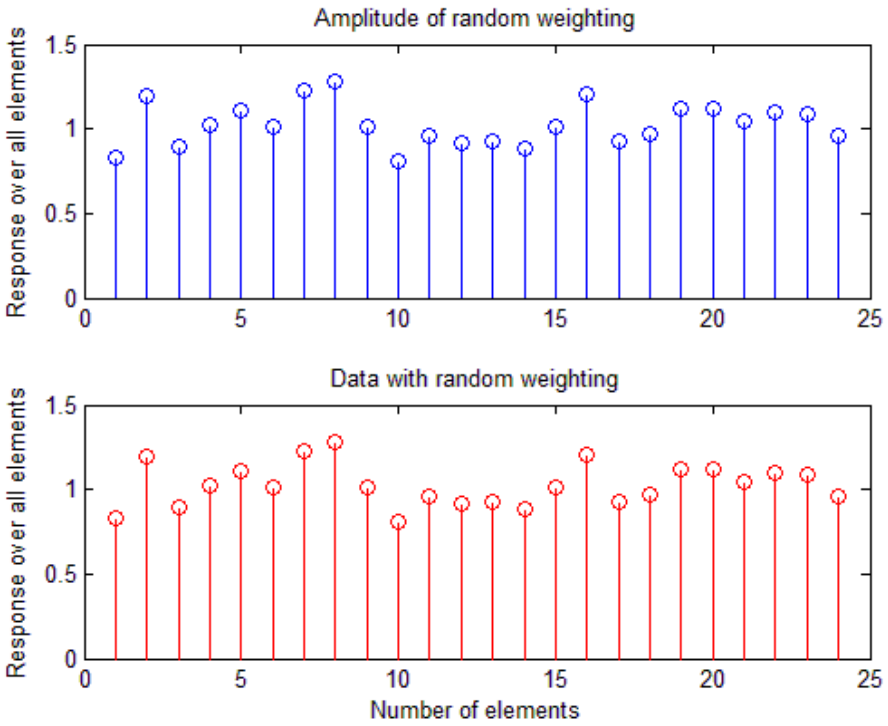


*Figure 5.17: Response over all elements with random weights*

We see in Figure 5.16 that by putting a hanning window on the data we get a bell shaped response similar to the shape of the window function. We can see that the amplitude values for the window function and the weighted data is the almost the same. If we compare the two amplitude values we find that the max differences for all elements are 1.94e-14 which is virtually zero. In Figure 5.17 we see the response after a random weighting of the elements. Like Figure 5.16 the difference in amplitude values is close to zero. The maximum difference in the elements is 8.95e-14. We also see that the least square error is 6.24e-24 with a hanning window and 1.12e-22 with random weights which is neglectable.

By doing this test we see that with a calibrated system it is possible to find the element weights.

## Element Phase

We are interested in determining if any of the elements or system channels has an extra delay compared to the others. In a narrow band system, a time delay can be approximated by a phase delay. In a broad band system using a short pulse this approximation is less valid. To determine the delay of an element we approximate this with the phase of the element. To do this we have to analyze the complex signals. This is done in the same matlab script KjorKode.m. In the previous simulations the parameters for the simulations is set in the matlab function Parametre.m. When operating with complex values the parameters are set in the matlab function ParametreKompll.m. When analyzing the phase the pulse has to be altered to get a good result. The reason for this is that phase is narrowband. In the previous simulations we have used a pulse with 3.5 periods. This pulse is too short and contains too many frequencies which make it broadband. With a short pulse we get problems with the response and the phase. An example, if we use the same parameters as the previous simulations. Pulse period set to 3.5 and speed of sound 1488 m/s. We put a small time delay on the third element which corresponds to a 45 degree phase shift of the center frequency. This should give us a phase change in the third element:

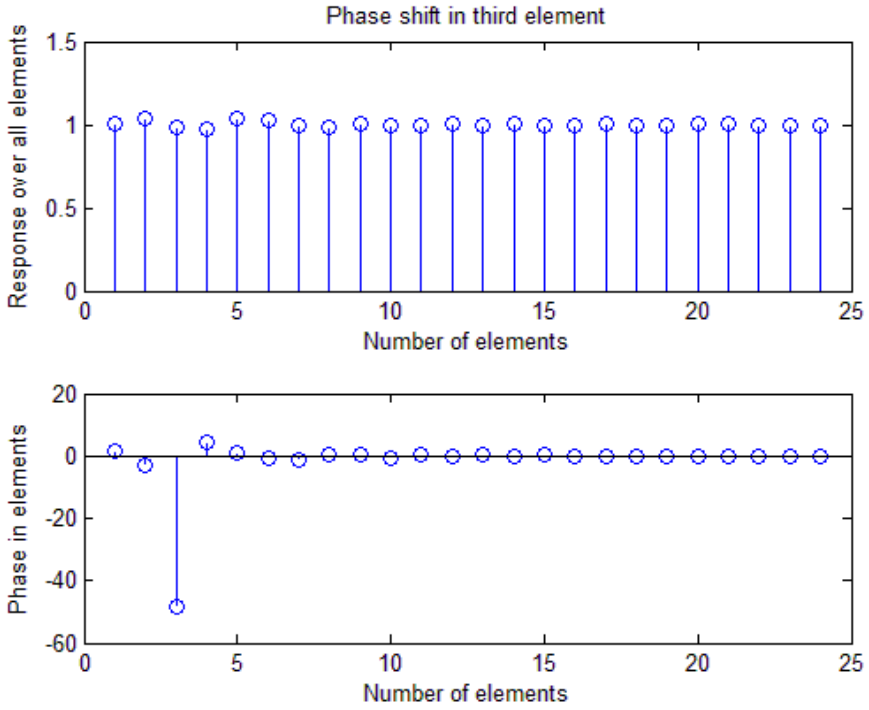


*Figure 5.18: Response over all elements with phase shift in third element and with a pulse period 3.5 periods.*

We see in figure 5.18 that by using a pulse period of 3.5 periods we get some fluctuations in the response and some phase changes in all elements. We get a least square error of 117.77 and in the third element we get a phase shift of -48 degrees. This is because of the short pulse and if we use a longer pulse with more periods we should get a better result. If we set the pulse period to 15 periods we get:

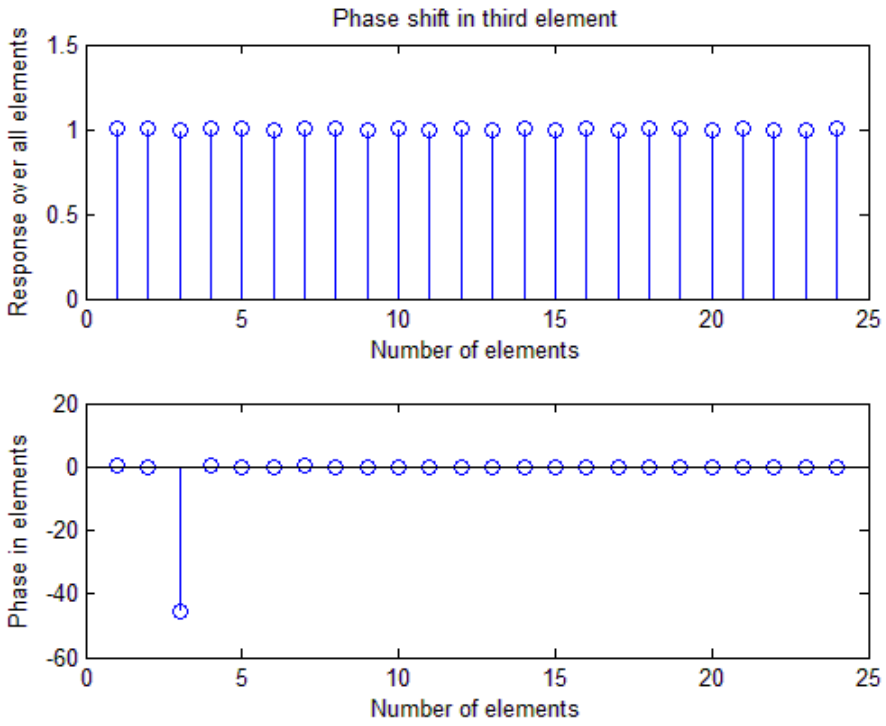


*Figure 5.19: Response over all elements with phase shift in third element and with a pulse period 15 periods.*

In Figure 5.19 we see that by using longer pulse we get a better result in the response and the phase. We get a least square error of 3.75 and a phase shift of -45.5 degrees in the third element. This is can be improved by using more periods in the pulse. We set the pulse period to 40 periods and get:

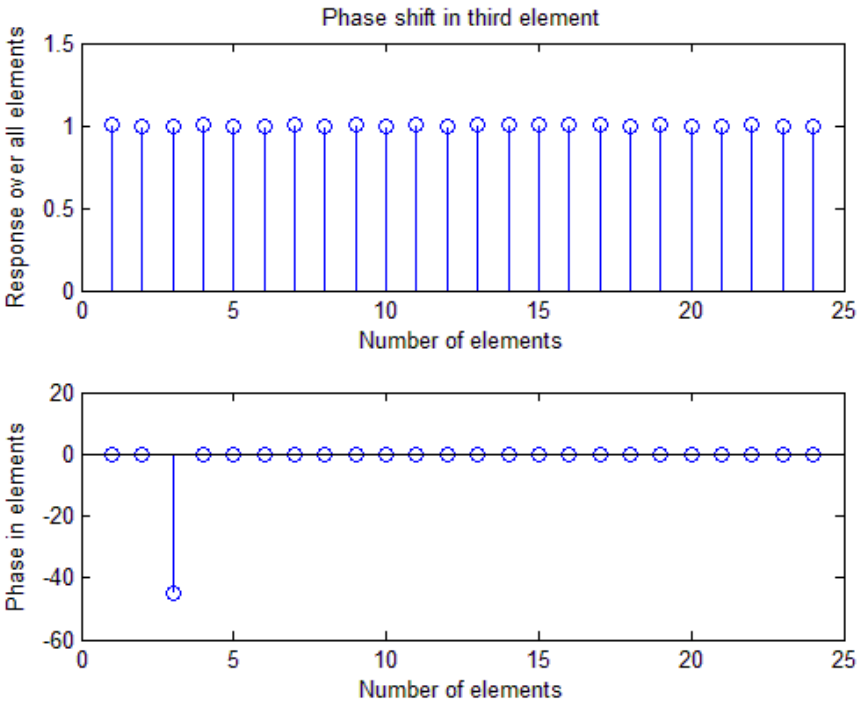


*Figure 5.20: Response over all elements with phase shift in third element and with a pulse period 40 periods.*

We see in Figure 5.20 that with 40 periods we get a better response over all elements. This gives us a least square error of 0.0024 and a phase shift in the third element of -45 degrees. 40 periods correspond to a pulse length more than 1 m. This is a very long pulse when we do measurements in a small water tank. We saw in Figure 5.19 that 15 periods should be sufficient. With these results we could start testing the system by putting a random phase on the elements. We have seen that the system works with phase shift in one element. How does the system preform if there is a random phase shift in all the elements?

