UiO : **Department of Informatics**
University of Oslo

# Static Elastography With Ultrasound Using Adaptive Beamforming

Ole Marius Hoel Rindal

Master's Thesis Spring 2014

UNIVERSITAS OSLOENSIS MDCCCXI

*This thesis is dedicated to my mother.*

*If cancer diagnosis and treatment was further developed, you might have lived to see this thesis completed.*

# Abstract

## Background and motivation

The health of human tissue can be indicated by the stiffness of the tissue. It is known that the risk of a nodule being malignant is increased with the stiffness of the nodule. Elastography is an imaging mode capable of displaying the stiffness of the tissue. Static elastography with ultrasound consists of creating a pre- and post-compression ultrasound image where the tissue being imaged has been compressed between the images. The displacement of tissue is calculated along the axial dimension based on the assumption that speckle pattern follows tissue movement. Tissue strain, indicating the stiffness of tissue, can then be found from the displacement of the tissue.

Speckle statistics and the speckle pattern are different for images created with conventional and adaptive (Capon) beamforming. The speckle pattern created with adaptive beamforming has a smaller and more distinct pattern because of the improved resolution by adaptive beamforming. Hypothetically a more distinct pattern should result in better correlation and thus better displacement estimation.

Recently it has been shown that lateral oversampling is needed to achieve lateral shift-invariance between image frames when using adaptive beamforming. Shift-invariance between frames is especially important for elastography since the displacement estimate is based on correlation between two nearly identical frames.

## Approach

To simulate static elastography two speckle images are created with Field II simulations based on the same scatter phantom, where the scatterers have been displaced axially to create pre- and post-compression ultrasound images. The images are created with the conventional beamformer and the adaptive beamformer with different parameters. In the middle of the phantom a circular object has constant displacement to mimic a hard malignant nodule in the tissue.

## Results and conclusions

We show that lateral oversampling is necessary for single frame scenarios when doing adaptive beamforming and to achieve shift- invariant imaging of speckle. The speckle pattern from adaptive beamforming is more distinct, but our research shows that adaptive beamforming with certain parameters gave similar performance for axial correlation for displacement estimation as conventional beamforming and thus similar accuracy when doing static elastography.

# Contents

*"The highest activity a human being can attain is learning for understanding, because to understand is to be free."*

Baruch Spinoza

# Acknowledgement

This thesis could not have been written without the inspiration from the wonderful DSB-group at the University of Oslo. The people there have provided an excellent working environment, as well as fruitful lunch discussions. I have benefited from years of industrial experience, through Sten Roar Snare, and the academical authority of Fritz Albrightsen, who was so kind to read this whole thesis and provided valuable feedback.

Especially I would like to thank my supervisors Sverre Holm and Andreas Austeng. Sverre introduced me to the topic of elastography and with his great technological insight suggested to investigate the topics in this thesis. Andreas never closes the door to his office and always leaves the door ajar. My interpretation of this open door, hopefully correctly, has been that he is always open for questions. Even though these questions could be as frequent as many times a day, Andreas welcomed me with his friendly nature and contagious laughter helping me solve whatever problem I had.

My family deserves my gratitude for always supporting me, especially my mother who I dearly miss - and will always admire. Lastly Tuva, who have beard through long days of me working with this thesis, and willingly shared both the great frustrations and joys that arose from the work with this master's thesis.

Thank you all.

# Chapter 1

# Introduction

The stiffness of tissue is in many cases a good indicator on the health of human tissue. An example demonstrating this statement is cirrhosis of the liver, where the liver tissue is replaced by much harder types of tissue like fibrosis and scar tissue, which degrades the liver functions. Research also reports that a harder nodules in tissue is associated with increased risk of malignancy (Rago et al., 2007).

The traditional way of examining the tissue stiffness is palpation, where the tissue is felt with the fingers or hand during a physical examination. Palpation has its obvious drawbacks; it is not quantitative, and there is large variability between examinations. Palpation also requires that the tissue to be examined is close to the surface, because deeper tissue is hard to access with the hands and can be hidden by more solid structures.

Elastography is a relatively new approach to measure tissue stiffness. Ultrasound elastography creates an image of the tissue indicating stiffness using harmless ultrasound techniques. Elastography is superior to palpation in many ways, but especially when regarding quantitative measures and examination variability. For these reasons elastography is a significant tool in more precise and correct diagnosis, offering possibilities to discover potential disease at an earlier stage using noninvasive techniques (Rago et al., 2007).

In medical ultrasound imaging adaptive beamforming has been applied resulting in increased resolution creating more detailed images. Another effect of the adaptive beamforming is a smaller and more distinct speckle pattern in the ultrasound images. Hypothetically this could mean better estimation of tissue movement and thus being beneficial when doing elastography.

## 1.1 Objective of thesis

The original objective of this thesis was to investigate if we could get quantitative results from static elastography by measuring the force applied from the ultrasound probe when using the probe to compress the tissue. A second objective was to investigate if the smaller and more distinct speckle pattern created by adaptive beamforming has benefits when doing static elastography.

To measure the force applied from the ultrasound probe, a system was developed as described in Chapter 4. This system turned out to be very inaccurate. Because of this, the main objective of the thesis shifted, and became to thoroughly investigate if adaptive beamforming has benefits when applied to static elastography. Towards this overall objective we introduce and describe both conventional and adaptive beamforming. We will investigate simulation of ultrasound images, both single point scatterers and speckle, and describe and discuss some well known and new results. Especially the recent research result showing that lateral oversampling is needed when doing adaptive beamforming. Two displacement estimators will be implemented and discussed to compare adaptive and conventional beamformed images for static elastography.

Because of the wide scope of this thesis, and the many partial results, we have permitted a more informal structure of this thesis than the common *IMRaD*. Using a more of a interweaved structure allows us to discuss some results as they arrive, and use the partial results to make qualified choices in the later parts of the thesis.

## 1.2 Key results

Adaptive (Capon) beamforming applied to ultrasound static elastography have similar performance for axial correlation for displacement estimation as conventional (DAS) beamforming, and thus similar accuracy when doing static elastography.

Our hypothesis was that the more distinct speckle pattern, smaller speckles, created by the Capon beamformer, would provide better correlation between the pre- and post-compression images. However, our research showed that it was in fact the Capon parameters creating similar speckle statistics as DAS, and not the Capon parameters giving the most distinct speckle pattern, that produced the best displacement estimation results for the Capon beamformer.

For single frame scenarios lateral oversampling, closer beam distance, is necessary when using Capon beamforming in ultrasound imaging. Imaging well developed speckle needed a lower oversampling factor than images of single point scatterers.

## 1.3 Thesis outline

**Chapter 2** briefly introduces ultrasound imaging, elastography, conventional and adaptive beamforming providing the background and some theory for the rest of the thesis.

**Chapter 3** describes how ultrasound images can be simulated and provides theoretical discussions on how the ultrasound probe influence the resolution and thus the details in the ultrasound images. We investigate and compare the adaptive and conventional beamforming for single point scatterer images and speckle images and especially investigate a recent result indicating that lateral oversampling is needed when creating images with the adaptive beamformer.

**Chapter 4** stands alone and describes the construction of a system to measure the force applied by the ultrasound probe towards tissue.

**Chapter 5** continues from Chapter 3 and investigates static elastography and describes two estimators to find tissue displacement. A method to calculate strain in the tissue from the estimated displacement is described. Simulation of static elastography is described and we build the framework, including comparison criteria for the two beamformers applied to static elastography, needed for the next chapter.

**Chapter 6** compares the performance of adaptive and conventional beamforming applied to static elastography. Multiple parameters and setups based on previous results from the thesis are applied and compared. The results is discussed in detail and an explanation of the results is suggested.

**Chapter 7** concludes our most important results and suggests some interesting future work, which sadly was beyond the scope and time restriction of a master thesis.

# Chapter 2

# Background and theory

***Chapter abstract:*** *This chapter will give a brief insight into the physics behind an ultrasound image, and briefly explain how an ultrasound image is created. Elstography is introduced in the second part, and both static and shear wave ultrasound elastography are briefly described. The third part of the chapter introduces beamforming. The theoretical background of conventional and adaptive beamforming is presented.*

## 2.1 Medical ultrasound imaging

Medical ultrasound imaging enables us to noninvasively create images of the inside of the body, by transmitting high frequent sound into the body. We will let us inspire by parts of the introduction to ultrasound by Jensen in his book *Estimation of Blood Velocities Using Ultrasound* (Jensen, 1996b), and get a brief insight into the physics behind ultrasound imaging.

Sound waves are compressional waves, compressing the medium along the direction the wave is traveling. When we speak, our voice cause pressure differences in the air. Ultrasound transmitted into the body, creates small disturbances in the medium in which the wave is propagating.

The wave will propagate in a constant manner as long as the medium has similar acoustic properties. If the properties change, a part of the wave will be reflected, while another part will continue to propagate through the medium. The pressure reflection coefficients are given as

$$R = \frac{Z_2 - Z_1}{Z_1 + Z_2}$$

$Z_n = p_n c_n$ : Characteristic impedance of medium $n$ where $p_n$ is medium density, and $c_n$ is speed of propagation.

The transmitted wave's direction is given by the angle $\theta_t$ dependent on the angle of incidence $\theta_i$, both angles are given by the well known Snell's law:

$$\frac{c_1}{c_2} = \frac{\sin\theta_t}{\sin\theta_i}.$$

So far our arguments require a sharp boundary of change between the acoustic properties of two different medium. This is rarely found in the human body, and is thus a simplification. What we are actually imaging is scattering of the ultrasound waves. Ultrasound waves are scattered into all directions because of small changes in the impedance of the medium e.g. small changes in density or absorption. Some parts of these scattered waves will travel back to the transducer where they are recorded and combined to display the ultrasound image.

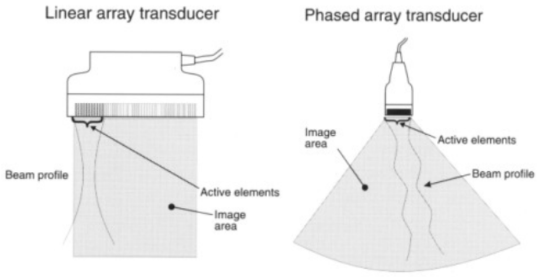
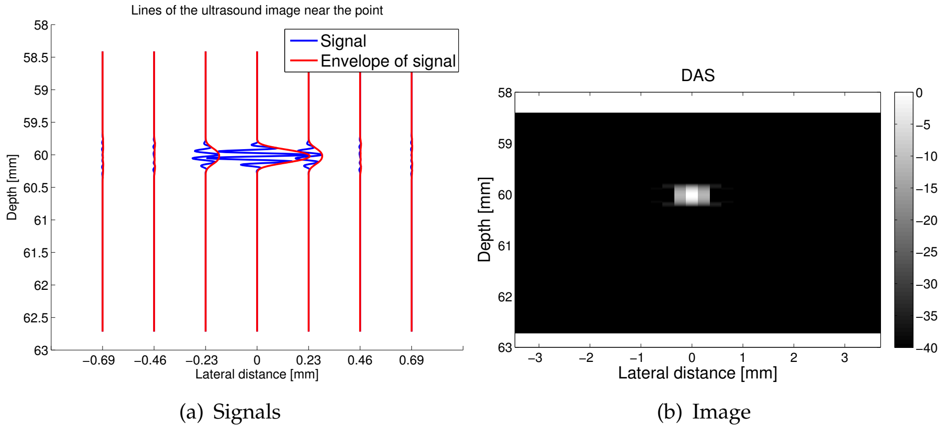


*Figure 2.1: A linear array and a phased array transducer. Figure from (Jensen, 1996b).*

There are many types of transducers used for ultrasound imaging. The most common are the linear array transducer and the phased array transducer, see Figure 2.1. The difference between these two transducers is how they scan the image area. The phased array transducer creates beams in a fan-shaped area in front of the transducer, and creates a fan-shaped image. The linear array transducer creates parallel beams straight in front of the transducer only using a given number of active elements, and then creates the next beam by moving which elements are used. The linear array transducer will therefore create a rectangular image.



(a) Signals

(b) Image

*Figure 2.2: Ultrasound image of a single point scattering the ultrasound waves.*

In Figure 2.2 we have simulated a single point scattering the ultrasound waves imaged with a linear array transducer. In plot (a) we have vertically plotted the received signals of the 7 central beams of the image. We see that at 60 mm depth we have gotten a backscattered signal with the most energy at the three central beams. The blue lines in the plot indicate the actual RF-signal (radio frequency-signal) received, while the red is the envelope of the signal, see Appendix B. In the image (b) we have taken the decibel values of the envelope and displayed the decibel amplitude as different color intensities.

A single scatterer does not occur *in vivo*. What we see in ultrasound images is the constructive and destructive interference of backscattered signals from many small structures of much smaller size than the ultrasound wavelength . The resulting patterns in the image is known as speckle and is something we will investigate in depth in Chapter 3. The speckle pattern does not directly reveal the underlying structure, it is actually a random process, but slight movements in the tissue will only create a slight movement in the speckle pattern and thus different measurements can be correlated to find the movement of the tissue. The fact that we can estimate tissue movement from the movement of speckle leads us to our next topic; elastography.

## 2.2 Elastography

Elastography is the technique used to measure the stiffness or elasticity of tissue. When doing ultrasound elastography there are mainly two techniques; static and shear wave elastography. Both techniques follow three common steps (Bercoff, 2008);

Step 1 Generate low frequent vibration in the tissue to induce shear stress.

Step 2 Image the tissue to analyze the resulting stress.

Step 3 Extract from movement of tissue a parameter related to the tissue stiffness.

What differentiates the two techniques is how each step is performed.

The goal of elastography is to find the stiffness of the tissue, mathematically this is measured by Young's modulus:

$$E = \frac{\text{stress}}{\text{strain}} = \frac{\sigma}{\epsilon} = \frac{F/A_0}{\Delta L/L_0} \quad \left| \quad \begin{array}{r l} F : & \text{The force applied to the tissue} \\ A_0 : & \text{The original area the force is applied} \\ \Delta L : & \text{The change in length of the object} \\ L_0 : & \text{The original length of the object.} \end{array} \right. \tag{2.1}$$

Verbally this means that a force applied to an area causes compression, stress $\sigma$, which induces a deformation, strain $\epsilon$, in the tissue. Young's modulus is measured in Pascal (Pa) and a list of typical values for tissues in the body is listed in Table 2.1.

The main differences between the two ultrasound elastography techniques are how the movement in the tissue is created. The two techniques

| Type of tissue | | Young's Modulus (E in kPa) | Density |
|---|---|---|---|
| Breast | Normal fat | 18-24 | 1000 ± 8% ≈ water |
| | Normal glandular | 28-66 | |
| | Fibrous tissue | 96-244 | |
| | Carcinoma | 22-560 | |
| Prostate | Normal anterior | 55-63 | |
| | Normal posterior | 63-71 | |
| | BPH (beningn) | 36-41 | |
| | Carcinoma | 96-241 | |
| Liver | Normal | 0.4-6 | |
| | Cirrhosis | 15-100 | |

*Table 2.1: Elasticity values for different tissues (Bercoff, 2008).*

and their differences are briefly described in the following sections. For in depth descriptions see the articles (Ophir et al., 1991) and (Bercoff et al., 2004). In Chapter 5 we will examine the details of static elastography and implement different methods to estimate the displacement of tissue.

### 2.2.1 Static elastography

Static elastography was the first technique suggested to do ultrasound elastography (Ophir et al., 1991). In static elastography the movement in the tissue is created by the ultrasound probe itself. First a pre-compression image of the tissue with an initial force between the probe and surface is created. Then the operator applies more force on the surface with the probe compressing the tissue and a post-compression image is created. We then have two images of the tissue, pre- and post-compression, and the next step is to analyze the two images and extract how much the tissues have moved. A tissue moving less than another tissue with the same force applied is assumed to be stiffer. A good illustration of this technique applied to a phantom can be seen in Figure 2.3.

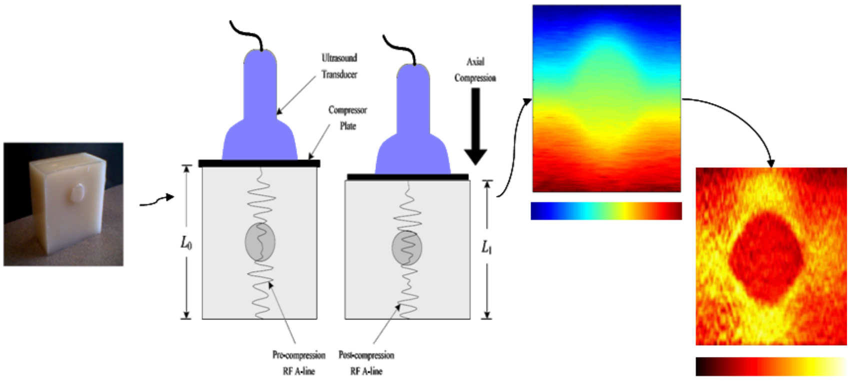


*Figure 2.3: Static elastography. Figure from (http://www.ultrasonix.com/wikisonix/index.php/Elastography).*

The image in the middle in Figure 2.3 displays the displacement of the tissue. From the displacement of tissue we can calculate the strain, displayed in the image to the right. There are different techniques to estimate the displacement, and we will investigate two of them in Chapter 5; the crosscorrelation technique and a pulsed-Doppler technique. For now we will concentrate on the crosscorrelation technique introduced by Ophir et al. (1991) and maybe better explained in (Ophir et al., 2002). The two pre- and post-compression images consist of a number of RF-data lines. RF-data is the raw beamformed data from the ultrasound probe, the data plotted in blue in Figure 2.2 (a). We will introduce beamforming in the next sections. Each set of corresponding RF-data lines are subdivided into small temporal windows that are crosscorrelated to find the change in arrival times of the echoes before and after compression. Since we know the approximate wave velocity in tissue the change in arrival times gives us the displacement of tissue. The local strain can then be computed, from (Ophir et al., 2002), as

$$\epsilon = \frac{(t_{1b} - t_1) - (t_{2b} - t_{2a})}{t_{1b} - t_{1a}} \tag{2.2}$$

$t_{1a}$ : Arrival time of the pre-comp. echo from the proximal window

$t_{2a}$ : Arrival time of the post-comp. echo from the proximal window

$t_{1b}$ : Arrival time of the pre-comp. echo from the distal window

$t_{2b}$ : Arrival time of the post-comp. echo from the distal window.

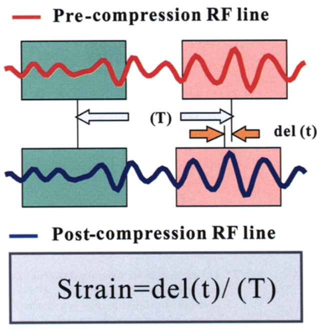


*Figure 2.4: "A schematic showing the process of computing the strain in a tissue segment. Congruent windowed segments of the pre-compression and post-compression signals are compared by crosscorrelation. While the early windowed segments exhibit virtually no delay, a finite delay (designated del (t)) is detected between the later segments. The strain is computed as the gradient of the time delay (or displacement), i.e. strain = del(t)/T, where T is the initial (pre-compression) separation between the windowed segments." Ophir et al. (2002).*

Figure 2.4 is a graphical illustration and explanation of Equation (2.2). The windows are created in small overlapping steps along the temporal axis and the calculation is done for all steps. This technique assumes that the speckle pattern in the image follows the motion of the tissue.

### 2.2.2 Shear wave elastography

In shear wave elastography the movement in the tissue is created by shear waves induced by the regular compressional ultrasound waves. So, first we need to distinguish between the different types of waves. Since we are dealing with waves traveling inside tissue we disregard surface waves and only look at the two types of body waves.

#### Compressional waves

Compressional or pressure waves is the first kind of body waves created by deforming the material along the direction that the wave is traveling, see Figure 2.5 a. An example of pressure waves are sound waves that changes the pressure in the medium when they propagate. Compressional waves travel through all types of materials including solids, liquids and gases. Sound waves are pressure waves, so ultrasound imaging is as we know done with compressional waves. Compressional waves propagate at a speed given by

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} \approx \sqrt{\frac{\lambda}{\rho}}$$

$\lambda$ : The bulk modulus
$\mu$ : The shear modulus
$\rho$ : The density of the material.

The bulk modulus, $\lambda$, measures the material's resistance to uniform compression measured in Pascal (Pa) usually in the order of $10^9$ Pa. The shear modulus, $\mu$, measuring the ratio between shear stress and shear strain, is also measured in Pascal. The value of $\mu$ varies between $10^2$ and $10^7$ Pa, so $\lambda \gg \mu$ and we can do the approximation above.

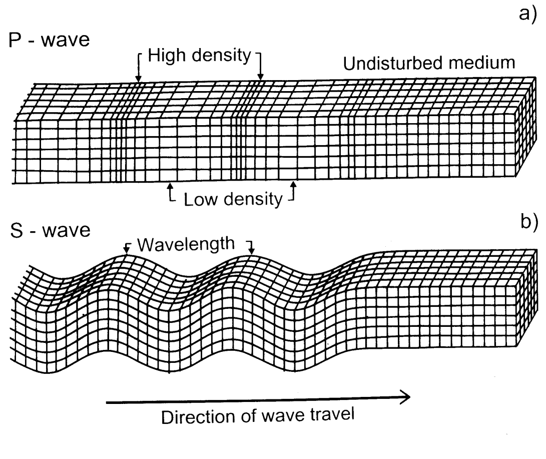


*Figure 2.5: Pressure wave (p-wave) and shear wave (s-wave). Figure from (http://www.astro.uwo.ca/~jlandstr/planets/webfigs/earth/slide1.html).*

### Shear waves

Shear waves is the second kind of body waves. Shear waves create a transverse or shear motion perpendicular to the direction the wave is moving, see Figure 2.5b. We can imagine a wave traveling through a rope moving the rope perpendicular to the direction the wave is moving. While pressure waves travel through all kinds of materials shear waves only propagate through solids, not liquids and gases. This is not the only property differentiating the two; the speed of propagation is also very different. Shear waves propagate at

$$c_s = \sqrt{\frac{\mu}{\rho}} \quad \Bigg| \qquad \begin{array}{l} \mu : \text{The shear modulus} \\ \rho : \text{The density of the material.} \end{array}$$

If we compare the two waves' propagation speeds, we see that the shear wave propagate at a much slower speed than the pressure wave, allowing us to image the traveling shear wave.

### Creating and imaging shear waves

To induce the shear wave in the tissue there are mainly two different techniques. The first was to use an external mechanical vibrator e.g. Fibroscan® (Audiere et al., 2009). Firboscan is used clinically to measure the stiffness of the liver. The clinical applicability of this technique is limited because of the bulky external vibrator (Bercoff et al., 2004). Another, and more interesting approach, is to induce the shear waves by ultrasound; supersonic shear imaging (SSI) (Bercoff et al., 2004). This is done by sending multiple ultrasound *pushes*, 400 oscillations at 4.3 MHz, giving a pushing time of 100 $\mu$s for each push. The pushes are placed along the beam direction causing the shear sources to interfere constructively along a *Mach cone* creating two plane shear waves propagating in opposite directions, see Figure 2.6. The shear waves are thus created by a series of high power compressional waves and since the compressional waves move much faster than the shear waves it is possible to create multiple shear wave sources leading to the constructive interference. For a more in depth explanation and description see Bercoff et al. (2004).

Even though the shear waves move slower than the compressional waves, one of the main challenges in SSI is to have high enough ultrasound image frame rate to be able to *catch* the traveling shear waves. While conventional ultrasound typically has a frame rate of 50 Hz, the SSI system needs a frame rate of 3000 - 6000 Hz. The ultrasound technique allowing these frame rates is plane wave imaging (Bercoff et al., 2004) and (Austeng et al., 2011), which is out of scope for this thesis.

Shear wave elastography provides a quantitative measure of Young's modulus in the tissue the shear wave is traveling. Equation (2.1) can be rewritten by introducing the fact that shear elasticity, $\mu$, is directly linked to shear elasticity if the medium is purely elastic. So we can rewrite it as

$$\mu = \rho c^2.$$

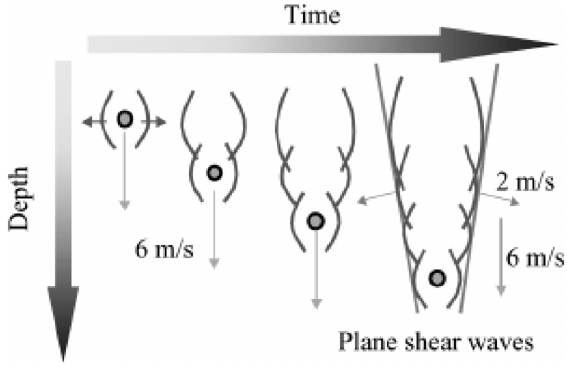


*Figure 2.6: Shear waves created by ultrasound shear sources. Figure from (Bercoff et al., 2004).*

If we also take into account that we are looking at soft tissues, so $\lambda \gg \mu$ we end up with (from Bercoff et al. (2004))

$$E \approx 3\mu = 3\rho c^2.$$

Since density, $\rho$, in body tissue is close to water, all that is needed is to estimate the shear wave speed to quantitatively estimate Young's modulus. In SSI this is done by using crosscorrelation techniques on the images of the propagating shear wave. An example image of an elasticity map of a phantom containing a 20-mm hard inclusion can be seen in Figure 2.7.