.

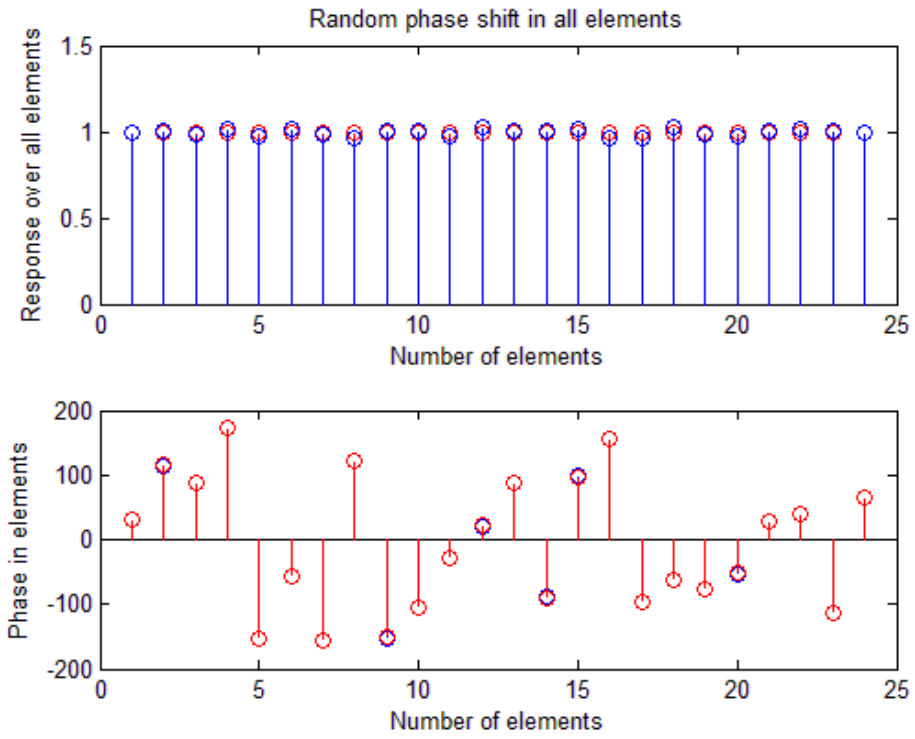


*Figure 5.21: Response over all elements with random phase shift in all elements and with a pulse period 15 periods.*

We see in Figure 5.21 that the phase is different in all elements. We see that there is some difference in phase and response in the elements but this is small. If we look at the maximum amplitude difference in the response it is 0.036 and the phase difference is 1.9 degrees. This is relatively small. We get a least square error of 1.58e+4. This is very high and can indicate that the least square error is not a god estimate when there is extra delay in the elements. In all the simulations we have used 120 measuring points. If we use the maximum amplitude difference and phase difference, can we find how many points is needed before the results starts degrade? This is important when trying to match the simulated data with the measured later in Chapter 5. In Table 5.1 we have tested different number of measuring points. We have put a hanning window on the elements and a random phase:

*Table 5.1: Number of measuring points needed for a good response*

| Number of measuring points | Max amplitude diff. | Max phase diff. |
|---|---|---|
| 120 | 0.036 | 1.9 degrees |
| 110 | 0.036 | 2.0 degrees |
| 100 | 0.034 | 1.8 degrees |
| 90 | 0.033 | 1.7 degrees |
| 80 | 0.058 | 2.8 degrees |
| 70 | 0.069 | 3.8 degrees |
| 60 | 0.039 | 1.8 degrees |
| 50 | 0.050 | 2.8 degrees |
| 40 | 0.038 | 2.0 degrees |
| 30 | 0.026 | 2.1 degrees |
| 20 | 0.178 | 12.5 degrees |
| 10 | 0.732 | 348.5 degrees |

We see in Table 5.1 that the difference in amplitude and phase compared to the window function and random phases is small for all measurements with more than 30 measuring points. It has some small variations in the measurements over 30 points but at 20 and 10 measuring points the max amplitude and phase the differences becomes big and thereby the response becomes more random. This means it is possible to do measurements with significantly lesser measuring points than 120. This becomes important when processing the measured data in Chapter 5.2.2 when trying to fit the measured data to the simulation program.

## 5.2 Experiment data

The results presented in this chapter are the result of the measurements done with 120 measuring points at a distance 1m and 0.5 m from the transducer. The measurements were done in the water tank at Ifi and controlled by the dynamic position system

### 5.2.1 Measured Sonar Field

The measurement data was first processed in the matlab script maling10_4.m. This was to see the dB plot of the sonar field.

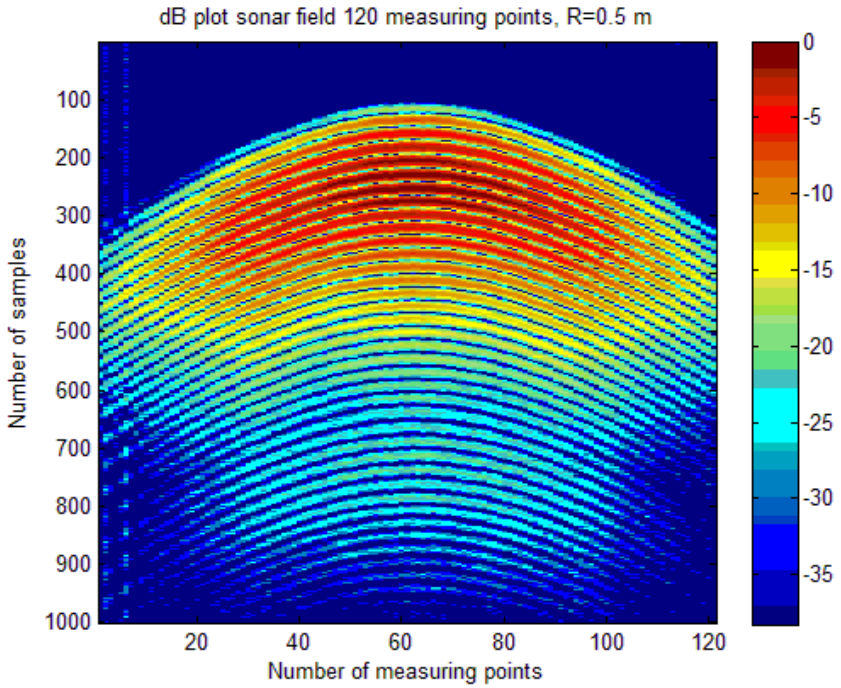


*Figure 5.22: dB plot of the sonar field at distance 0.5m with 121 measuring points.*

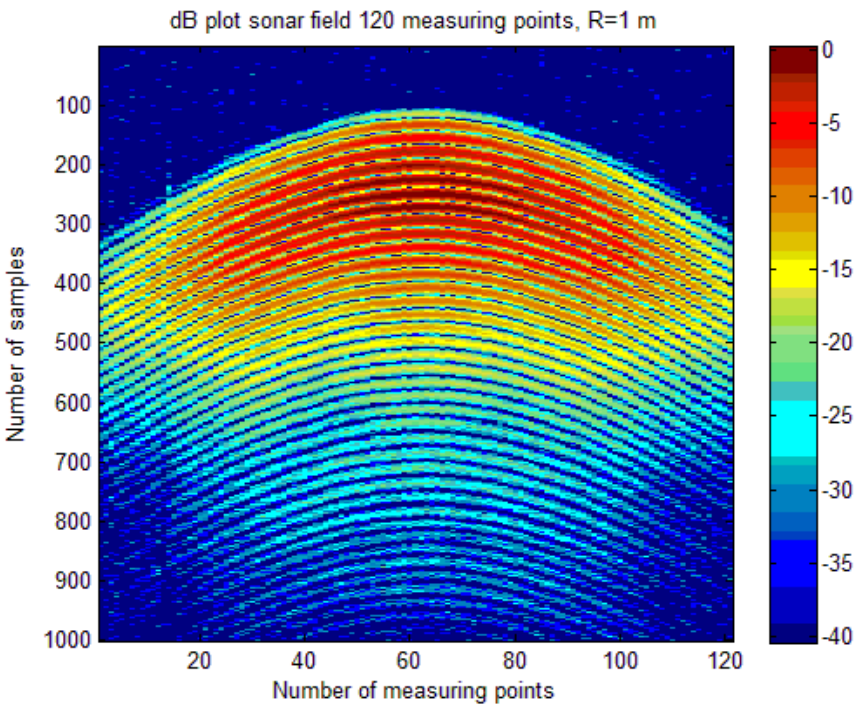


*Figure 5.23: dB plot of the sonar field at distance 0.5m with 121 measuring points.*

We see in Figure 5.22 and 5.23 that the resolution in x direction is good with 120 measuring points. There is some quantization noise in both measurements but much better than the measurements done with 180 measuring points that is presented in Chapter 4. There is more ringing in the received pulse in the 1 m measurement than in the 0.5 m measurement.