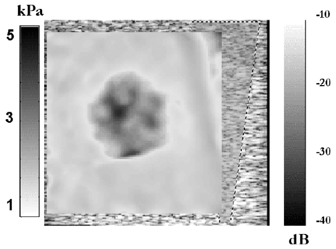


*Figure 2.7: Elasticity map of a phantom containing a 20-mm hard inclusion (Bercoff et al., 2004).*

## 2.3 Beamforming

Beamforming is a variety of array signal processing algorithms that focuses an array's signal capturing abilities in a particular direction (Johnson and Dudgeon, 1993, p. 111-112). In other words, while there are ways to alter the physical antennas to achieve better signal directivity and resolution these physical alterations of an antenna are not reversible nor flexible. Beamforming is to alter the recorded signal data to achieve better directivity and resolution. Beamforming is therefore a cheaper and much more flexible alternative. The drawback of beamforming is increased computation time and more complex signal processing algorithms. The best results are often obtained by a combination of good array properties combined with a sophisticated beamforming technique.

### 2.3.1 Conventional beamforming

Delay-and-sum (DAS) beamforming is often thought of as the conventional way of doing beamforming. Briefly explained DAS is delaying the signal on each individual sensor to steer in one direction and summing the delayed version of each sensors signal to one output signal. This gives a resulting signal with $M$ (the number of sensors) times better signal to noise ratio than one sensor, e.g. $SNR_{\text{sensor}} = \frac{\sigma_s^2}{\sigma_n^2}, SNR_{\text{array}} = M\frac{\sigma_s^2}{\sigma_n^2}$. In other words the array gain is equal to the number of sensors in the array. This simple example assumes uncorrelated white noise. An illustration of DAS beamforming can be seen in Figure 2.8.

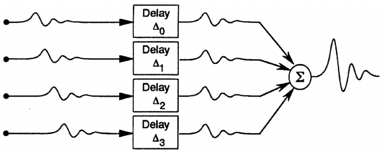


*Figure 2.8: Delay and sum beamforming. Figure from (Johnson and Dudgeon, 1993, p. 119).*

Mathematically we define the DAS beamformer as

$$z(t) = \sum_{m=0}^{M-1} w_m y_m(t - \Delta_{m,t}) \left| \begin{array}{rl} M & : \text{Number of elements} \\ m & : \text{Element number} \\ w_m & : \text{Element weight} \\ y_m & : \text{Signal from sensor } m \\ \Delta_{m,t} & : \text{Delay for sensor } m \text{ at time } t. \end{array} \right. \tag{2.3}$$

From this definition we see that DAS has another element not yet mentioned; weights. These weights can simply be set to $\frac{1}{M}$, favoring all sensors the same, or we can apply different windows e.g. Hamming which favors the central elements in the array.

### Delay-and-sum on vector form

The definition in Equation (2.3) can easily be extended to vector form. We can arrange the different parts of the equation as matrices

$$\left.\begin{array}{r}\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}, \widehat{\mathbf{Y}}(t) = \begin{bmatrix} y_0(t) \\ y_1(t) \\ \vdots \\ y_{M-1}(t) \end{bmatrix}, \\ \mathbf{Y}(t) = \begin{bmatrix} y_0(t-\Delta_0) \\ y_1(t-\Delta_1) \\ \vdots \\ y_{M-1}(t-\Delta_{M-1}) \end{bmatrix} \end{array}\right| \begin{array}{r} \mathbf{w} : \text{Element weights} \\ \widehat{\mathbf{Y}}(t) : \text{Received signals} \\ \mathbf{Y}(t) : \text{Delayed received signals.} \end{array}$$

This allows us to simplify Equation (2.3) to

$$z(t) = \sum_{m=0}^{M-1} w_m y_m(t - \Delta_{m,t}) = \mathbf{w}^H \mathbf{Y}(t).$$

Here $\mathbf{w^H}$ is the Hermitian of $\mathbf{w}$.

We see that the DAS beamformer allows a very simple and fast implementation. It is also very robust, the only assumptions made is that the speed of sound is constant. DAS is therefore on of the most used and best known beamforming techniques.

### Near field, far field

When doing array signal processing there is one important limit we need to keep track of, the limit between near field and far field. If the source sending (or reflecting) a signal is in the near field the signal will propagate as a spherical wave. If the transmitting source is further away from the array the signal propagating will seem like a plane wave propagating. What actually happens is that the radius of the sphere is so large that the wavefront approaches a plane wave. The limit between far field and near field is defined as (Wright, 1997)

$$R = \frac{D^2}{k\lambda} \;\Big|\; k = 1, 2, 3, 4, ...$$

Since $k$ is an integer we typically choose between 1 and 4, there is no hard limit. The limit depends on how much error we allow, and vary between fields of study.

In Figure 2.9 we have plotted propagating waves from sources placed at different distances from the array. The plots are created with the Field II (Jensen, 1996a)(Jensen and Svendsen, 1992) simulation environment. It is used an array with 10 elements of width 1 mm placed with pitch $\frac{\lambda}{2}$, giving an aperture of $D = 12.31$ mm when the frequency of the signal is 3 MHz assuming speed of sound $c = 1540$ m/s. The softest far field limit is $R_1 = \frac{D^2}{4\lambda} = 73.8$ mm while the strictest is $R_2 = \frac{D^2}{\lambda} = 295.2$ mm. If we look

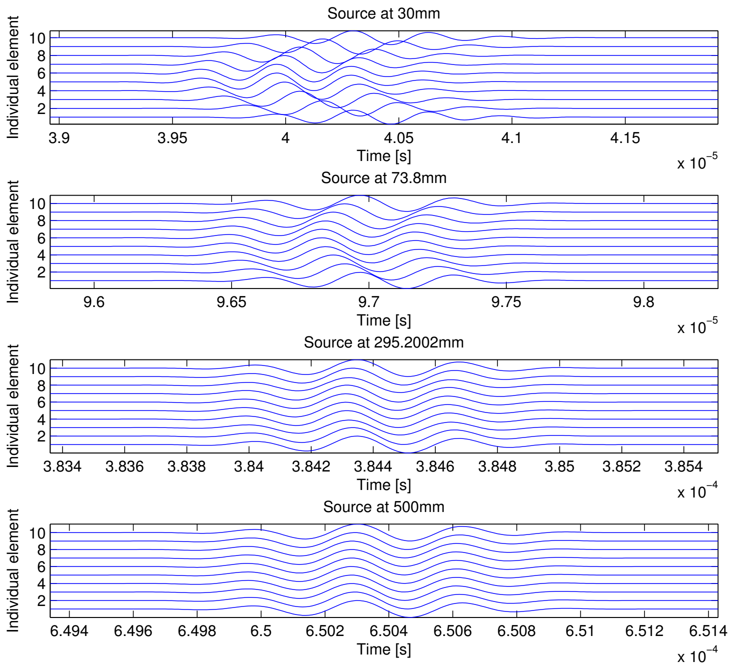


*Figure 2.9: Source placed at 30 mm, 73.8 mm, 295.2 mm and 500 mm for a array with aperture $D = 12.31$ mm. The first source is very near field, while the second is at the softest far field near field limit $R_1 = \frac{D^2}{4\lambda}$, the third at the strictest $R_2 = \frac{D^2}{\lambda}$ and the fourth is far into far field.*

at Figure 2.9 we see that a source placed at $R_1$ is in the second plot, while a source placed at $R_2$ in the third plot. From these two plots we clearly see that the wave from $R_1$ can barely be called plane while the source at $R_2$ is very close to being plane.

The reason for this digression on far field and near field is because we need to take this into account when finding the delays for the DAS beamformer. For a far field source we can assume that the direction of propagation $\xi$ is equal for all sensors in the array, while for a near field source the direction of propagation varies between the elements in the array, giving a $\xi_m$ for every element. A nice illustration of this can be seen in Figure 2.10. The error we allow between $\xi$ and $\xi_m$ indicates where the far field limit is.

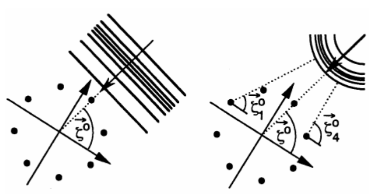


*Figure 2.10: In far field, left, all element have the same direction of propagation $\xi$, while in near field, right, the $\xi_m$ varies between the elements. (Johnson and Dudgeon, 1993, p.115)*

### Delay-and-sum in medical ultrasound imaging

The array described in Section 2.3.1 is a typical array used for ultrasound imaging, except that in ultrasound there are typically much more elements. If we for example have 48 elements we get $D \approx 60$ mm giving the softest far field limit $R = \frac{D^2}{4\lambda} = 1757$ mm, so ultrasound is definitely near field. This means that we need to use a technique called dynamic focusing, delaying the signals from the more central elements with respect to those from the outer most elements, this focus is automatically and dynamically advanced to match the depth of origin of echoes (Whittingham, 2007). This will *straighten* the spherical waves recorded before summing them. A nice intuitive illustration of this can be seen in Figure 2.11.

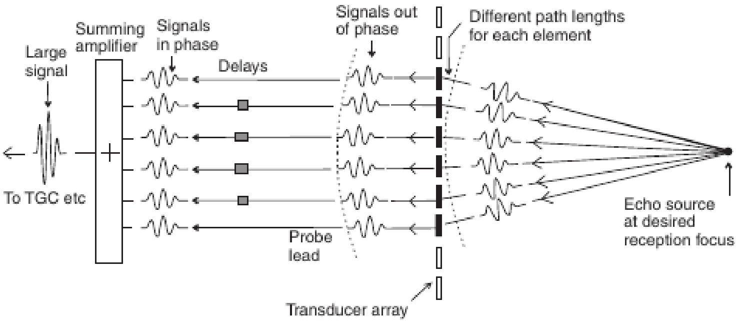


*Figure 2.11: "Focusing in reception. (...) For a particular receive focus position, the signal from all elements can be made to arrive at the same time at a summing amplifier by having an appropriate electronic delay in each channel." Whittingham (2007).*

## 2.3.2 Adaptive beamforming

While the conventional beamforming offers fast and simple implementation with fairly good results, research in the field of array signal processing has given us many new and more sophisticated adaptive beamforming techniques (Krim and Viberg, 1996). These techniques are adaptive because they take advantage of the characteristics of the observations, and seek to adapt the computation of the beamformer output to these observations. This might give much better signal processing performance than the conventional techniques, but are less robust regarding e.g. coherent signals, signals that are delayed and scaled versions of each other (Johnson and Dudgeon, 1993, p. 349-350). In this thesis we focus on one adaptive beamforming technique; the Capon (minimum variance) beamformer first introduced by Bryn (1962), but is better known from Capon (1969). A more intuitive description of Capon's beamformer is given in e.g. (Synnevåg et al., 2007a) or (Johnson and Dudgeon, 1993).

### The spatial covariance matrix

Central in Capon's beamformer is the spatial covariance matrix. This is easiest derived by examining the power of a signal. If we keep the delay-

and-sum on vector form from 2.3.1 in mind we can find the power of the output as

$$\begin{aligned}P(z(t)) &= E\{|z(t)|^2\} = E\{\mathbf{w}^H\mathbf{Y}(\mathbf{w}^H\mathbf{Y})^H\} = E\{\mathbf{w}^H\mathbf{Y}\mathbf{Y}^H\mathbf{w}\}\\ &= \mathbf{w}^H E\{\mathbf{Y}\mathbf{Y}^H\}\mathbf{w} = \mathbf{w}^H\mathbf{R}\mathbf{w}.\end{aligned}$$

From this we get the spatial covariance matrix $\mathbf{R} = E\{\mathbf{Y}\mathbf{Y}^H\}$ where $\mathbf{Y}$ is the delayed received signals. With the covariance matrix in hand Capon's method tries to minimize the variance of the power ($E\{|z(t)|^2\}$) while maintaining gain equal to one in the direction we are steering. This optimization problem can be formulated as

$$\underset{\mathbf{w}}{\text{minimize}} \;\; P(z(t) = E\{|z(t)|^2\} = \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{2.4}$$

$$\text{subject to} \;\; \mathbf{w^H a = 1}. \tag{2.5}$$

Where $\mathbf{a}$ is the steering vector, further explained in a later section. The solution to the optimization problem is

$$\mathbf{w} = \frac{\mathbf{R}^{-1}\mathbf{a}}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}. \tag{2.6}$$

So the result from Capon's beamformer is weights suppressing unwanted signals and noise while focusing in one direction.

### Estimating the Spatial Covariance Matrix

An estimate of the covariance matrix $\mathbf{R}$ is the sample covariance matrix, where we now assume that we have a sampled version of $\mathbf{y}$ and $\mathbf{Y}$:

$$\widehat{\mathbf{R}} = \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{y}[n]\mathbf{y}^H[n] = \frac{\mathbf{Y}\mathbf{Y}^H}{N} \left| \begin{array}{r l} N: & \text{Number of samples}\\ \mathbf{y}[n]: & \text{Vect of sample } n \text{ from } M \text{ elmnts.}\\ \mathbf{Y}: & \left[\mathbf{y}[0]\;\mathbf{y}[1]\;\ldots\;\mathbf{y}[N-1]\right].\end{array}\right. \tag{2.7}$$

### Challenges with Capon's Beamformer

As mentioned earlier the adaptive beamformers are often less robust than the conventional. One particular problem is coherent signals. Coherent signals are signals that are delayed and scaled versions of each other. Since Capon's beamformer tries to minimize the power, letting the coherent signals cancel each other might seem like a good solution to the minimization problem, but actually removing the signal is clearly not a good idea. To cope with signal coherence it is common to average the spatial correlation matrix in space (Tie-Jun et al., 1985). Averaging in space is known as subarray averaging, and can be viewed as in Figure 2.12, dividing the spatial covariance matrix into $L$ submatrices and average the submatrices into one smaller spatial covariance matrix. This corresponds to dividing the array into $L$ smaller subarrays creating one covariance matrix for each subarray and average them into one.

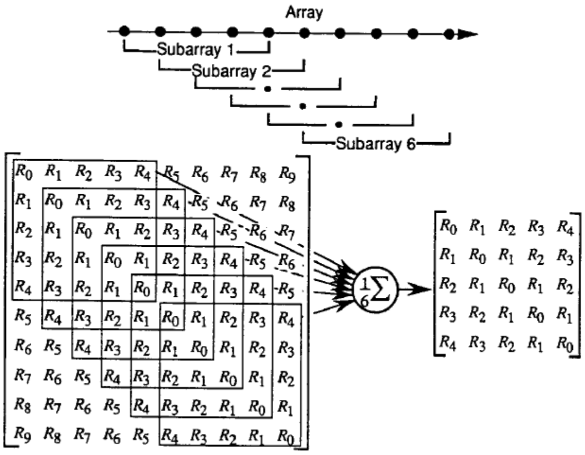

Array
Subarray 1
Subarray 2
Subarray 6
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9
R1 R0 R1 R2 R3 R4 R5 R6 R7 R8
R2 R1 R0 R1 R2 R3 R4 R5 R6 R7
R3 R2 R1 R0 R1 R2 R3 R4 R5 R6
R4 R3 R2 R1 R0 R1 R2 R3 R4 R5
R5 R4 R3 R2 R1 R0 R1 R2 R3 R4
R6 R5 R4 R3 R2 R1 R0 R1 R2 R3
R7 R6 R5 R4 R3 R2 R1 R0 R1 R2
R8 R7 R6 R5 R4 R3 R2 R1 R0 R1
R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
1/6 Σ
R0 R1 R2 R3 R4
R1 R0 R1 R2 R3
R2 R1 R0 R1 R2
R3 R2 R1 R0 R1
R4 R3 R2 R1 R0

*Figure 2.12: Subarray averaging with $L = 6$ (Johnson and Dudgeon, 1993, p. 188).*

A nice mathematical description (from (Synnevåg et al., 2009)) of subarray averaging is:

$$\hat{\mathbf{R}} = \frac{1}{N+M-L+1} \sum_{n=0}^{N-1} \sum_{l=0}^{M-L} \overline{\mathbf{y}}_l[n] \overline{\mathbf{y}_l}^H[n] \Bigg| \quad \overline{\mathbf{y}}_l[n] = \begin{bmatrix} y_l[n] \\ y_{l+1}[n] \\ \vdots \\ y_{l+L-1}[n] \end{bmatrix}. \tag{2.8}$$

As stated earlier, Capon's beamformer places a weight of one in the direction we are steering, while suppressing signals from other directions. This also means that the beamformer is very sensitive to wrong assumptions about e.q. acoustic velocity. If we steer slightly in the wrong direction the performance of Capon's beamformer might actually be worse than the conventional approach (Li et al., 2003). A second technique used to increase the robustness of Capon's beamformer is diagonal loading. This means adding a constant $\epsilon$ to the diagonal of the covariance matrix before evaluating the weights, Equation (2.6). Mathematically $\widehat{\mathbf{R}}_{\text{diagonal loaded}} = \widehat{\mathbf{R}} + \epsilon \mathbf{I}$.

When increasing the robustness of Capon's beamformer, what we actually do is making it more similar to a DAS beamformer with constant weights. We can see this by acknowledging that by choosing a large $\epsilon$ then $\widehat{\mathbf{R}}_{\text{diagonal loaded}} = \widehat{\mathbf{R}} + \epsilon \mathbf{I} \approx \mathbf{I}$. Giving us, from Equation (2.6),

$$\mathbf{w} = \frac{\widehat{\mathbf{R}}^{-1}_{\text{diagonal loaded}} \mathbf{a}}{\mathbf{a}^H \widehat{\mathbf{R}}^{-1}_{\text{diagonal loaded}} \mathbf{a}} \approx \frac{\mathbf{I}^{-1}\mathbf{a}}{\mathbf{a}^H \mathbf{I}^{-1} \mathbf{a}} = \frac{\mathbf{a}}{\mathbf{a}^H \mathbf{a}} = \frac{\mathbf{a}}{||\mathbf{a}||^2}.$$

So the weights become a scaled version of the steering vector, and thus a scaled version of DAS with constant weights.

Subarray averaging gives us a $\mathbf{R}$ of dimension $L \times L$, this means that the weights from Equation (2.6) will be of dimension $L \times 1$. So to get the output, $\hat{z}$, from the beamformer we can use the weights and average over the $L$ subarrays of elements (Synnevåg et al., 2007a), mathematically

$$\hat{z}[n] = \frac{1}{M-L+1}\sum_{l=0}^{M-L}\mathbf{w}^H[n]\overline{\mathbf{y}}_l[n]. \tag{2.9}$$

This is known as the amplitude Capon, in contrast to the so-called power Capon where the output is found by averaging the individual subarrays. The length of the subarrays, $L$, is an important factor in Capon's beamformer. The shorter we choose the length, the more similar Capon's beamformer becomes to DAS. If we choose $L = 1$ the weights from Equation (2.6) will be of dimension $1 \times 1$, thus a constant $c$, and Equation (2.9) simplifies to

$$\hat{z}[n]_{L=1} = \frac{1}{M}c[n]\sum_{l=0}^{M-1}\mathbf{y}_l[n],$$

which again is simply DAS with constant weights.

In the other end of the scale, we can increase $L$ too much and risk that the spatial covariance matrix becomes singular and not invertible. Therefore we use a upper limit of $L <= M/2$ to be sure that $\widehat{\mathbf{R}}$ has full rank and is invertible (Synnevåg et al., 2007a). Choosing $L$ is a tradeoff between performance and robustness.

### Capon's beamformer in medical ultrasound imaging

As we stated in Section 2.3.1 we need to take into account that we are operating in the near field, also when we are using Capon's beamformer in medical ultrasound imaging. That means that we need to do the same dynamic focusing when receiving the signals. We have not paid much attention to the steering vector **a** introduced in Equation (2.6). This is because the steering vector in ultrasound simply becomes a vector of ones since we already have steered in the direction we want using dynamic focusing.

When using Capon's beamformer in ultrasound we meet another challenge. The central element of Capon's beamformer is the spatial covariance matrix **R**. **R** represents the measured *field*. Since ultrasound imaging uses transmitted pulses that are short and non stationary the field is rapidly changing with time (Synnevåg et al., 2007a). This means that **R** should be calculated from a single or only a few temporal samples, changing Equation (2.8) to

$$\hat{\mathbf{R}}[n] = \frac{1}{N+M-L+1}\sum_{n=-K}^{K}\sum_{l=0}^{M-L}\overline{\mathbf{y}}_l[n]\overline{\mathbf{y}_l}^H[n]\Bigg| \quad \overline{\mathbf{y}}_l[n] = \begin{bmatrix} y_l[n] \\ y_{l+1}[n] \\ \vdots \\ y_{l+L-1}[n] \end{bmatrix}. \tag{2.10}$$

This changes the temporal averaging to be over $2K + 1$ samples, instead of all the samples as earlier. The temporal averaging, not just one sample, comes from observations in Synnevåg et al. (2007b) that $K = 0$ did not capture the statistics of a speckle process. So to get the same speckle statistics as DAS, temporal averaging was introduced. Even though we average over $2K + 1$ samples to create the $\mathbf{R}$ matrix, we still only apply the weights for one time sample. This means that each sample is used many times for different estimations of $\mathbf{R}$, but in different combinations with other time samples.

Using Capon's beamformer is very computationally expensive since we need to calculate a $\hat{\mathbf{R}}[n]$ for every sample at every line. The heavy computation load is the main obstacle in using Capon's beamformer for real time medical ultrasound imaging. A solution could be to do the computations on a GPU as done by Åsen et al. (2014b).

The implementation of the Capon beamformer used in this thesis is given in Appendix A.

## 2.4 Summary

In this chapter we have given a brief introduction to the physical phenomena behind ultrasound imaging and also given a simple illustration on how linear ultrasound images are created. We have also been introduced to elastography, both static and shear wave, and seen that this is a promising and harmless technology for more accurate medical diagnosis. The shear wave elastography is very promising and interesting, but is beyond the scope of this thesis. We will pick up the static elastography in Chapter 5, where we will dig deeper into the details and also simulate different elastography setups.

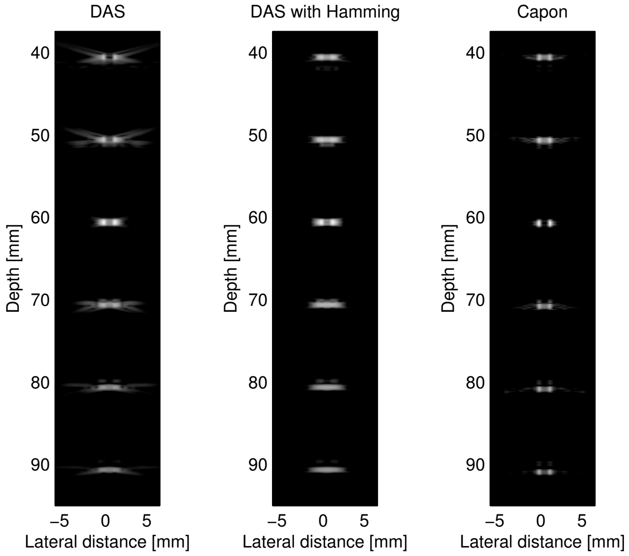


*Figure 2.13: Ultrasound images of a 12-point phantom simulated with a 57.2 mm 124 element array, using 96 active elements in Field II. The image to the left is created with DAS beamforming using uniform weights, the image in the middle is DAS with Hamming weights and the image to the right is created with Capon's beamformer with $K = 1$ and $L = \frac{M}{2} = 48$.*

The fascinating technique of beamforming has been introduced and we have looked at two different techniques of beamforming, conventional delay-and-sum (DAS) and the adaptive Capon's beamformer and seen how this can be used for ultrasound imaging.

To demonstrate a comparison between the two beamformers we have simulated ultrasound images of a 12-point phantom using Field II (Jensen, 1996a)(Jensen and Svendsen, 1992) in Figure 2.13. The images were created with the DAS beamformer with a rectangular window, a Hamming window and by using Capon's beamformer. From this figure we see that the resolution achieved with Capon's beamformer is superior that of the DAS beamformer. The details will be further discussed in the next chapter together with further details and hopefully a better intuition of Capon's beamformer. We will also see that Capon's beamformer creates a more distinct and smaller speckle pattern, and in the later chapters we will investigate if this different pattern has any benefits when we do static elastography.

# Chapter 3

# Simulating ultrasound images

***Chapter abstract:*** *In the first part of this chapter we will have a thorough theoretical investigation of the ultrasound probe we will use in the simulations. Especially we will define the resolution in all dimensions, the resolution cell and the two-way resolution. The spacing of the ultrasound beams in the image will also be defined, and we will introduce a lateral oversampling factor for the beam spacing. In the second part we simulate ultrasound images of point scatterers, and thoroughly investigate the effect of the Capon beamformer with and without lateral oversampling. The third part digs deeper into the details of the Capon beamformer, by investigating two examples, to hopefully increase our intuition on how it works. In the fourth and final part of the chapter we simulate speckle, discuss well developed speckle, and investigate how the Capon beamformer influences the statistics of speckle. The lateral oversampling factor for Capon beamforming is also investigated for the speckle images. This chapter is pretty heavy, but necessary to make sure we create correct simulations before we compare the performance of the two beamformers applied to static elastography.*

When testing different ultrasound cases and setups there is an advantage to be able to simulate data. This allows us to easily compare different setups and change what is being imaged but still have comparable results. The Field II Simulation Program (Jensen, 1996a)(Jensen and Svendsen, 1992) created by Professor Jørgen Arendt Jensen at the Technical University of Denmark has become the *de facto* standard when simulating ultrasound images. For this thesis many hours have been spent to create a simulating environment in MATLAB using Field II. This environment has allowed to compare different beamformers and different phantoms and has allowed us to easily test different scenarios and setups. In this chapter we will go through the details on how the images have been simulated while we also describe and discuss some theoretical details and the results of our simulations.