We compare the measured field with the simulated field in Field II. All the parameters in Field II is set to the same values as in the measuring experiment. The speed of sound is set to 1488 m/s, frequency set to 100 kHz, distance 0.5 m and 1m and the number of measuring points is set to 120. In the water tank we only used one element with 10 sub elements; in Field II we used the same. The element is placed in front of the array illustrated in Figure 5.24 and the simulation of the sonar fields is illustrated in Figure 5.25 and 5.26:

:

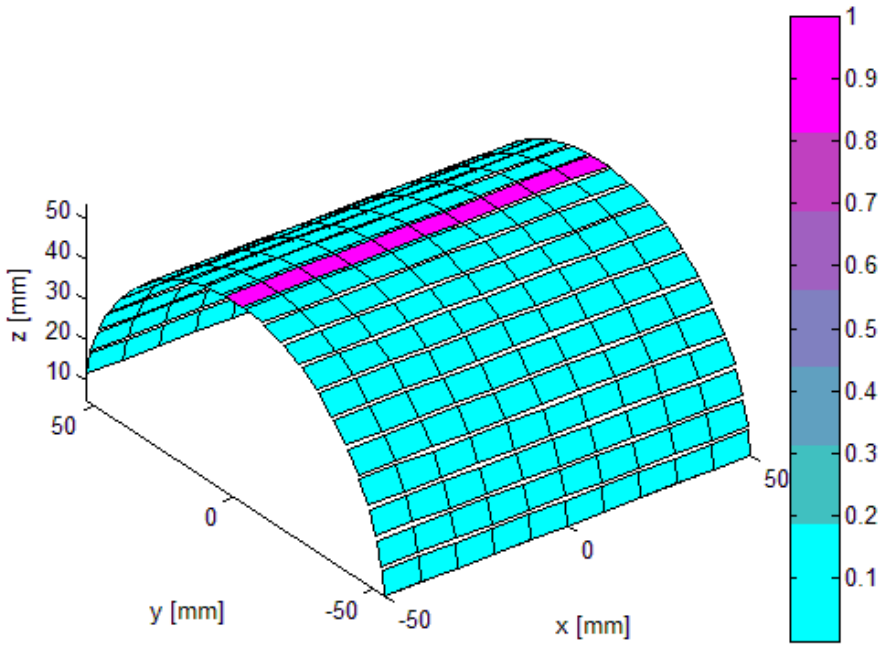


*Figure 5.24: Array with 24 elements and 240 sub elements in field II. The front element is active and sending.*

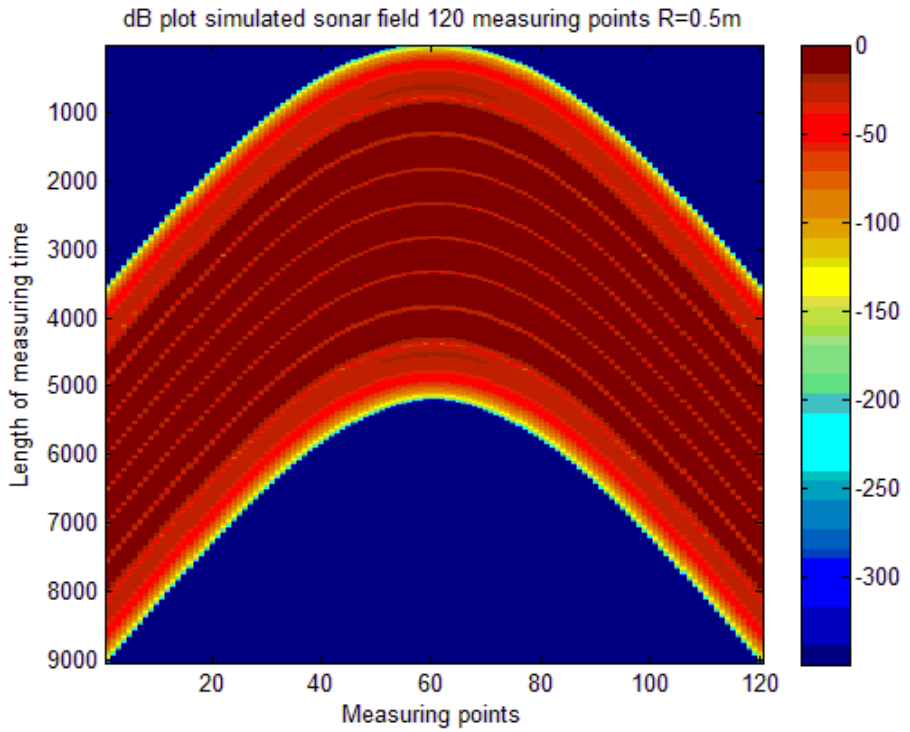


*Figure 5.25: dB plot of the simulated sonar field with 121 measuring points at a distance 0.5 m from the transducer.*

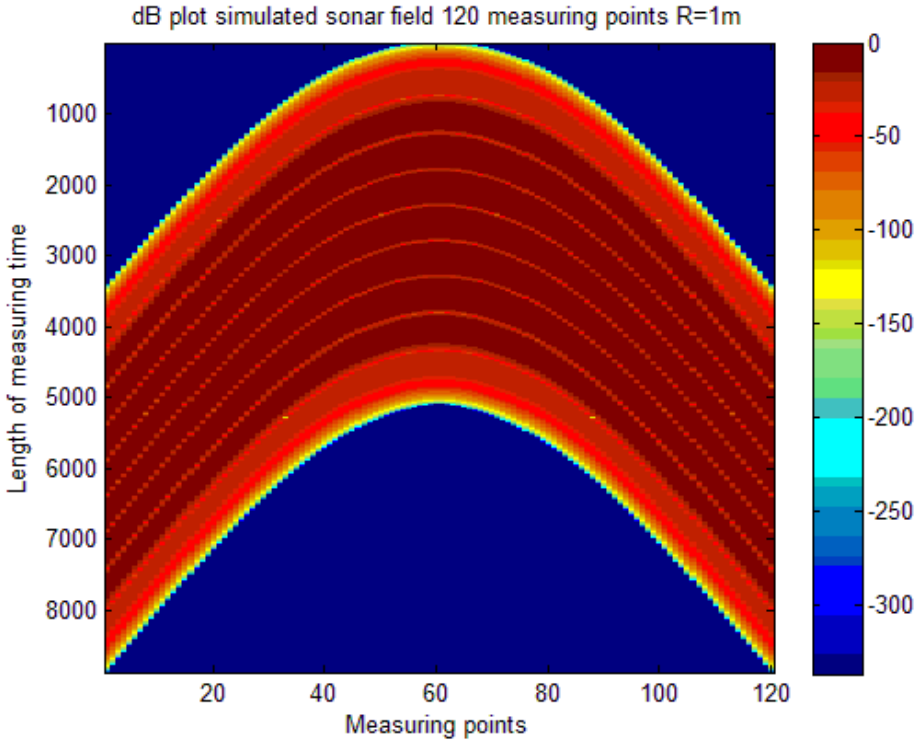


*Figure 5.26: dB plot of the simulated sonar field with 121measuring points at a distance 1 m from the transducer.*

If we compare the measured sonar field with the simulated we see in Figure 5.25 and 5.26 that they are fairly similar. There is more noise in the measured data but the simulation is the ideal world were noise don't exist. We see also that using one element produces similar results in the simulation and the measurement experiment. We can assume that using all elements in the SH-90 transducer will produce the same results as the simulations in Field II. This is not tested in this thesis since we only have a one-channel transmit-receive system and is only an assumption. After verifying the measurements with the simulations in Field II we can look at back propagating the measured data.

### 5.2.2 Processing of measured data

The processing of the measured data is done in the matlab script Testmaal.m. This program is based on the simulation program KjorKode.m. The idea is that by replacing the simulated field with the measured we could propagate the measured field back to the array. In the measurements the 121 measuring points were spread over a 180 degree sector covering the half circle around the transducer. We have seen in the simulations that 120 measuring points is more than we need and Table 5.1 show that everything over 30 measuring points is sufficient for a good result. In the simulations the measuring points and the array elements is placed in a sector of 120 degrees. This means we have to convert the measured data to match the sector of 120 degrees. The 121 measurements are done between -90 degrees and 90 degrees which gives us one measurement per 1.5 degrees. This corresponds to 81 measurements in a sector of 120 degrees. Because we want to use a 120 degree sector of the transducer, 17 elements in the array will give us an array equivalent to the SH-90 transducer which has 24 elements in the 180 degree radius. This was done with both measurements. After converting the measurement matrices the sonar field became:

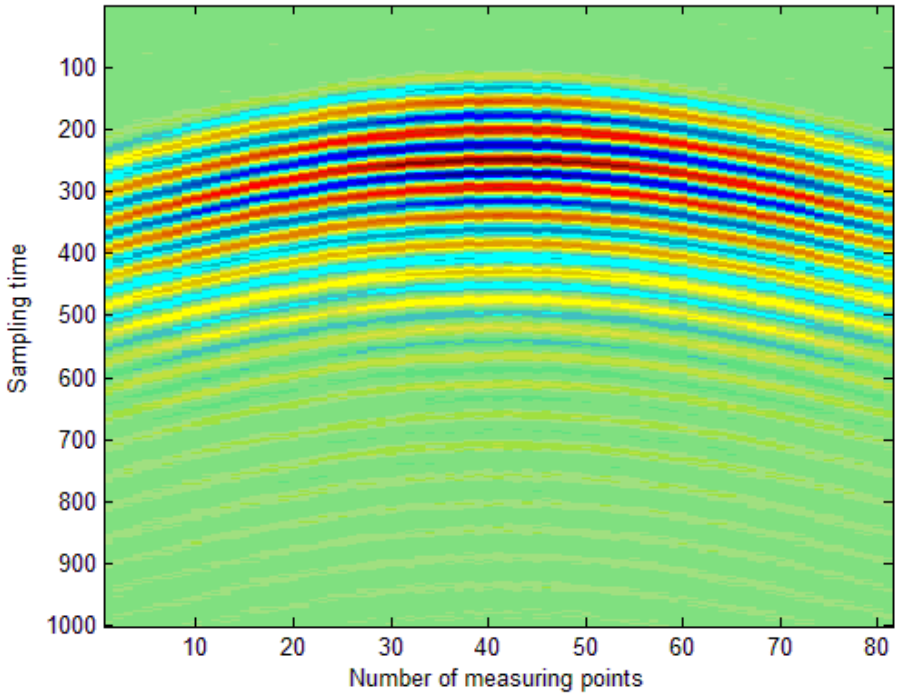


*Figure 5.27: The sonar field after converting it to 81 measuring points in a sector of 120 degree and 17 elements in the arrays. Distance is 1 m from the array.*

We see in Figure 5.27 that by converting the measurement matric we get a field of view of 120 degrees. To get the front view of the field we used the measuring points 21 to 101. With this data we could propagate the data back to the array to see the response over the elements. We have observed in the simulations that by knowing the speed of sound we get a response of one over all the elements. Since we don't know the speed of sound but have measured it, we can assume that the sound velocity is 1488 m/s and get a response of one over all elements. This can be illustrated by plotting the response for both measurements at distance 1 m and 0.5 m with assumed sound velocity 1488 m/s.