## 3.1 The ultrasound probe

In all the simulations we are simulating a 7.5 MHz 192 element linear array transducer with 128 active elements, from now called “the probe”. The probe specs are similar to a commercial linear array from the mid 1990's. The probe has a long aperture of 40 mm (39.93 mm) and an element pitch of $d = \frac{39.93mm}{192element} = 0.208$ mm. The kerf, the *cut* between the elements, is 50 microns (0.05 mm) giving us an element width of 0.1580 mm. The element height is 4 mm. For simplicity the probe specs and other important constants we will later use are summarized in Table 3.1.

| **Ultrasound Probe Specs** | |
|---|---|
| Number of elements | 192 |
| Active elements ($M$) | 128 |
| Aperture ($D_{full}$) | 39.93 mm |
| Active aperture ($D$) | 26.62 mm |
| Element width ($d_e$) | 0.1580 mm |
| Element height ($h$) | 4 mm |
| Element pitch ($d$) | 0.208 mm |
| Kerf | 0.05 mm |
| Center frequency ($f_c$) | 7.5 MHz |
| Speed of sound in tissue ($c$) | 1540 m/s |

*Table 3.1: Simulated ultrasound probe specs and other important constants.*

### 3.1.1 Aperture smoothing function

To investigate how the different specs influence the performance of the probe it is nice to create the aperture smoothing function. The aperture smoothing function is given for a linear aperture as (Johnson and Dudgeon, 1993, Chapter 3)

$$W(\mathbf{k}) = \frac{\sin(k_x d_e/2)}{k_x/2} \quad \Bigg| \quad \begin{aligned} \mathbf{k} &: \text{wavenumber vector} \\ d_e &: \text{element width.} \end{aligned} \tag{3.1}$$

In our case this is the beampattern for one element. The discrete smoothing aperture function, describing an array of elements, is given as the DFT (Discreet Fourier Transform) of the weights on the elements

$$W(\mathbf{k}) = \sum_{m=0}^{M-1} w_m e^{j\mathbf{k}\mathbf{x_m}} \quad \Bigg| \quad \begin{aligned} M &: \text{Number of active elements} \\ k &: \text{wavenumber vector} \\ w_m &: \text{weight for element } m \\ x_m &: \text{position for element } m. \end{aligned} \tag{3.2}$$

In the discrete smoothing aperture function we are assuming that the elements are infinitely small only occupying a single point in space. To get the total aperture smoothing function we need to combine the two aperture smoothing functions, giving us $W_{total} = W_{array} W_{element}$.

If we assume uniform weights and the specks from Table 3.1 the aperture smoothing function for our probe is as plotted in Figure 3.1.

The red vertical lines in the top plot in Figure 3.1 are indicating the *visible region* of the array. This region is given by $\pm\frac{2\pi}{\lambda}$. We see from the

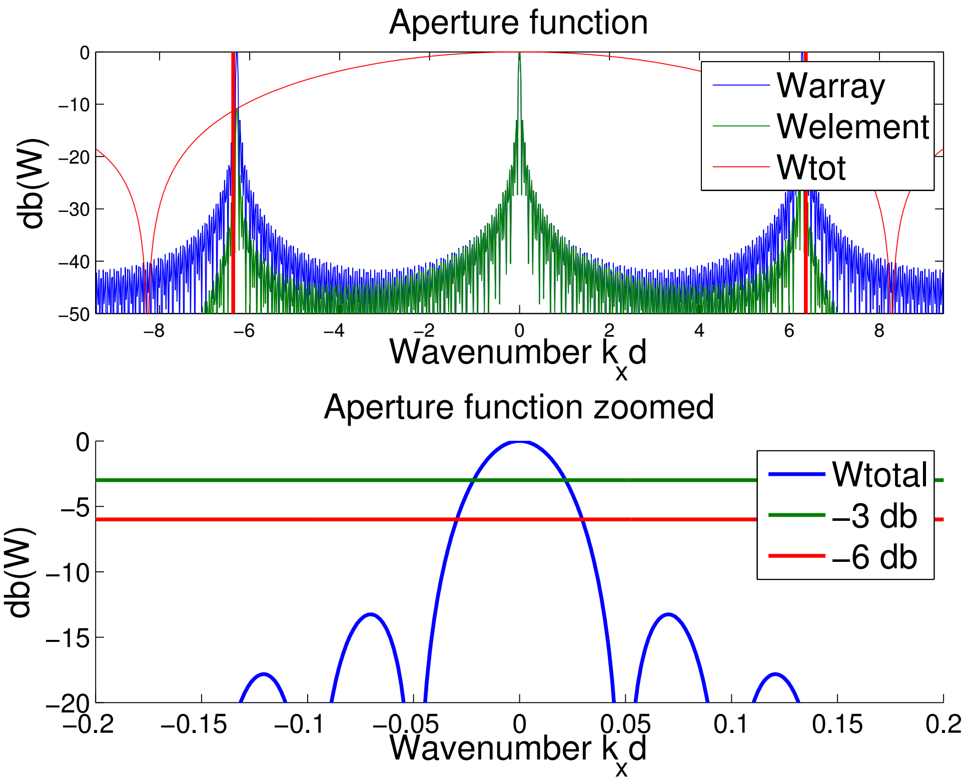


*Figure 3.1: Aperture function of the probe we are simulating. The probe specs are in Table 3.1.*

figure that the grating lobes are in fact inside the visible region. This indicates that we could get spatial aliasing when using 7.5 MHz as central frequency, in other words we do not fulfill the spatial Nyquist $d \leq \frac{\lambda}{2}$, our probe actually uses $d \approx \lambda$. However, since this probe is intended to do linear imaging the steering angle is usually small. This is a pulsed wave system, not continuous wave, so the pulses will only align constructively in focus, the main lobe. The pulses originating from the grating lobes will not align in time and thus be suppressed by the beamformer. This aperture function assumes 7.5 MHz as frequency, so the grating lobes will be different for different frequencies. For lower frequencies the grating lobes will be outside the visible region, while for higher frequencies the grating lobes will be at different angles inside the visible region. We will also later see that ultrasound imaging is a two-way system which gives even more suppressed sidelobes and grating lobes. We will therefore allow the somewhat high grating lobes.

### 3.1.2 Resolution [1]

The resolution of an array is the ability two separate objects along the dimension of interest. An array with good (small) resolution can resolve two objects at a shorter distance than an array with worse resolution. In ultrasound we have three dimensions; lateral (across beam), axial (along beam) and elevational (into beam). There is no exact definition of resolution, as there exist many different resolution criteria. A soft resolution criterion is the full width of the mainlobe while a very strict is the Sparrow resolution limit, defined as the closest separation between two objects that can still be perceived as separate by an observer, this will be defined in Section 3.3.2. The most common resolution criterion is defined by Lord Rayleigh and is defined as the limit where there is a 3 dB drop in amplitude between objects. We will look further into these different criterions in the next sections.

#### Lateral resolution

The lateral resolution is given by the aperture smoothing function of the array, see Figure 3.1. The width of the mainlobe defines how well we can separate two objects in the lateral dimension. The resolution is given as an angular resolution and is thus dependent on range. The relation between the angle and the wavenumber vector, $\mathbf{k}$, is given by $\theta = \sin^{-1}(\frac{\lambda \mathbf{k_x}}{2\pi})$. This follows from decomposing $\mathbf{k}$ and knowing that $|\mathbf{k}| = \frac{2\pi}{\lambda}$. From this formula and by reading from the plot we can find the angle for the different resolution criterion. The -3 dB (Rayleigh) and -6 dB (FWHM, full with half maximum) resolution is at $k_x d = 0.0215, \theta_{-3\text{dB}} = \sin^{-1}(\frac{\lambda k_x}{2\pi}) =$ 0.0068 radians , and at $k_x d = 0.0295, \theta_{-6\text{dB}} = \sin^{-1}(\frac{\lambda k_x}{2\pi}) = 0.0093$ radians. This also agrees with the approximation formulas (Harris, 1978)

$$\theta_{-3\text{dB}} \approx \frac{0.89\lambda}{D} = 0.0069 = 0.3953^\circ \tag{3.3}$$

and

$$\theta_{-6\text{dB}} \approx \frac{1.21\lambda}{D} = 0.0093 = 0.5317^\circ. \tag{3.4}$$

This is the angular resolution while the actual lateral resolution can be found by $\tan(\frac{\theta}{2}) = \frac{x/2}{R}$ which simplifies to $x = R\theta$ with $x$ being the lateral distance and $R$ is the range, if we use small angle approximation.

#### Axial resolution

The axial resolution is given by

$$\Delta r = \frac{c\tau}{2} = \frac{c}{2B} \approx \frac{c}{2 \times 0.5 f_c} = \frac{1540}{2 \times 0.5 \times 7.5 \times 10^6} = 0.2053 \text{ mm}. \tag{3.5}$$

[1] The resolution discussed here is the resolution in far field, see Section 2.3.1, but it can be shown that the resolution in focused near field is the same as the far field resolution (Steinberg, 1976, p. 36).

This is derived from the properties that the bandwidth of a pulse is inverse proportional to the length of the pulse. E.g. a *sinc* in time domain is a square in frequency domain and vice versa. The approximation made is that the bandwidth is ±50 % of the center frequency, which agrees with investigations of the frequency content in our later simulations. We can observe that for our case the axial resolution $\Delta r$ is actually equal to $\lambda$.

#### Elevation resolution

The elevation resolution, the dimension perpendicular to the axial and lateral dimension, is given by the height of the elements in the probe. Our probe has elements with height 4 mm giving an elevation resolution of

$$\theta_{\text{elevation -3dB}} \approx \frac{0.89\lambda_c}{h} = 0.0457 \text{ radians} = 2.6177^\circ. \tag{3.6}$$

### 3.1.3 Resolution cell

If we combine the resolution in all dimensions, Equations (3.3), (3.6) and (3.5), we end up with a volume defining the so called resolution cell of the system:

$$V_{\text{resolution cell}} = \theta_{-3dB\ lateral} R \times \theta_{elevation} R \times \Delta r. \tag{3.7}$$

### 3.1.4 Two way resolution

So far we have only seen half the truth. Ultrasound imaging is a so called two-way system meaning that it both transmits and receives signals. This affects the effective aperture. The one-way beam pattern was as in Equation (3.2) the DFT of the aperture function. In our system we use the same aperture for both transmit and receive, so we have $w_m = w_{m_t} = w_{m_r}$, thus giving us $W_t = W_r = DFT[w_m]$. The two-way aperture function will thus be

$$W_{\text{two-way}} = W_t W_r = DFT[w_m \otimes w_m]. \tag{3.8}$$

If we also throw the element response into the mix we get the two-way aperture function as plotted in Figure 3.2.

If we inspect the plot in Figure 3.2 we can find that the relation between the one-way angular resolution at -3 dB, $\theta_{-3\text{dB}}$, and the two-way angular resolution, $\alpha_{-3\text{dB}}$, is

$$\frac{\theta_{-3\text{dB}}}{\alpha_{-3\text{dB}}} \approx \sqrt{2}. \tag{3.9}$$

We can also follow the more theoretical arguments from (Hergum et al., 2007), which states that for a rectangular aperture the Nyquist sampling requirements for beam spacing is found from the Fraunhofer

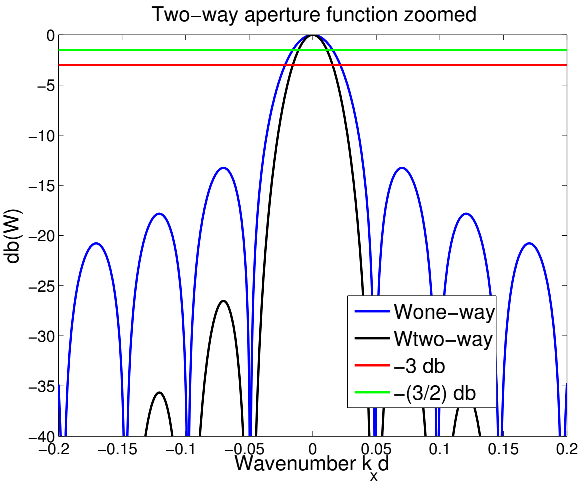


*Figure 3.2: Two way aperture smoothing function of the ultrasound probe we are simulating.*

approximation to be one beam per $\lambda f_{\#}$, the Rayleigh criterion for beam spacing. The f-number is defined as

$$f_{\#} = R/D \qquad \left| \qquad \begin{array}{l} R : \text{Range or depth in the image} \\ D : \text{Size of aperture.} \end{array} \right. \tag{3.10}$$

The lateral distance between the beams, if we only regard the one-way resolution, is

$$\Delta_{\text{one-way}} = \lambda f\# = \frac{\lambda R}{D}. \tag{3.11}$$

Then, from the Fraunhofer approximation and the Equation (3.8) the two-way array pattern includes a convolution of the transmit and receive aperture, and thus gives us the approximated two-way lateral beam distance as

$$\Delta_{\text{two-way}} = \frac{\lambda R}{D_{tr} + D_{rx}} \overbrace{=}^{D_{tr}=D_{rx}} \frac{\lambda R}{2D}. \tag{3.12}$$

The factor two, instead of the $\sqrt{2}$ as observed, comes from the assumption that the two-way response gives a twice as long aperture, but this is only partially true. When we convolve two rectangles of $w_{m_t}$ and $w_{m_r}$ we get a triangle, see Figure 3.3.