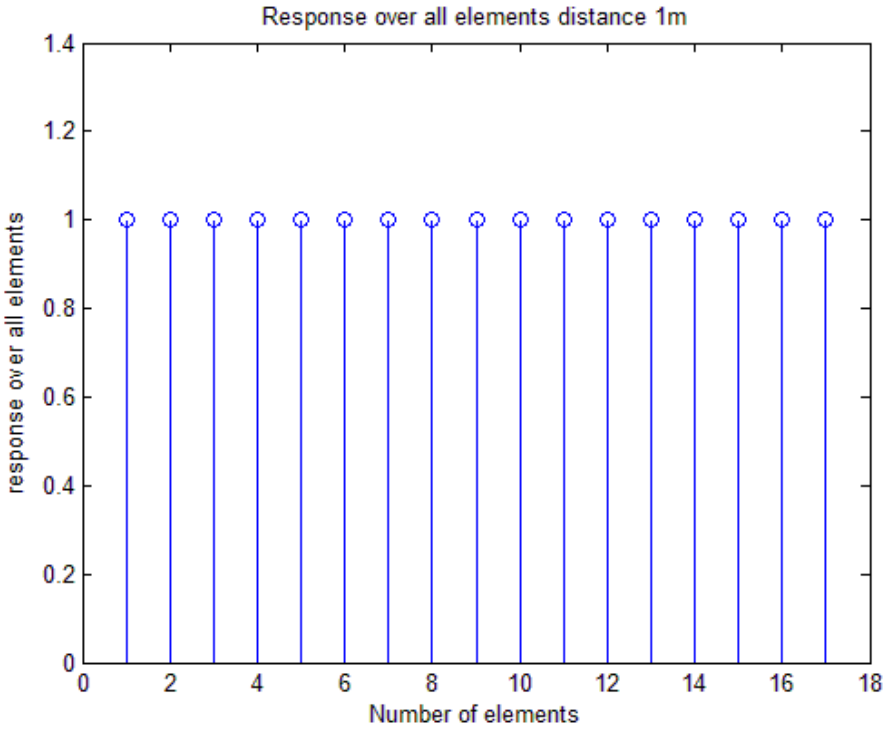


*Figure 5.28: The response over all elements with measured data at a distance 1 m from the transducer*

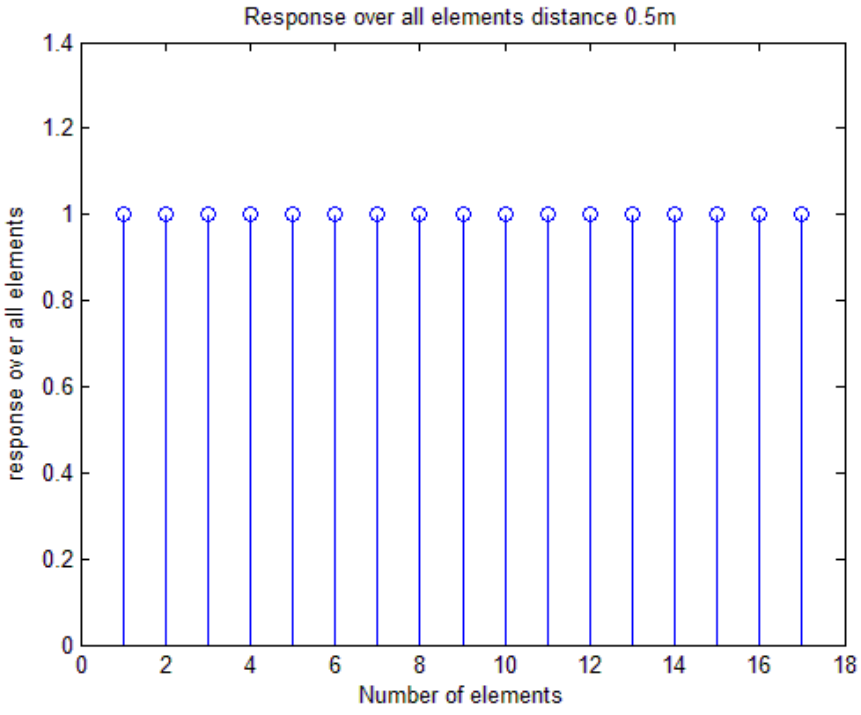


*Figure 5.29: The response over all elements with measured data at a distance 0.5m from the transducer*

If we compare the measured data with the simulated data we can look at the least square error. This will give us an overview of the difference of the measured and simulated data. We have seen in Figure 5.28 and 5.29 that the response over the elements with the measured data is one with the sound speed at 1488 m/s. If we use a simulated pulse with 3.5 periods we get the response:

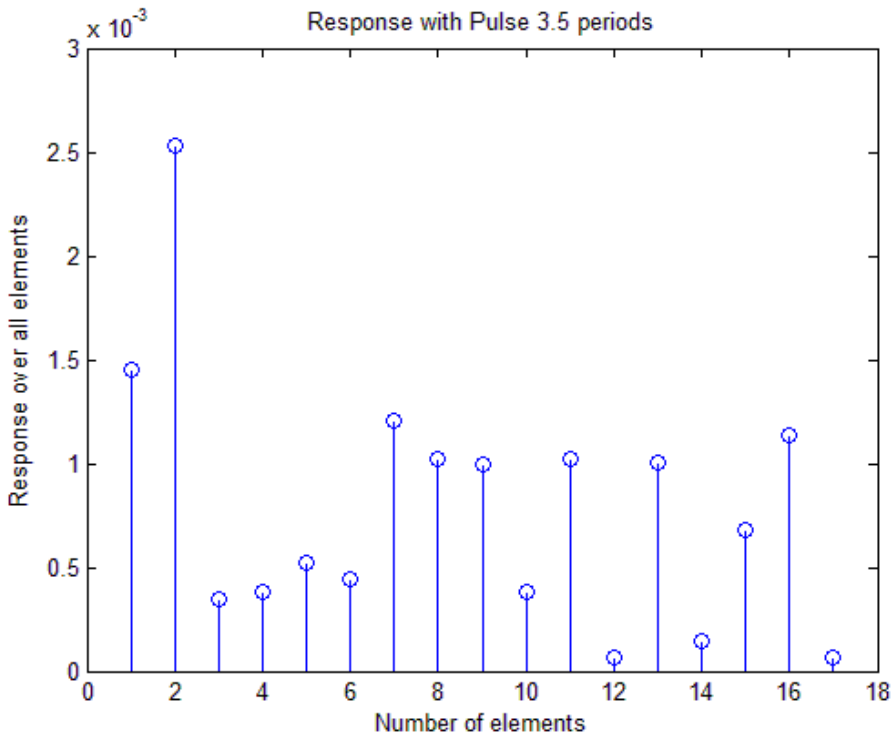


*Figure 5.30: Response over all elements with a pulse with 3.5 periods at distance 1 m*

We see in Figure 5.30 that the amplitude values are low, ten to the power of minus three. That is very low amplitude values that implicate that we have some problems with the parameters in the measured data. Since we only used 1 element in the measurements we know that the elements have the same phase. This should give us a low error but the least square error is 7.19e+3 which mean that the simulated and the measured data do not match up. Is this because of incorrect sound velocity? We test with 1487 m/s to see if there is a difference in the least square error. The least square error becomes 7.18e+3 which implicate that there is a trend of descending error with descending sound velocity. Can this be used to find the exact sound velocity? We test different sound speeds.

*Table 5.2: Least square error with different sound velocities*

| Speed of sound (m/s) | Least square error |
|---|---|
| 1486 | 7.178e+3 |
| 1485 | 7.170e+3 |
| 1484 | 7.164e+3 |
| 1483 | 7.157e+3 |
| 1482 | 7.150e+3 |
| 1481 | 7.144e+3 |
| 1480 | 7.136e+3 |
| 1479 | 7.131e+3 |
| 1478 | 7.123e+3 |
| 1477 | 7.117e+3 |
| 1476 | 7.109e+3 |
| 1475 | 7.103e+3 |

We see in Table 5.2 that the difference in error is very small. The smallest error we could obtain was when the speed of sound was at 1342 m/s where the least square error was at 12.67. This sound velocity is too slow. This indicates that there are more factors that have to be taken in to consideration than the sound velocity. This will be discussed in Chapter 5.3.

## 5.3 Summary and Discussion

In this part we sum up and discuss the results in the master thesis.

### 5.3.1 The simulations

As we have seen in the simulations, the system is sensitive to small changes in the sound speed. A difference of 1 m/s gives a least square error which suggests there is a large variation in the response. With differences of 2-5 m/s the response values drop and the least square error is significant. We also saw fluctuations in the outer elements when the sound speed was changed. These fluctuations appear to increase with more elements in the array. The least square error also seems to increase with the number of periods in the pulse. Some improvements can be done by changing the pulse periods. The fluctuations seemed to decrease with fewer periods. The least square error was still high, ten to the power of four, but that was the best result. This suggests that the system is sensitive to changes in sound speed, number of elements and pulse form.

Does this give us a working system? We have seen in the simulations that this works if we know the pulse length and the response to the elements. The least square error can be used to calibrate the system if the only unknown parameter is the speed of sound. By using the systems sensitivity to sound velocity it is possible to find the true sound speed. The correct sound velocity is when the least square error is closest to zero.

How can we use this? When we have a calibrated system and know the speed of sound we can characterize an unknown transducer. This is important if you want to calculate the transducers far-field response. To calculate the transducers far-field response you have to determine the element weights and the elements phase. This was done by implementing different weights on the elements and trying out random phases. By comparing the responses with the window functions we could see if the amplitude values changed. High max amplitude difference between the weighted signal and the window function indicated wrong weighing on the elements. This worked when the sound velocity was known but with a small change in the sound speed the amplitude difference became higher. We also saw that the system worked with different phase in one element as long as we used longer pulse. With a random phase in each element we saw that least square was not a god estimate on the error but the response was still one over all elements. By looking at the maximum amplitude difference between the window function and the response, and the difference in the random phase it was possible to find out how many measuring points that are needed to give a god response. With a hanning weight and random phase in the elements we could try out different numbers of measuring

points. We found that with less than 30 measuring points the amplitudes in the response started to vary. This means that in a measuring experiment it is possible to use 30 measuring points and still get a good response.

With these simulations it is possible to calculate the transducers far-field response. The simulations showed us that with a calibrated system it is possible to characterize a transducer in the near-field. The important parameters to know are the weights and the phase of the elements. To get these we have to have a calibrated system which means we have to know the speed of sound. If all this is known it is possible to calculate the far-field response.