This is why the two-way array pattern is a $sinc^2$, but it also means that it's a simplification to say that the two-way aperture is twice as long. The two-way aperture is twice as long, but it is not rectangular. The two-way aperture is a triangle so the weights decrease towards the sides.

However, we will stick with this simplification and use a factor 2 to find the distance between the beams. This means that we are sampling at a bit higher rate than the critical sampling rate.

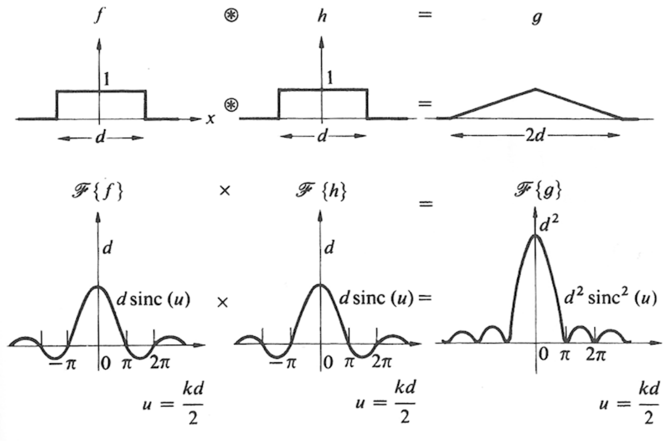


*Figure 3.3: Convolution of two rectangles shown in the time domain (top) and in the frequency domain (bottom).*

### 3.1.5 Image beam spacing and simulation details

In ultrasound B-mode linear array imaging the image is created by *shooting* a beam of sound into the tissue and then sweep the beam sideways creating multiple parallel *lines* of recorded data. These lines are combined into the resulting image. To be sure not to lose information in the image, we need to sample the lines at a certain distance. The standard beam density is given by the Rayleigh criterion for beam spacing as one beam per $\lambda f\#$, as seen in the last section. So the lateral beam displacement is $\Delta = \lambda f\# = \frac{\lambda R}{D}$, where $R$ is the range, often the range of the transmit focus. If we take the two-way resolution into account we get $\Delta = \frac{\lambda R}{2D}$. If we also introduce an oversampling factor $q$, as Åsen et al. (2014a), we get the expression

$$\Delta = \frac{\lambda R}{2qD}. \tag{3.13}$$

This expression and especially the oversampling factor $q$ will be useful in our future discussion.

The first simulations we will create and discuss are simple point scatterer simulations where we are transmitting beams with fixed transmit focus at 60 mm, and use dynamic focus on reception.

For the later speckle simulation we begin our simulation at 27 mm depth. This gives us a $f\# > 1$ given that we use 128 active elements. For simplicity we image a region of 13.3 mm. The motivation for this is that the probe has 192 elements but only 128 active elements, giving us a image width of $(192 - 128) \times d = 13.3$ mm. In *real life* linear probes usually create images as wide as the probe, but then the image towards the sides uses fewer active elements giving a worse resolution at the sides than at the center of the image. To simplify the simulation and later theoretical arguments we only create the part of the image which uses all the 128 active elements giving equal resolution in the center as towards the sides in the simulated image.

In our speckle simulations we are simulating dynamic transmit focus. Since the closest point of focus is at 27 mm, we use this point when calculating the beam spacing. From Equation (3.13) we end up with a distance between the beams when we use no lateral oversampling, $q = 1$ as $\Delta = \frac{\lambda 27mm}{2D}$. If we use $\Delta$ spacing at the 13.3 mm region we get $\frac{13.3\,\text{mm}}{\Delta} \approx$ 128 lines in our later images of speckle. If we use an oversampling factor of $q = 2$ we will get 256 lines for the 13.3 mm regions and so on.

## 3.2 Point scatterer simulation

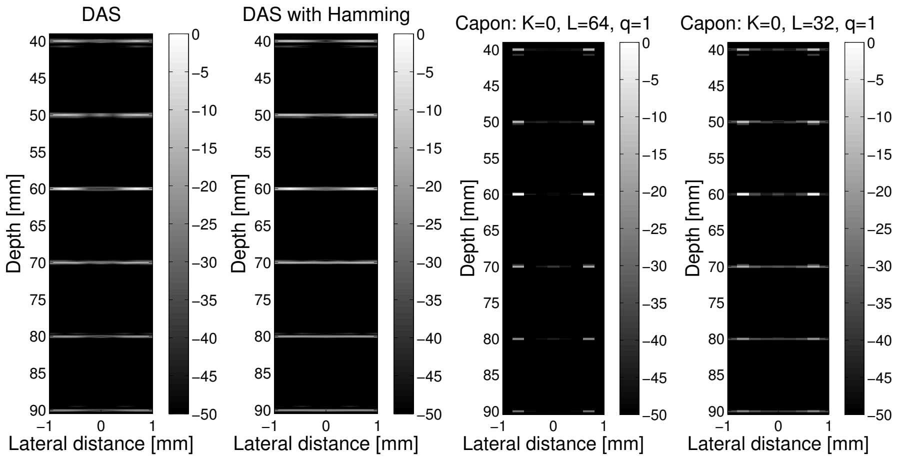
First we create a very simple simulation with 12 point scatterers placed a 6 different depths. This simulation is intended to show the benefits of Capon beamforming in regards to lateral resolution. This is basically a reproduction of some of the most important results from (Synnevåg et al., 2009) and (Synnevåg et al., 2007a).

### 3.2.1 Point scatterer on the scan beam

The point scatterers in the phantom are placed about 1 mm apart laterally, at $\pm 0.52mm$, and separated by 10 mm axially. We have applied dynamic focus on reception but a fixed transmit focus at 60 mm on transmit. The lateral -3 dB resolution at 60 mm is

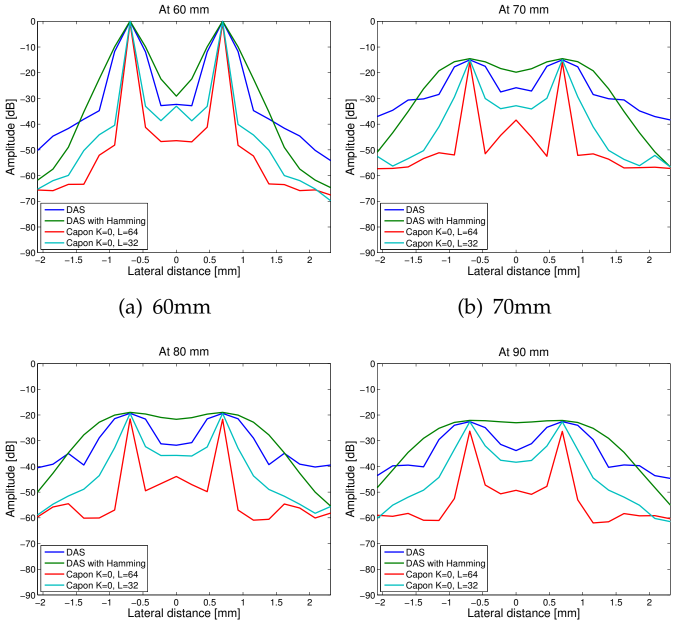
$$x_{-3db} = \theta_{-3\text{db}} R = 0.0068 \times 60 = 0.408\text{mm}$$

so the points should be well separated at focus. We have plotted four different beamformed images of the same simulated RF-data in Figure 3.4. We have beamformed with DAS with uniform weights, DAS with Hamming weights, Capon with $L = M/2 = 64$ and Capon with $L = M/4 = 32$. We see that as expected all beamforming techniques successfully separate the points in focus, but already at 70 mm, 10 mm below focus, the DAS beamforming starts to smear the two points into one. This is not the case for the Capon beamforming which separates the points very well at all depths and demonstrates a lateral resolution superior to DAS. The superior resolution is easily seen by plotting the steered response at 60, 70, 80, and 90 mm. This is done in Figure 3.5.



(a) DAS (b) DAS with Hamming (c) Capon with K=1, L=M/2=64 (d) Capon with K=1, L=M/4=32

*Figure 3.4: Simulation of point scatterers demonstrating the superior lateral resolution of the Capon beamformer.*



(a) 60mm (b) 70mm

(c) 80mm (d) 90mm

*Figure 3.5: Response at different depths for the different beamformers for the simulation of point scatterers. Notice the superior resolution of the Capon beamformer, especially out of focus (focus is at 60 mm), where DAS does not separate the two points.*

### 3.2.2 Point scatterer between the scan beams

The previous simulation of scatterer points was ideal when regarding the position of the points. The points were placed at $\pm 0.52mm$ which is exactly where two of the ultrasound beams focus in the lateral direction. It has recently been pointed out that the signal suppression, or cross frame scalloping loss, is significant if the scan line miss the target (Åsen et al., 2014a). The problem was first addressed by Cox (1973), but most of the work in adaptive beamforming seem to ignore the problem by positioning the scatterers exactly on the beam. Cox estimated the drop in output power between beams, and Åsen et al. graphically showed this in their paper. We have created the same plot for our probe in Figure 3.6[2]. The plotted response shows how dramatically the amplitude of single scatterers will drop if the beam misses the scatterer when using Capon beamforming.

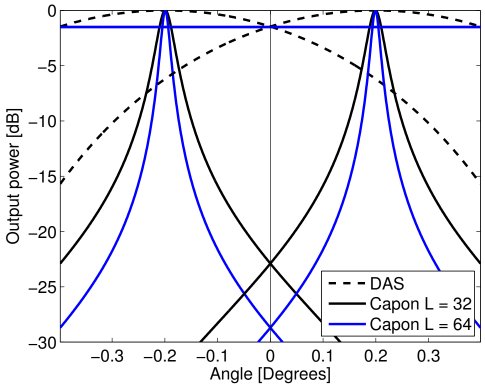


*Figure 3.6: The estimated drop in amplitude midway between the beams for DAS, Capon with $L = M/4 = 32$ and Capon with $L = M/2 = 64$ for our simulated probe.*

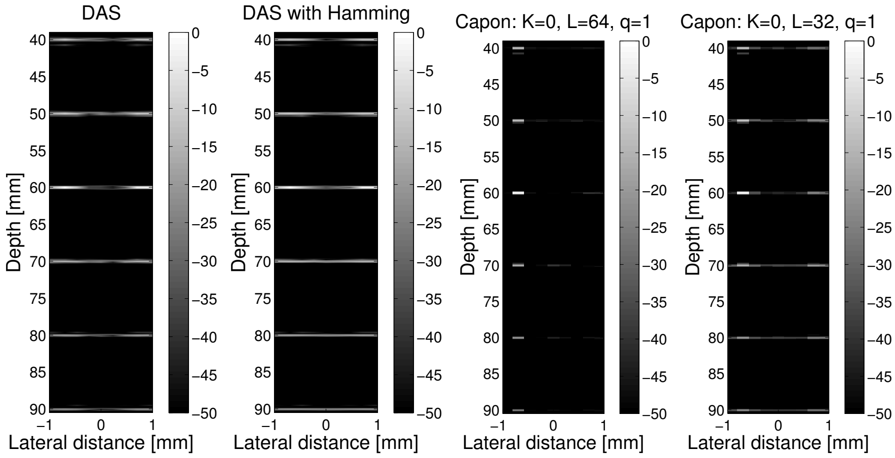
Åsen showed that Capon beamforming introduces a higher shift variance between two image frames. We will show that this is also a problem for single frame scenarios with two points in the same frame. We have simulated a worst case scenario by deliberately placing one point scatterer between two scan beams, the point scatterer is placed at 0.624 mm while the closest scan beams are at 0.520 mm and 0.728 mm laterally. The first point scatterer is placed directly on the scan line at -0.52 mm for reference. The same four beamforming techniques as earlier were used to create the images, and the results are plotted in Figure 3.7 and 3.8.

We see from the plots of the response at the point positions in Figure 3.8 that for DAS we only lost a few dB, actually the Rayleigh -3 dB, in focus when missing the point, and none when we are out of focus. For the most aggressive Capon beamforming, $L = M/2 = 64$, we lost more than 40 dB. In other words the point is lost. Åsen et al. demonstrated that this suppression is a problem for cross frame lateral shift-invariance. This simulation demonstrates that this is in fact also a problem for a single frame ultrasound image. If the effect is not dealt with we might lose information.

If we study the images in Figure 3.7 and the plots in Figure 3.8 more

[2] Acknowledgment goes to Jon Petter Åsen for providing the script for the plot.

closely we see that the problem is not as crucial for the less aggressive Capon beamformer, $L = M/4 = 32$. If we look back at the discussion in 2.3.2 we remember that choosing a smaller subarray, $L$, makes the Capon beamformer more similar to a DAS with constant weights. So trading off performance with robustness is one way of dealing with the problem of signal suppression. Another approach is to introduce lateral oversampling as we will investigate in the next sections.



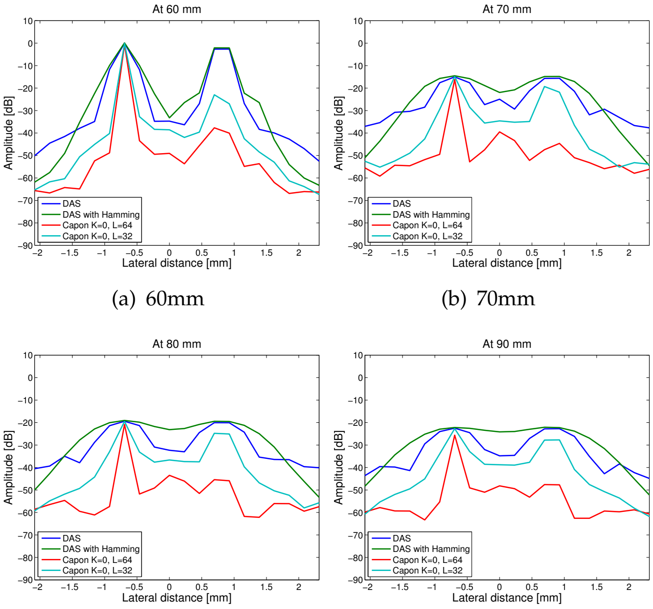
(a) DAS

(b) DAS with Hamming

(c) Capon with K=1, L=64

(d) Capon with K=1, L=32

*Figure 3.7: The same point scatterer simulation as in Figure 3.4, but the point to the right is placed between two scan lines. Notice how the second point almost disappear for the most aggressive Capon beamformer.*



(a) 60mm

(b) 70mm

(c) 80mm

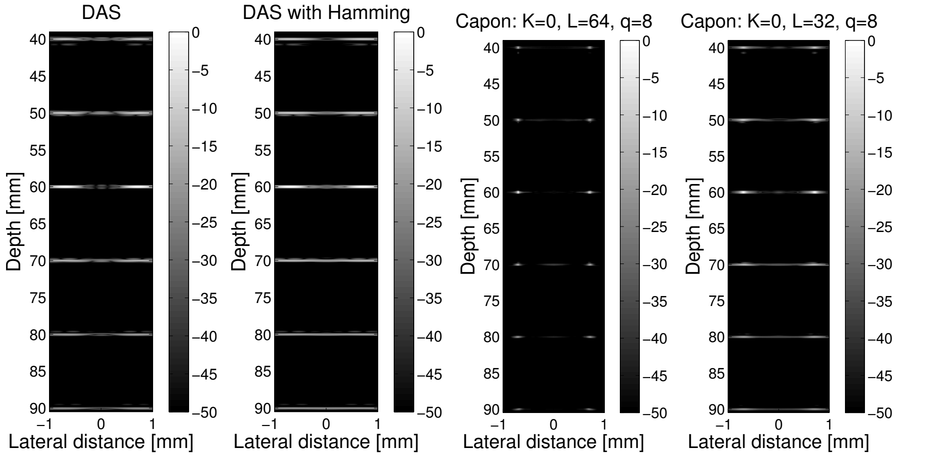
(d) 90mm

*Figure 3.8: Response at different depths for the different beamformers for the simulation of point scatterers. Notice how the Capon beamformer with $L = M/2 = 64$ loses 40 dB at focus, 60 mm. The loss is not as large for Capon with $L = M/4 = 32$.*

### 3.2.3 Point scatterer between the scan beams: laterally oversampled

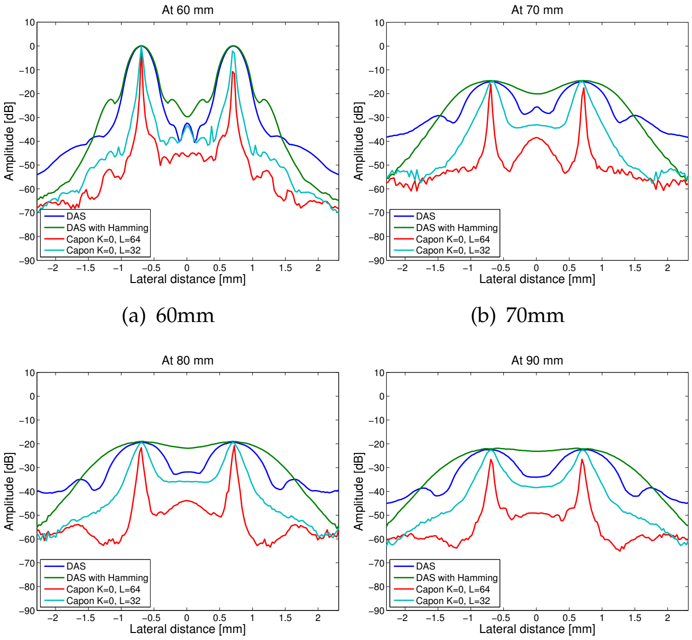
The most straightforward solution to the loss of information is to oversample on transmit (Åsen et al., 2014a). For a linear array this means decreasing the distance between each scan beam. In Figure 3.9 and 3.10 we have simulated the same sized phantom as earlier, but increased the lateral beam sampling with a factor $q = 8$. This means that we have $8 \times 128 = 1024$ beams in the image. In the same fashion as earlier we have deliberately placed the left point scatterer exactly on a transmit beam, while the right one is placed between two beams.

From Figure 3.9 (c) we see that an oversampling factor of 8 brings back the second point for Capon with $L = M/2 = 64$. With a dynamic range of 40 dB it is hard to tell the difference between the points. If we look at the plots in Figure 3.10 we see that we still have about 10 dB loss between the points at focus (a). When we are out of focus the two points have approximately the same amplitude. For the less aggressive Capon beamformer, $L = M/4 = 32$, the two points in focus are only separated by a few dB, so the suppression is not visible in the image.



(a) DAS (b) DAS with Hamming (c) Capon with K=1, L=64 (d) Capon with K=1, L=32

*Figure 3.9: Point scatterer simulation with an oversampling factor $q = 8$. This brings back the second point for the Capon beamformer with $L = M/2 = 64$, but from the plots in Figure 3.10 we see that it is still about 10 dB loss at focus.*



(a) 60mm

(b) 70mm

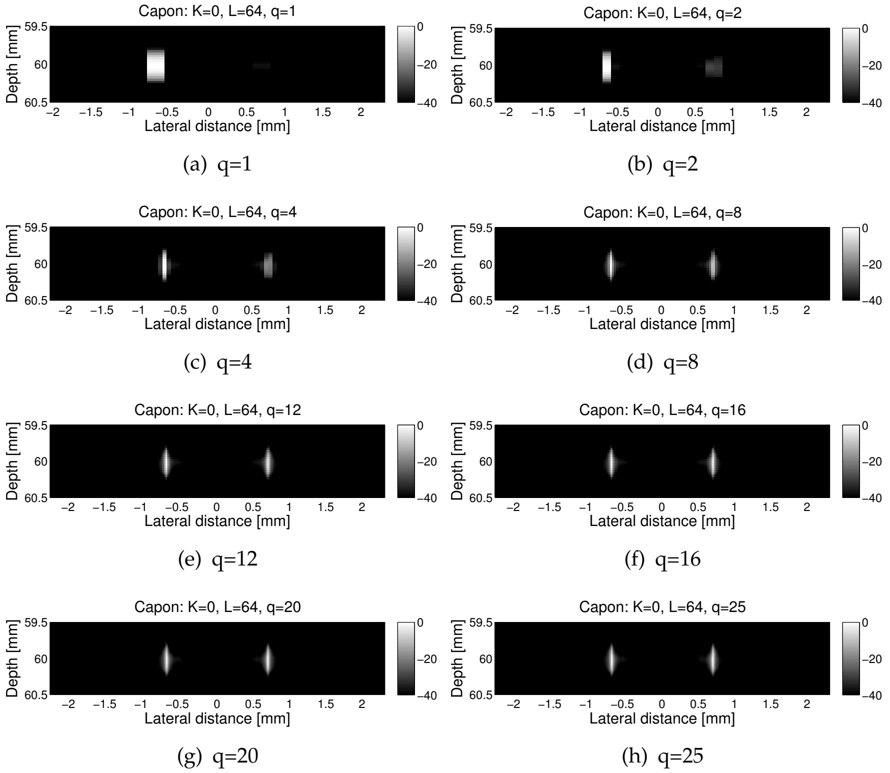
(c) 80mm

(d) 90mm

*Figure 3.10: Response at different depths for the different beamformers for the simulation of point scatterers with an oversampling factor $q = 8$. The Capon beamformer with $L = M/2 = 64$ still loses some amplitude in focus, but out of focus the amplitudes of the two points are about the same.*

### 3.2.4 Lateral oversampled point scatterer in focus

Since it is only at the point of focus that the suppression problem is evident we will further investigate this part of the image. In Figure 3.12 the images at focus, 60 mm depth, is shown for Capon with $L = M/2 = 64$ with oversampling factor, $q$, from 1 to 25. We see that it is only visible difference between the two points until an oversampling factor, $q$, of 12 or maybe 16 if we give it a very close look. If we look at the plots in Figure 3.12 and especially plot (d) where the difference between the point scatterers is plotted against the oversampling factor, we see that for Capon with $L = M/2 = 64$ we need a factor $q = 16$ oversampling before the difference between the points is less that 3 dB. For Capon with $L = M/4 = 32$ it is enough with an oversampling factor of about $q = 8$.



(a) q=1

(b) q=2

(c) q=4

(d) q=8

(e) q=12

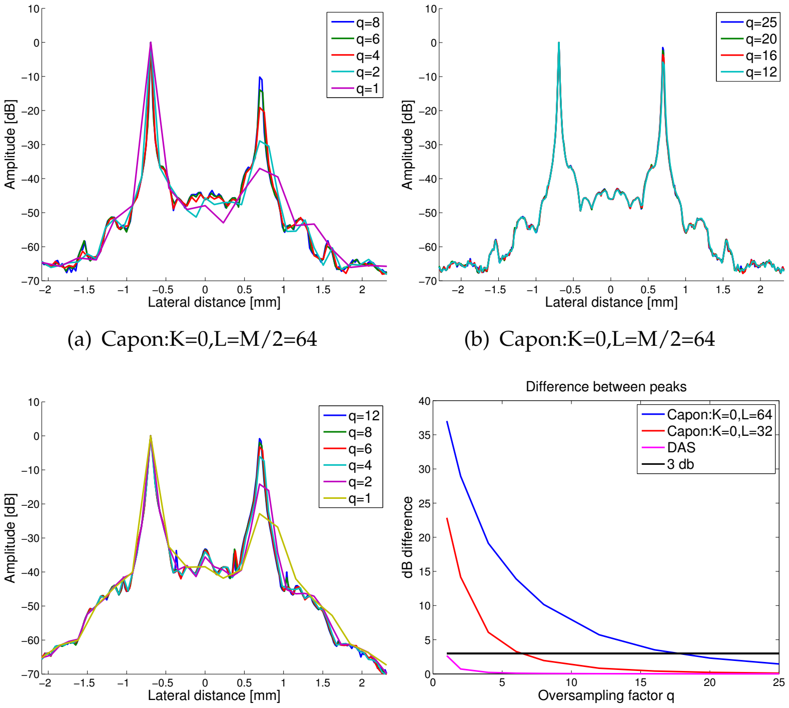
(f) q=16

(g) q=20

(h) q=25

*Figure 3.11: Images of two point scatterers with different oversampling factor, q, at focus (60 mm) where the left point is directly on the beam and the right point is placed between two scan beams. All images created with Capon $K = 0, L = 64$.*

This is different from what Åsen et al. found, and indicates that the oversampling factor needed is dependent on the system, meaning the probe, and what subarray length the Capon beamformer uses.



(a) Capon:K=0,L=M/2=64

(b) Capon:K=0,L=M/2=64

(c) Capon:K=0,L=M/4=32

(d) Difference between peaks vs q

*Figure 3.12: The plotted respons for the points in focus for the Capon beamformer with $L = 64$ in (a) and (b), and the Capon beamformer with $L = 32$ in (c). The difference between the peaks are plotted against the oversampling factor, q, are plotted in (d).*

### 3.2.5 Oversampling methods

So far we have used the simple method of oversampling on transmit. Since we are simulating data this is okay, but oversampling on transmit has many drawbacks. Especially the frame rate will be reduced, proportional to the oversampling factor, making real time ultrasound impossible with an oversampling factor of 16. The computing complexity will also increase by the same oversampling factor (Åsen et al., 2014a). As we have mentioned earlier the computing cost of Capon beamforming is the main obstacle for using it in real time ultrasound. By doing the computation on a GPU (Åsen et al., 2014b) real time Capon is possible, but not with 16 times the data.