### 5.3.2 The experiment

In the water tank we measured the temperature in the water. By calculating the time period between two measurements and using *Lubbers and Graaff's equation (equation 4.5)* we found that the speed of sound in the water was 1488 m/s. If we look in table 4.1, we see that this speed corresponds to a temperature between 21 and 22 degrees Celsius. Since the measured temperature in the water tank was 20 degrees Celsius the speed could be somewhere between 1482.3 m/s and 1485.4 m/s. This is a difference of 6-3 m/s from the measured speed and there could be a number of sources for this error. The most important is probably that the measured temperature is not exact. The measurement was done with a thermometer and measuring the exact temperature is difficult. The temperature may differ +/- 1 degree or more. As we see in table 4.1, +/- 1 degree makes a significant difference in sound speed. The simulations showed us that the system was sensitive to small changes in the sound speed and when simulating we used the measured speed as the true sound velocity. With the simulations we had found a way to characterize a transducer. This helped us find the parameters for back propagating the measured data to a transducer similar to the SH-90 transducer. The simulation programs were made so they would match the conditions in the water tank, so with a characterized transducer we could back propagate the measured data with a program based on the simulation programs.

The back propagating of measured data proved to be a little difficult. Implementing the measured data into the simulation system was done by replacing the simulated pulses with the measured pulses and back propagating this to the array. Since the simulation programs were made so they would match the conditions in the water tank, implementing these data was not a problem. Since we only used one element in the measurements all the elements should have the same phase and amplitude. The problems came when comparing the measured to the

simulated pulses. Ideally when using the same sound speed those two results would match up and the least square error would be close to zero. If not we could use the least square error to find the correct sound velocity. This proved to be difficult since the amplitude values in the simulated response were so low compared to the measured and the least square error became high. By trying out different sound speeds we saw small improvements in the error with descending sound speed but the improvements were so small that there were more factors contributing to the errors.

So the big question is what are the contributing factors to these errors? When reanalyzing the system we saw that there were some new unknowns that contribute to the problem. If we have a calibrated system the pulses from the measured data and the simulated data would match up in the measuring points. For this to happen we have to know four main parameters, the distance, the speed of sound, the time of arrival and the pulse shape. We have seen in Chapter 4, equation 4.6 that the distance is dependent on the sound velocity and the time. In the simulations there is an ideal situation where these parameters can be set and found accurately. The problem with the measurement data is that in the measurement experiment these parameters had an approximate value. The reason for this is that we get an extra time delay in the measuring equipment. When sending from the pulse generator trough the oscilloscope in to the transducer we get a time delay in the system called instrumental delay, a $\Delta t$. That means to find the exact speed of sound we have to take into account that we have a time which is $t+\Delta t$. This means that if we don't know $\Delta t$ we can only get an approximate distance, speed of sound and time. This gives us a mismatch in the time of arrival in the measuring points between the simulated and the measured data. If this mismatch is greater than $\lambda/8$ we get a problem with the response.

The pulse shape can also be a contributing factor to the mismatch. In the simulations we send a pulse with 3.5 periods and measuring points measure a pulse with 3.5 periods. Since the transducer is a physical object sending out a pulse the elements will use some time to start and end the oscillations. This means that sending a pulse with 3.5 periods will give a longer pulse with ringing noise in the measuring points. Matching the two pulse shapes is something to take into account and can be difficult but an approximate matching of the pulse length will improve the result. But without knowing $\Delta t$ the matching of pulse shapes will not have a significant impact on the response. From these disheartening discoveries, we concluded that we could not use the measured data to verify the method. Since collecting new data would not be possible due to the MSc project limited time, we chose not to pursue this.

# Chapter 6

## 6. Conclusion

The work in this thesis can be divided into two main parts, the measurement experiment and the simulations. Like stated in the objectives in Chapter 1, preparing the SH-90 transducer, making the necessary hardware and learning the dynamic position system was important tasks to make the necessary measurements in the water tank. This was work that had never been done at the sonar lab at Ifi so a lot of testing was necessary in order to get these measurements. This had an impact on the final result.

As we have seen in this thesis it was difficult to get the measured data to match up with the simulated data. The simulations showed that characterization of a transducer in the near-field by finding the right element weights and element phase is possible. The system was sensitive to changes in the sound velocity. We saw drops in the response and an increase of the least square error with small changes in the sound velocity. When the sound velocity is known it was possible to accurately estimate the weights of the elements and the phase characteristics. With the least square error we saw that we could find the right sound speed with the least square error closest to zero. This showed us that in theory it is possible to characterize a transducer in the near-field and calculate the far-field response. The problem with the measured data was probably the time aspect. When comparing it with the simulated pulses we saw a significant drop in the amplitude values in the response. The results didn't improve noticeably by changing the sound velocity.

There can be a number of reasons for the poor response. The main reason might be the delay in the measuring system called the instrumental delay which gives us a $\Delta t$ on the arrival time at the measuring points. To get a good response over all elements it is important that the timing of the pulses is right in the measuring points. A mismatch in the arrival time gives us a poor result. The pulse shape is also an important aspect. A mismatch in the pulse shape will contribute to the errors in the response. By matching the periods in the measured and the simulated pulses it should be possible to minimize the errors. It is also important to emphasize that this was the first time a measurement was done on this system. A lot of the time went in to getting the system operational. Waiting for the right temperature in the water, air bubbles on the transducer surface, leakages in the water tank and optimization of the dynamic position

system made the measurements in the water tank a time consuming task. A master thesis is a time limited project so to get a fully calibrated system more work is needed in the water tank.

But despite this the conclusion at the end of the project is that this method could work. The simulations have showed that in theory we can characterize a transducer in the near-field and by that calculate the far-field response. With more work on the measuring experiment we believe it is possible to match the measured data to the simulated data. By compensating for the instrumental delay it should be possible to get the timing right in the measuring points and improve the element response. An answer to the underlying question whether it is possible to say something about the sonars far-field by measuring the near field depends on the calibration of the system. With a good calibrated measuring system it is possible to characterize the transducer with measurements in the near field.

# Chapter 7

## 7. Further work

The propositions for further work could be several but only the ones considered interesting is mentioned. The first that comes to mind is trying to improve the results done in this work. In the measurement experiment it is possible to optimize the calibration of the system as stated in Chapter 5. To find the instrument delay would be a first priority and try to match the simulated and measured data better. This is a problem that can be investigated more and hopefully gives better results. Other things that would be natural to examine are:

- The element directivity, will it have an impact on the results.
- We saw in the simulations that 30 evenly spaced measuring points was sufficient for a good result, what about uneven placement of the measuring points. Is it possible to further reduce the number of measuring points?
- Measure at multiple depths and combine data. By doing this we get more data and possibly a better result.
- Look at the reflection effects in the tank. Since the tank is small they will come from different angels at various depths.
- Do the measurements with a 2D array. The SH-90 transducer is a 2D cylindrical array. We used the SH-90 transducer as a 1D array. An investigation of what is needed to characterize a 2D array is a natural extension of this work.

# Appendix A

This appendix contains the matlab scripts used in this thesis. Several scripts were made during the project but only the most important programs to recreate the simulation results and the processing of the measured data is included in this chapter. The scripts that control the dynamic position system is not included and are only available for those doing research at the sonar lab at Ifi.

## A1. Linear array

Contains the scrips ForoverPropagering.m and BakoverPropagering.m

### ForoverPropagering.m

```
%%Simulerer bølgefelt med et lieært array
clear all
format long

M=24;                    %antall elementer
R0=1000;                 %avstand til målepunkter
numPoints=121;           %antall målepunkter
t = [ 0 : 1 : 450 ];%Time Samples Puls
f =100e3;               %Input Signal Frekvens KHz Puls
fs =100*f;             %Sampling frekvens MHz Puls
Length=length(t);               %Lengde hammingvindu
%
P.fs=100e3;
P.c=1488;
P.dato=today;
P.focus= [0 R0 0]';


%% Linært array med 10 elementer med avstand lambda/2 mellom hvert element
P.M = M;
lambda=P.c/P.fs;
d= lambda/2;             %Element avstand
%d= lambda;             %Element avstand
L=(M-1)*d;               %lengde array
x=[0:d:L]-L/2;           %Element posisjon
y=zeros(1,length(x));    %Element posisjon


    P.elpos=zeros(3,M);
    P.elpos(1,:) = x;
    P.elpos(2,:) = y;
    P.elpos(3,:) = 0;


%legger på elementvekter på elementene
elVekt = 0.5 + .5*hamming(M);

 for i=1:M
 elVekt(:,10) = 0;
 elVekt(1)=1;
 end
%

if 0
    figure(100); clf
plot(X,Y,'o');
axis([-0.6 0.6,0 0.5]);
grid on
hold on
end
%% Målepunkter
z=0;
```

```
Th=linspace(0,pi,numPoints);
Th=linspace(pi/6,pi-pi/6,numPoints);
%Th=linspace(-pi/2,2*pi-pi/2,numPoints);
%Th = ([1:numPoints]-1)/numPoints*pi;
%Th=([1:numPoints]-1)/numPoints*2*pi;
rho=ones(1,numPoints)*R0;
[X,Y] = pol2cart(Th,rho,z);
P.malepos=zeros(3,numPoints);
P.malepos(1,:) = X;
P.malepos(2,:) = Y;
P.malepos(3,:) = 0;
P.numPoints = numPoints;

figure(1);clf
%subplot(1,2,1);
plot(1e2*P.elpos(1,:),1e2*P.elpos(2,:),'o');
hold on
plot(1e2*X,1e2*Y,'o','linewidth',2);
hold on
plot(1e2*P.focus(1),1e2*P.focus(2),'r+');
hold off
xlabel(['R0=',num2str(R0*1e2,'%.1f'),' cm, array in center origo']);
ylabel(['R0=',num2str(R0*1e2,'%.1f'),' cm']);
title('Linear array with 10 elements, 121 points')
axis equal
grid on

x1 =sin(2*pi*f/fs*t);          %Genererer Sinus Wave
vindu=hanning(Length);         %Hammingvindu
V=vindu';
%V=1;
Puls=V.*x1;                    %Sinusbølge med hammingvindu
figure(2);
plot(Puls);
grid on
hold on
%Puls=S(:,60);
%% Målesekvens

el_focus_norm =  sqrt(sum(bsxfun(@minus,P.elpos, P.focus)'.^2,2));
[dmin] = min(el_focus_norm);

tm_f = (el_focus_norm - dmin)./P.c;

%Array for time delays
tmn = zeros(P.M, P.numPoints);

for n = 1:P.numPoints

   el_male_norm = sqrt(sum(bsxfun(@minus, P.elpos, P.malepos(:,n))'.^2,2));
   tmn_ra = el_male_norm./P.c;

   tmn(:,n) = tmn_ra - tm_f;
end

max_delay = max(max(tmn));
min_delay = min(min(tmn));

%Array for output
left_margin = 2000; right_margin = left_margin;
window_size = round( (max_delay-min_delay)*fs + Length + left_margin + right_margin);
Resp = zeros(P.numPoints, window_size);
%%
for n = 1: P.numPoints
   for m = 1: P.M

       from = left_margin + round((tmn(m,n)-min_delay)*fs);
       to = from + Length-1;

        ra = sqrt(sum(bsxfun(@minus, P.elpos(:,m), P.malepos(:,n))'.^2,2)); %reise avstand
       Resp(n,from:to) = Resp(n,from:to) + Puls;%*1./ra./ra;

   end
end
%%
figure(3)
imagesc(db(Resp'),[-25,max(max(db(Resp)))]);
```

```
colorbar;

xlabel('Measuring points');
ylabel('Length of measuring time');

%% ------------------------------------------------

% Lagre resultater
P.Resp = Resp';
P.tidMale = [round(min_delay*fs): round(min_delay*fs) + window_size-1 ] ;
P.Puls = Puls;
P.tidPuls = P.tidMale;


figure(4)
kx=linspace(-2*pi/lambda,2*pi/lambda,numPoints);
phi=radtodeg(kx/(90*pi));
plot(phi,db(max(Resp')));
axis([phi(1),phi(length(phi)),-10,30])
xlabel('Phi(degrees)');
ylabel('Response [dB]');
grid on;

figure(5)
imagesc(db(Resp'));
colorbar;
xlabel('Measuring points');
ylabel('Sampling time');
```

## BakoverPropagering.m

```
%Array for time delays
tmn = zeros(P.M, P.numPoints);

for n = 1:P.numPoints

   el_male_norm = sqrt(sum(bsxfun(@minus, P.elpos, P.malepos(:,n))'.^2,2));
   tmn_ra = el_male_norm./P.c;

   tmn(:,n) = tmn_ra;


end
%

%max_delay = max(max(tmn));
%min_delay = min(min(tmn));

%Array for output
Resp_tilbake = ones(M,window_size*2);


for m = 1:P.M
    for n = 1:P.numPoints

    from = round(min_delay*fs - tmn(m,n)*fs)+4000+1;
    to = from + window_size-1;

    ra = sqrt(sum(bsxfun(@minus, P.elpos(:,m), P.malepos(:,n))'.^2,2)); %reise avstand

    Resp_tilbake(m,from:to) = Resp_tilbake(m,from:to)+ Resp(n,:);%*ra;

    end
end


figure(4)
stem((max(Resp_tilbake')));
xlabel('Number of elements');
ylabel('Amplitude');

% plot(db(max(Resp)))
```

## A2. Field II script

Contains the script LagSimulering.m. This script needs the Field II program to run.

### LagSimulering.m

```
addpath FieldII

%
%
% Generate the transducer apertures for send and receive
f0=100e3;
fs=100e6;
c=1488;

lambda=c/f0;


kerf_h = 0.001%(0.47-0.7)*lambda; % avstand mellom elementer, gir 0.7 %lambda bredde
kerf_w = 0.001;

width = 0.1; % dvs 10 cm
height = 0.47*lambda-kerf_h; % 48 elementer gir full sylinder

no_sub_x =10; % 10 cm er 6.6 lambda. Må ha sub-biter mindre enn lambda/2
no_sub_y =1; %2*2;



focus=[0 0 0.5];  %focus point

%Radius = 3;
Radius = 0.1078/2;
%a_cnf= 1;
a_cnf =  ones(1,23);
if 0
    a_cnf(:) = 1e-7;
    a_cnf(12) = 1;
    a_cnf(23) =0; 1e-7;
end

field_init(0);

Th = xdc_curved_linear_array(width, height, kerf_w, kerf_h, ...
    no_sub_x, no_sub_y, focus, Radius, a_cnf);
figure(1);
show_xdc(Th);
hold on

% Set the exitation
excitation=1e6*sin(2*pi*f0*(0:1/fs:3/f0));
xdc_excitation (Th, excitation);
%
% Set the impulse response for the  aperture
impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(impulse_response)))';

xdc_impulse (Th, impulse_response);

% Which points to observe the pulse:
%
Rad = 0.5;  % radius in meter.
Theta = deg2rad(-35:7:35); % Degrees
Theta = deg2rad(linspace(-90,90,120)); % Degrees
[z,y] = pol2cart(Theta,Rad);

xpoints = [-60:5:70].'*1e-3;
xpoints = 0;
points = zeros(length(Theta)*length(xpoints),3);
```

```
for ii=1:length(y)
    points(((ii-1)*length(xpoints)+(1:length(xpoints))),1) = xpoints;
    points(((ii-1)*length(xpoints)+(1:length(xpoints))),2) = y(ii);
    points(((ii-1)*length(xpoints)+(1:length(xpoints))),3) = z(ii);
end
%
%points = [0 0 1.5]

disp(points)
%points = points(10,:)

% Calculate the received response
[hp, start_time] = calc_hp(Th, points);


 % Here the display of the data is inserted
%
figure(2);
plot3(points(:,1),points(:,2), points(:,3),'*')

figure(3);
plot(hp)

figure(4);
imagesc(db(hp/max(hp(:)))); colorbar
xlabel('Measuring points');
ylabel('Length of measuring time');
title('dB plot simulated sonar field 120 measuring points R=0.5m');

figure(5);
plot(rad2deg(Theta),db(max(hp)))
xlabel('Measuring points');
ylabel('Response in dB');

figure(6);
imagesc(hp);
colorbar;
xlabel('Measuring points');
ylabel('Length of measuring time');

figure(7)
mesh(db(hp));
```

## A3. Curved array

Contains the main script KjorKode.m and the functions Parametre.m,ParametreKompl1.m and CalcField.m.

### KjorKode.m

```
% Script som kjører kode ved hjelp av funksjonene CalcField og
%Parametre og Parametrekompl1. Simmulerer sonar felt med et krumt array
%Siste versjon er endret 05.09.2014

% 1: Definerer P (Kan kalle Parametre.m)
P = Parametre;

% 2: Beregner feltet som måles i målepunkter(Kaller på CalcField.m):
B = CalcField(P);

% 3A: Finner et eksempel på målt signal, gitt parametre i P:

BB = reshape(B.Pulser,length(P.MalePkt.tMalePkt)*P.MalePkt.numPoints,P.M);

%
Vekt1 =ones(P.M,1);
MaltSignal =  BB* Vekt1;

% LS-løsning
```

```
disp('Ideell verden, rett lydhastighet')
x1 = BB\MaltSignal

% LS-Feil:
sum((MaltSignal - BB*x1).^2)


%%
% 3B: Forandrer lydhastighet og lager annet målt signal:
Pny = P;
Pny.c = 1486;

B2 = CalcField(Pny);

%


BBny = reshape(B2.Pulser,length(P.MalePkt.tMalePkt)*Pny.MalePkt.numPoints,Pny.M);

%Legger på random vekter
Window=ones(Pny.M,1)+0.1*randn(Pny.M,1);
Vekt2 =ones(Pny.M,1)+0.1*randn(Pny.M,1);
MaltSignal2 =  BBny* Vekt2;

% LS-løsning:Tester målematrise BB (beregnet med P.c = 1500) med
% målte data gitt at lydhastigheten i vannet var 1488m/s:
disp('Ideell verden, feil lydhastighet')
x2 = BB\MaltSignal2

% LS-Feil:
sum((MaltSignal2 - BB*x2).^2)
%%
% 4A: Lager kompleks signal. I tillegg langt signal ...
% 1: Definerer P (Kan kalle ParametreKompl1.m)
Pkomp = ParametreKompl1;

% 2: Beregner feltet som måles i målepunkter:
BBkomp = CalcField(Pkomp);

% 3A: Finner et eksempel på målt signal, gitt parametre i P:

BBkomp =
reshape(BBkomp.Pulser,length(Pkomp.MalePkt.tMalePkt)*Pkomp.MalePkt.numPoints,Pkomp.M);

% Lager vekt hvor et element er kompleks ...
Vekt = ones(Pkomp.M,1);
Vekt(3) = cosd(45)+i*sind(45);
MaltSignalkomp =  BBkomp * Vekt;

% LS-løsning
disp('Ideell verden, rett lydhastighet')
x3 = BBkomp\MaltSignalkomp
[abs(x3) angle(x3)/2/pi*360]

% LS-Feil:
sum(abs(MaltSignalkomp - BBkomp*x3).^2)

% 4B: Lager hack, legger til litt tid til en av elementene
PkompEkstra = Pkomp;
PkompEkstra.EkstraDelay = zeros(PkompEkstra.M,1);
%Legger til random fase i alle elementer
PkompEkstra.EkstraDelay(:) =(360*rand(PkompEkstra.M,1 )-180)/360*PkompEkstra.fs /
PkompEkstra.f;
%PkompEkstra.EkstraDelay([3]) =(45)/360*PkompEkstra.fs / PkompEkstra.f;

BkompEkstra = CalcField(PkompEkstra);

% 3A: Finner et eksempel på målt signal, gitt parametre i P:

BBkompEkstra =
reshape(BkompEkstra.Pulser,length(PkompEkstra.MalePkt.tMalePkt)*PkompEkstra.MalePkt.numPoints,
PkompEkstra.M);

% Vekt =hann(PkompEkstra.M); %ones(PkompEkstra.M,1);%+ 0.1*randn(PkompEkstra.M,1);
Vekt =ones(PkompEkstra.M,1);%+ 0.1*randn(PkompEkstra.M,1);
MaltSignalkompEkstra =  BBkompEkstra * Vekt;
```

```
% LS-løsning: NB: Tester målematrise BB (beregnet med P.c = 1500) med
% målte data gitt at lydhastigheten i vannet var 1488M/s:
disp('Ideell verden, rett lydhastighet')
x = BBkomp\MaltSignalkompEkstra
[abs(x) angle(x)/2/pi*360]

% LS-Feil:
sum(abs(MaltSignalkompEkstra - BBkomp*x).^2)

disp('Maks feil i amplitude og fase:')
[max(abs(Vekt-abs(x))) max(abs(angle(x)/2/pi*360 - (-
PkompEkstra.EkstraDelay*PkompEkstra.f/PkompEkstra.fs)*360))]
```

## Calcfield.m

```
function B = CalcField(P);
%
% Function that calculates the field measured in a set of measuring points,
% P.MalePkt, when a signal P.Pulse are sent from a set of transmit points,
% P.elpos.
%
% Input: Struct P:
%
% MalePkt.numPoints : No. of points to measure the response
%     MalePkt.Coord : [3xP.MalePkt.numPoints], array of (x,y,z)
%                      coordinates [meter]
%                 f : Center frequency [Hz]
%                fs : Sampling frequency [Hz]
%                 c : Speed-of-sound [m/s]
%             Focus : Coordinates for focus point [3x1] [meter]
%             Pulse : Pulse sent from each element
%            tPulse : Time axis for P.Pulse [1xlength(P.Pulse)]
%                 M : No of transmit elements
%             elpos : Coordinates of Tx element positions [3xP.M] [meter]
%          elCenter : Coordinate of element center
%                     (from which the focus is calculated from) [3x1] [meter]
%          tMalePkt : [1x1001 double]
%
%       [optional]
%       EkstraDelay : Add extra constant delay for one or more channels [1xP.M]
%
% Output: Struct B:
%
%
%     AA, 18/6-2014 :    First version
%     AA, 19/6-2014 :    Added "P.EkstraDelay"



% Allocate memory
B.Pulser = zeros(length(P.MalePkt.tMalePkt),P.MalePkt.numPoints,P.M);

% For all measuring points, calculate the contribution from each Tx element, 1..P.M

% figure(1); clf; grid on; hold all
for i_m = 1:P.MalePkt.numPoints  % for-løkke over målepunkter
    % Koordinat dette målepunkt:
    for i_el = 1:P.M  % for-løkke over elementer
%
        % Tidspkt element starter å sende, målt i sampler
        tauEl = P.fs/P.c*(norm(P.Focus- P.elCenter) - norm(P.Focus - P.elpos(:,i_el)));

        if isfield(P,'EkstraDelay') & P.EkstraDelay(i_el) ~=0
            tauEl = tauEl + P.EkstraDelay(i_el);
        end
        % Når når pulsen fra dette elementet fram til dette målepkt
        narFram = P.fs/P.c*(norm(P.MalePkt.Coord(:,i_m) - P.elpos(:,i_el)) + norm(P.elCenter))
+ tauEl;

        % Dette elementets bidrag til puls i dette målepunkt
        Bidrag = interp1(P.tPulse + narFram,P.Pulse,P.MalePkt.tMalePkt,'linear',0).';
        %Bidrag = interp1( P.Pulse,P.MalePkt.tMalePkt,'linear',0).';

        %plot(P.tMalePkt,2*i_m + Bidrag)


        B.Pulser(:,i_m,i_el) = Bidrag;
```

```
        %pause
        % keyboar
    end
end
```

## Parametre.m

```
function P = Parametre

% Function that sets up a struct of parameters needed for CalcField()
P.verbose = 1;  %

P.f = 100e3;   % Input signal freq. in Hz
P.fs = 100*P.f;   % Samplingsfrekvens
P.c = 1488;
% Definerer målepunkter:
P.MalePkt.R0 = 1e0;  % Avstand til målepunkt målt i meter
if 0,
    P.MalePkt.numPoints = 2; % Antall målepunkter
    P.MalePkt.Theta = deg2rad([0 -45]);
else
    P.MalePkt.numPoints = 81; % Antall målepunkter
    P.MalePkt.Theta = deg2rad(linspace(-60,60,P.MalePkt.numPoints));
    %P.MalePkt.Theta = deg2rad(linspace(0,358,P.MalePkt.numPoints));
    %P.MalePkt.Theta = 2*pi*[0:P.MalePkt.numPoints-1]/P.MalePkt.numPoints;
    %P.MalePkt.Theta = linspace(-pi/2,pi/2,P.MalePkt.numPoints);
end
P.MalePkt.Coord = zeros(3,P.MalePkt.numPoints);
for i_m=1:P.MalePkt.numPoints
    [y,x,z] = pol2cart(P.MalePkt.Theta(i_m),P.MalePkt.R0,0);
    P.MalePkt.Coord(:,i_m) = [x y z].';
end


% Definerer måletidspunkter:
antMalePer = 1*10;
P.MalePkt.tMalePkt = (-P.fs/P.f*(antMalePer/2):P.fs/P.f*(antMalePer/2)) + ...
    P.MalePkt.R0 / P.c *P.fs;

P.Focus = [0, 1*P.MalePkt.R0, 0].';



% Definerer pulsen
P.PulsePeriods =3.5; % Antall perioder i pulsen
P.tPulse = -P.fs/P.f*(P.PulsePeriods/2):P.fs/P.f*(P.PulsePeriods/2);  % Time samples pulse
% Lage puls som ikke har DC
% Må tilpasses til lengden av pulsen. 'pdfStartStop' er testet for
% P.Pulsperiode == 3, 3.5 og 4.
pdfStartStop =3.01874285;
P.Pulse = (-1)^floor(P.PulsePeriods)*sin(2*pi*linspace(0,P.PulsePeriods,length(P.tPulse))).*
...
    normpdf(linspace(-pdfStartStop,pdfStartStop,length(P.tPulse)));
%P.Pulse=(S(:,60));
P.Pulse = P.Pulse / max(P.Pulse);
figure(4)
plot(P.Pulse);
xlabel('Puls period[{\mu}/s]')
ylabel('Amplitude ')
P.lambda = P.c/P.f;

% Definerer elementposisjoner
if 0,
    P.M = 2;  % Antall elementer
    P.elpos=zeros(3,P.M);
    P.elpos(1,:) = [0 P.MalePkt.R0/4];
    P.elpos(2,:) = 0;
    P.elpos(3,:) = 0;
    P.elCenter = [0 0 0]';
elseif 0,
    P.M = 2;  % Antall elementer
    P.elpos=zeros(3,P.M);
    tmpTh = atand(-4/3);
```

```
    P.elpos(1,:) =  P.MalePkt.R0/4*[sind(0) sind(tmpTh)];
    P.elpos(2,:) = P.MalePkt.R0/4*[cosd(0) cosd(tmpTh)];;
    P.elpos(3,:) = 0;
    P.elCenter = P.MalePkt.R0/4*[0 cosd(0) 0]';
elseif 0
    P.M = 3;  % Antall elementer
    P.elpos=zeros(3,P.M);
    P.elpos(1,:) = P.lambda.*[-(P.M-1)/2:(P.M-1)/2]
    P.elpos(2,:) = 0;
    P.elpos(3,:) = 0;
    P.elCenter = [0 0 0]';
else % sylinderarray
    P.M = 17;  % Antall elementer
    P.elpos=zeros(3,P.M);
    % Buen skal være (P.M-1)*lambda/2 lang
    b = (P.M-1)*P.lambda/2;
    % Arrayet skal dekke +-60 grader
    Th = 2*deg2rad(30);
    P.ElTheta = linspace(-Th/2,Th/2,P.M);
    % Kan da regne radius på sylinder
    P.ElR = b/Th;
    [y,x,z] = pol2cart(P.ElTheta,P.ElR,0);

    P.elpos(1,:) = x;
    P.elpos(2,:) = y;
    P.elpos(3,:) = z;
    P.elCenter = [0 P.ElR 0]';
end
```

## ParametreKompl1.m

```
 function P = Parametre

% Function that sets up a struct of parameters needed for CalcField()
%


P.verbose = 1;  %

P.f = 100e3;   % Input signal freq. in Hz
P.fs = 100*P.f;   % Samplingsfrekvens
P.c = 1488;
% Definerer målepunkter:
P.MalePkt.R0 = 1e0;  % Avstand til målepunkt målt i meter
if 0,
    P.MalePkt.numPoints = 2; % Antall målepunkter
    P.MalePkt.Theta = deg2rad([0 -45]);
else
    P.MalePkt.numPoints = 10; % Antall målepunkter
    P.MalePkt.Theta = deg2rad(linspace(-60,60,P.MalePkt.numPoints));
    %P.MalePkt.Theta = deg2rad(linspace(0,358,P.MalePkt.numPoints));
    %P.MalePkt.Theta = 2*pi*[0:P.MalePkt.numPoints-1]/P.MalePkt.numPoints;
    %P.MalePkt.Theta = linspace(-pi/2,pi/2,P.MalePkt.numPoints);
end
P.MalePkt.Coord = zeros(3,P.MalePkt.numPoints);
for i_m=1:P.MalePkt.numPoints
    [y,x,z] = pol2cart(P.MalePkt.Theta(i_m),P.MalePkt.R0,0);
    P.MalePkt.Coord(:,i_m) = [x y z].';
end


% Definerer måletidspunkter:
antMalePer = 1*20;
P.MalePkt.tMalePkt = (-P.fs/P.f*(antMalePer/2):P.fs/P.f*(antMalePer/2)) + ...
    P.MalePkt.R0 / P.c *P.fs;

P.Focus = [0, 1*P.MalePkt.R0, 0].';



% Definerer pulsen
if 0,
    P.PulsePeriods = 3.5; % Antall perioder i pulsen
    P.tPulse = -P.fs/P.f*(P.PulsePeriods/2):P.fs/P.f*(P.PulsePeriods/2);  % Time samples pulse
```

```
    % Lage puls som ikke har DC
    % Må sikkert tilpasses til lengden av pulsen. 'pdfStartStop' er testet for
    % P.Pulsperiode == 3, 3.5 og 4.
    pdfStartStop = 3.01874285;
    P.Pulse = (-
1)^floor(P.PulsePeriods)*sin(2*pi*linspace(0,P.PulsePeriods,length(P.tPulse))).* ...
        normpdf(linspace(-pdfStartStop,pdfStartStop,length(P.tPulse)));
    P.Pulse = P.Pulse / max(P.Pulse);
else
    tmp = cos(2*pi*(0:1/(P.fs/P.f):15));
    P.Pulse= tmp.*tukeywin(length(tmp),.4).';

    P.Pulse = P.Pulse / max(P.Pulse);
    P.tPulse = [ -(length(P.Pulse)-1)/2 : (length(P.Pulse)-1)/2];
end
if 1,
    % lager kompleks puls
    P.Pulse = hilbert(P.Pulse);
end
P.lambda = P.c/P.f;

% Definerer elementposisjoner
if 0,
    P.M = 2;  % Antall elementer
    P.elpos=zeros(3,P.M);
    P.elpos(1,:) = [0 P.MalePkt.R0/4];
    P.elpos(2,:) = 0;
    P.elpos(3,:) = 0;
    P.elCenter = [0 0 0]';
elseif 0,
    P.M = 2;  % Antall elementer
    P.elpos=zeros(3,P.M);
    tmpTh = atand(-4/3);
    P.elpos(1,:) =  P.MalePkt.R0/4*[sind(0) sind(tmpTh)];
    P.elpos(2,:) = P.MalePkt.R0/4*[cosd(0) cosd(tmpTh)];;
    P.elpos(3,:) = 0;
    P.elCenter = P.MalePkt.R0/4*[0 cosd(0) 0]';
elseif 0
    P.M = 3;  % Antall elementer
    P.elpos=zeros(3,P.M);
    P.elpos(1,:) = P.lambda.*[-(P.M-1)/2:(P.M-1)/2]
    P.elpos(2,:) = 0;
    P.elpos(3,:) = 0;
    P.elCenter = [0 0 0]';
else % sylinderarray
    P.M = 24;  % Antall elementer
    P.elpos=zeros(3,P.M);
    % Buen skal være (P.M-1)*lambda/2 lang
    b = (P.M-1)*P.lambda/2;
    % Arrayet skal dekke +-60 grader
    Th = 2*deg2rad(30);
    P.ElTheta = linspace(-Th/2,Th/2,P.M);
    % Kan da regne radius på sylinder
    P.ElR = b/Th;
    [y,x,z] = pol2cart(P.ElTheta,P.ElR,0);

    P.elpos(1,:) = x;
    P.elpos(2,:) = y;
    P.elpos(3,:) = z;
    P.elCenter = [0 P.ElR 0]';
end
```

## A2.Prosessing of measured data

Contains the matlab scripte for prosessing and back propagating of of the measured data
Testmaal.m

### Testmaal.m

```
%Prosesserer målte data.
load('exp_front120p_d100cm.mat');
```

```
Sgm = S;
S = S(:,21:101);

%Avstanden mellom transducer og målepunkter er 1m
%Lydhastigheten i tanken er på 1488 ved 20 grader celsius
%Ble gjort 120 målinger tilsvarened 120 målepunkter

% 1: Definerer P (Kan kalle Parametre.m)
P = Parametre;

% 2: Beregner feltet som måles i målepunkter:
B = CalcField(P);

% 3A: Finner et eksempel på målt signal, gitt parametre i P:

%BB = reshape(B.Pulser,length(P.MalePkt.tMalePkt)*P.MalePkt.numPoints,P.M);
%%
Data=zeros(1000,81,17);
for i_el =1:17
    PhiElSer = (i_el-9)*7.5+[-60:1.5:60];
    Phi_ind = find(PhiElSer >= -60 & PhiElSer <= 60);
    Data(:,Phi_ind+(i_el-9)*5, i_el)=S(:,Phi_ind);
end


if 0
for ii=1:17

    imagesc(squeeze(Data(:,:,ii)));
    %title(int2str(ii));
    xlabel('Number of measuring points');
    ylabel('Sampling time');
    disp('Press any key to continue;-)');
    pause
end
end

%%
%

BB = reshape(Data,(length(P.MalePkt.tMalePkt)-1)*P.MalePkt.numPoints,P.M);
Vekt1 =ones(P.M,1);
MaltSignal =  BB* Vekt1;

% LS-løsning
disp('Ideell verden, rett lydhastighet')
x1 = BB\MaltSignal;
% LS-Feil:
sum((MaltSignal - BB*x1).^2)
figure(1)
stem(x1);
xlabel('Number of elements');
ylabel('response over all elements');
title('Response over all elements distance 1m');
%%
% 3B: Forandrer lydhastighet og lager annet målt signal:
Pny = P;
Pny.c = 1488;

B2 = CalcField(Pny);

%
B2.Pulser=B2.Pulser(1:1000,:,:);
Skalering = max(BB(:));
B2.Pulser = B2.Pulser/max(B2.Pulser(:))*Skalering;
BBny = reshape(B2.Pulser,(length(P.MalePkt.tMalePkt)-1)*P.MalePkt.numPoints,P.M);

%
Window=ones(Pny.M,1)+0.1*randn(Pny.M,1);
Vekt2 =ones(Pny.M,1);
MaltSignal2 =  BBny* Vekt2;

% LS-løsning: NB: Tester målematrise BB (beregnet med P.c = 1500) med
% målte data gitt at lydhastigheten i vannet var 1496:
disp('Ideell verden, feil lydhastighet')
x2 = BB\MaltSignal2;
```

```
% LS-Feil:
sum((MaltSignal2 - BB*x2).^2)
figure(2)
x2=abs(x2);
stem(x2);
xlabel('Number of elements');
ylabel('Response over all elements');
title('Response with Pulse 3.5 periods');



figure(3)
stem(x1);
hold on
stem(x2,'r');
xlabel('Number of elements');
ylabel('Response over all elements');
hleg=legend('1488 m/s','1486 m/s');
title('Response different speed of sound')
return
figure(4)
subplot(2,1,1);
stem(Vekt2);
%xlabel('Number of elements');
ylabel('Response over all elements');
title('Amplitude of random weighting')
subplot(2,1,2);
stem(Vekt2);
hold on
stem(x2,'r');
axis([0 25 0 1.5]);
xlabel('Number of elements');
ylabel('Response over all elements');
title('Data with random weighting');
%hleg=legend('Hanning','Data with Hanning');
```

# References


[1] Ashok Ambardar. Digital signal processing: A modern Introduction, Thompson 2007 ISBN 0495082384

[2] Svein Bø. Tank positioning and measurements, User guide, DSB group Dept. of Informatics, University of Oslo 2013

[3] J. Christ and F.M. Landstorfer. "Overcoming the typical limitations of cylindrical Nearfield to farfield transformation", 25th European Microwave Conference, pp.1245-1250, IEEE 1995

[4] N.A. Cochrane. "Near field considerations for Simrad-Mesotech SM 2000 Multi-Beam Sonar" *Canadian Technical Report of Fisheries and Aquatic sciences,* Bedford institute Of oceanography, July 2002

[5] Cornell University. http://www.acousticsunpacked.org/acoustics/AcousticBacground/ NearFar.asp [Online; accessed Feb.2014]

[6] Cornell University. . http://www.acousticsunpacked.org/acoustics/AcousticBacground/ AcousticTransducer.asp [Online; accessed Jan. 2014]

[7] Fosstech. Position system, Technical specifications for position system, DSB wiki [Online; accessed Sep.2013]

[8] Food and Agriculture Organization of the United Nations FAO, http://www.fao.org/docrep/003/x6602e/x6602e02.htm [Online Accessed Jan. 2014]

[9] Roy Edgar Hansen. http://www.uio.no/studier/emner/matnat/ifi/INF-GEO4310/h13/ Undervisningsmateriell/sonar_introduction_2013.pdf, 2013. [Online; accessed Okt. 2013]

[10] Roy Edgar Hansen. http://heim.ifi.uio.no/~rhn/image_gallery.shtml, 2007 [Online; Accessed Aug. 2014]

[11] Monson H. Hayes. *Statistical digital signal processing and modeling*, John Wiley & sons Inc. 1996 ISBN 0471594318

[12] Igcse physics 2011. http://igcsephysics2011.tumblr.com/page/3 [Online; Accessed Jan.20024]

[13] Jørgen Arendt Jensen. Field II userguide, http://field-ii.dk/documents/users_guide.pdf [Online; accessed Aug. 2013]

[14] D.H. Johnson and D:E: Dudgeon. *Array Signal Processing: Concepts and Techniques*. Simon & Schuster 1993. ISBN 0130485136

[15] Jan Egil Kirkebø. *Curved and spherical arrays.* Cand. Scient. Thesis, Department of Informatics, University of Oslo, 2002.

[16] Jan Egil Kirkebø. *Beamforming for Imaging: A Brief Overview*. https://heim.ifi.uio.no /janki/BeamformingForImaging.pdf [Online; accessed Des 2013]

[17] Rodney A. Kennedy, Thushara D. Abhayapala and Darren B. Ward. "Broadband Nearfield beamforming using a radial beampattern transformation" *IEEE Trans. Signal Processing*, vol. 42, Issue 8, pp.2147-2156, Aug 1998

[18] Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppens and James V. Sanders *Fundamental of acoustics*, John Wiley & sons Inc. fourth edition 2000

[19] Kongsberg-Mesotech. http://www.km.kongsberg.com/ks/web/nokbg0397.nsf/ AllWeb/BFF55C1E99AF1340C1257B1D0040087F/$file/4-Field-Transition-of-a-Sonar-Beam.pdf?OpenElement, [Online; accessed April 2014]

[20] Qihu Li. *Digital Sonar Design in Underwater Acoustics: Principals and Applications* Springer, 2012 ISBN 9783642182891

[21] National Oceanic and Atmospheric Adminiatartion. http://oceanexplorer.noaa.gov/ explorations/sound01/background/acoustics/acoustics.html [Online; accessed Nov. 2013]

[22] National Physics Laboratory UK. http://resource.npl.co.uk/acoustics/techguides/ soundpurewater/content.html [Online accessed Feb. 2014]

[23] Trygve Sparr. "Imaging Radar" *Lecture notes*, UNIK, University of Oslo 2014

[24] Simrad. http://www.simrad.com/www/01/NOKBG03797.nsf/AllWeb/5AEA56CA 3E1AAAC7C125762A00324EF3/$file/337405aa_sh90_product_description_english .pdf?OpenElement, *Product description* [Online; accessed Aug 2013]

[25] Simrad. http://www.simrad.com/www/01/NOKBG03797.nsf/AllWeb/E76DF830EC25 90C4C1257C2B003581D/$file/388753aa_sh90_technical_specifications_english .pdf?OpenElement, *Technical specifications*[ Online; accessed Aug 2013]

[26] Bruno Siciliano and Oussama KHatib. *Springer Handbook of Robotics*, Springer 2008, ISBN 9783540239574

[27] Robert J. Ulrick. *Principals of underwater sound*, MacGraw-Hill book company, New york US, third edition 1983 ISBN 0932146627

[28] Wikipedia. http://en.wikipedia.org/wiki/Sonar [Online; accessed March 2014]

[29] Wikipedia. http://en.wikipedia.org/wiki/Beamforming [Online; accessed Jan. 2014]

[30] Wikipedia. http://commons.wikimedia.org/wiki/File:Spherical_wave.png 2014b [Online; accessed March 2014]