Åsen suggested a method to oversample by phase rotation (Åsen et al., 2014a). The method exploits the fact that the steering vector **a** in Capon beamforming is usually set to **1** because the data is pre-delayed on receive. The steering vector can be varied over a set of pre-defined vectors in narrow band applications. Åsen et al. claims that this can also be done in a broad band application as long as the phase rotation is less than one pulse length, so called coarse-fine beamforming. The maximum steering angle, $\Theta_{max}$, is given by the pulse length of

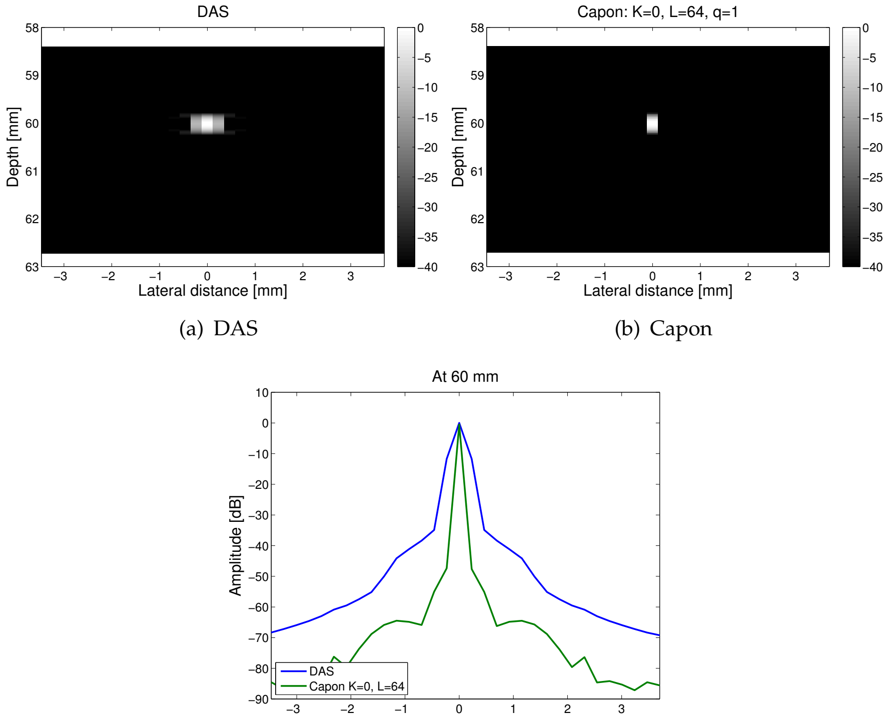
$$\mathbf{a}_\theta = \begin{bmatrix} e^{-j\frac{2\pi}{\lambda_c}x_0 \sin(\theta)} \\ e^{-j\frac{2\pi}{\lambda_c}x_1 \sin(\theta)} \\ \vdots \\ e^{-j\frac{2\pi}{\lambda_c}x_L - 1 \sin(\theta)} \end{bmatrix} \quad \left| \quad \begin{array}{r l} x_i : & \text{Element position} \\ \theta : & \text{Swept from } -\Delta/2 \text{ to } \Delta/2 \\ \lambda_c : & \text{Wavelength of center frequency} \end{array} \right. \tag{3.14}$$

The problem with this approach is that it assumes that we are doing phased imaging, meaning that the transmit beams are displaced by an angle giving a fan shaped image. In elastography we are using linear imaging, and as we know displacing the lines by a given distance. One possible solution to adapt this method to linear imaging could be to create a new steering angle for every point at every line. This is well worth to examine, but is out of scope for this thesis.

## 3.3 Understanding the Capon beamformer

### 3.3.1 Beampattern

We know from our background chapter, Section 2.3.2, that what Capon does is creating different weights for the beamformer dependent on the received data, in other words adapting the weights. If we bring back the simple simulation of one point scatterer and investigate how Capon reacts to this point we might get a better intuition on the *magic* of the Capon beamformer.



(a) DAS

(b) Capon

(c) Response for Capon and DAS

*Figure 3.13: Images of the single point scatterer created with DAS (a) and Capon (b). The response at 60 mm is plotted in (c).*

In Figure 3.13 we have displayed the images of one single point scatterer placed at 60 mm depth. We use a lateral oversampling factor of $q = 1$, but make sure that the point is exactly on the center beam. From the two images we see that when we use Capon beamforming the point scatterer is only visible in one of the lines when we use 40 dB as dynamic range. In the image with the DAS beamformer the point is spread out to the two lines next to the central line. In Figure 3.14 we have plotted the beampattern resulting from both the beamformers when creating the line to the left, straight on and to the right of the point.

If we remember back to Section 2.3.2 the goal of the Capon beamformer is to minimize the variance of the power of the signal while maintaining gain equal to one in the direction we are steering, Equation (2.4). This

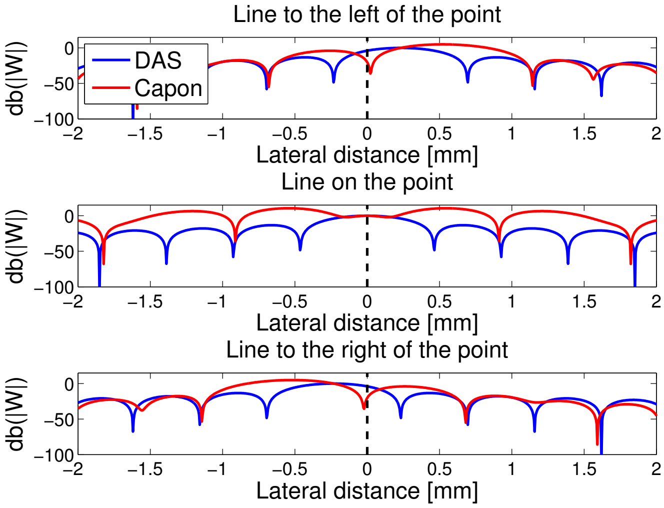


*Figure 3.14: Capon beampattern compared to the DAS beampatten when creating the line to the left, on the point and to the right of the point at 60 mm depth.*

optimization problem results in weights, Equation (2.6), on the elements before the signals are summed. In the top plot in Figure 3.14 we have plotted the beampattern for DAS and Capon when we create the line to the left of the point. We see that the Capon beamformer places a zero at lateral 0 mm and thus suppress the signal from the point scatterer, while the DAS does not have this adaptive capability and will *pick up* most of the signals reflected from the point. In the middle plot the beampattern from the line when hitting the point is plotted. Both the beamformers get all the signals reflected from the point and thus create the bright spot at 0 mm seen in the image. When we once again miss the point on the line right of the point Capon once again places a zero suppressing the point.

It is this adaptive behavior that gives the Capon beamformer better lateral resolution than DAS. The width of the mainlobe will still be approximately the same, but the adaptive behavior allows Capon to adjust the weights so that the mainlobe can be placed to the side of the point.

### 3.3.2 Sparrow's resolution limit

In our discussion so far we have seen that when using the Capon beamformer we need to oversample laterally by a fairly large factor to avoid losing information between the beams. In real time ultrasound the frame rate is very important, and when we have to oversample on the number of beams the frame rate will drop accordingly. The heavy computational cost of Capon beamforming also negatively influence the ability to do real time imaging.

There are also benefits of doing Capon beamforming, most of them looked at in (Synnevåg et al., 2009). The article concludes that Capon beamforming allows the use of reduced transducer size, or increase penetration depth without sacrificing image quality compared with DAS.

We will demonstrate how much the lateral resolution is improved by Capon beamforming. We know from the earlier discussion on resolution that there are many different resolution criterion. One interesting criteria is the Sparrow resolution limit, defined as where the saddle point between the two peaks first develops, thus where the gradient of the summed beampatterns is zero. Usually the Sparrow limit is used in optics, where the diameter of a circular airy disk is in interest. For an airy disk the Rayleigh resolution criteria is given by $\theta_{\text{Airy Rayleigh}} = 1.22\frac{\lambda}{D}$, while the Sparrow resolution is given by $\theta_{\text{Airy Sparrow}} = 0.94\frac{\lambda}{D}$ (Barakat, 1962).

To find the Sparrow limit for our system we have plotted two two-way aperture functions in Figure 3.15, the two two-way aperture functions represents two objects separated so that the sum of the two has a gradient of zero at the peak.

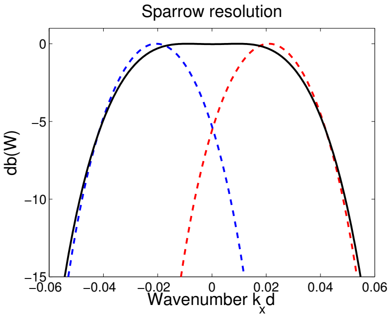
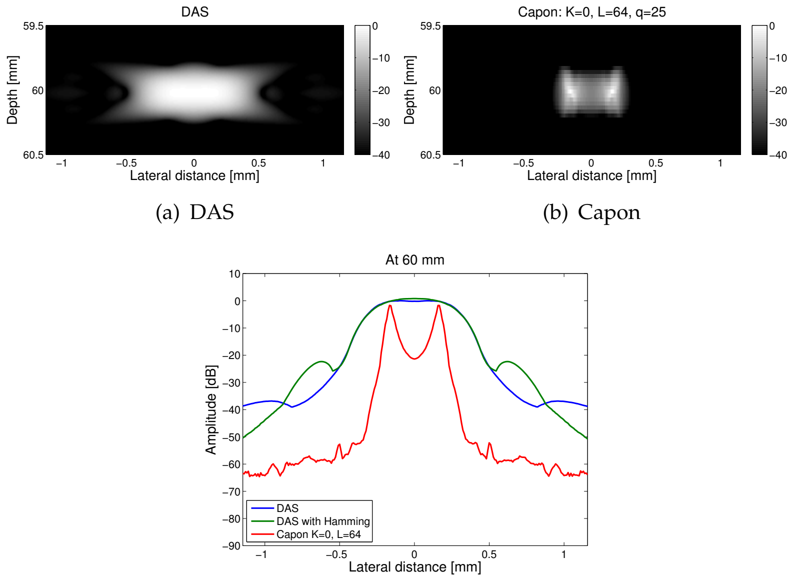


*Figure 3.15: Theoretical sparrow limit. Two aperture functions summing to a flat top.*

From this plot we get that the sources should be separated by approximately 0.38 mm if they are placed at 60 mm depth. This comes from the fact that the angle separating the two peaks is

$$\theta_{\text{sparrow two-way}} \approx \frac{1.2\lambda}{\sqrt{2}D} = \frac{1.68\lambda}{2D}. \tag{3.15}$$

This limit is based on an assumption that the signals reflected are incoherent. Two scatterers in ultrasound will not create incoherent signals, but when we create a simulation with two scatterers at 60 mm depth



(a) DAS

(b) Capon

(c) Resolution plot

*Figure 3.16: Sparrow simulation displaying the superior lateral resolution of Capon compared to DAS.*

laterally separated by 0.38 mm they fit, as we will see, the Sparrow limit fairly well.

In Figure 3.16 we have simulated two points placed at 60 mm depth laterally separated by 0.38 mm. We have oversampled with a factor $q = 25$ and placed the points exactly on the lines to make sure that Capon "hits" the points. From the image of the DAS beamformer we see that we cannot see that it is two points, instead of one wide, since the two points are smeared together, while in the Capon image to the right, with the most aggressive beamformer $L = M/2 = 64$ and $K = 0$, we can clearly see that there are two points. In the bottom plot of the amplitude response at 60 mm we see that both DAS beamformers smears the two points into each other so it appear as one point, while the most aggressive Capon beamformer has separated the two points by almost 20 dB.

This demonstrates the superior lateral resolution of the Capon beamformer and encourages us to further investigate possible benefits of the Capon beamformer. One possible benefit that we will explore in this thesis is the fact that Capon creates a different and more distinct speckle pattern than DAS. In the next sections we will discuss this in depth.

### 3.3.3 Analyzing the RF-data for Capon calculation

In the Field II simulation we use a sampling frequency of 100 MHz. Our center frequency is only 7.5 MHz, so we should be able to decimate quite a lot and still not lose any information. The reason why we use such a high sampling frequency in Field is to get good mathematical approximations. Decimation of the RF-signal will give a huge speedup for Capon since we need to calculate the covariance matrix for every sample.

Prior to Capon beamforming we create the one sided analytic signal of the real RF-signal. The analytic signal is defined as

$$x_a[n] = x[n] + j(\hat{x}[n]) \mid \hat{x}[n]\text{: Hilbert transform of } x[n]. \tag{3.16}$$

see Appendix B for details on the analytic signal and the Hilbert transform. We use the analytic signal since it is one sided, thus complex, and we want a complex signal to allow Capon to calculate complex weights necessary to have a non-symmetric beampattern as we saw in Figure 3.14

In Figure 3.17 we have plotted the power spectrum of the analytic signal from 10 elements prior to Capon beamforming. From the plot we see that is it *safe* to decimate with a factor 4 giving the new sampling frequency at 25 MHz.

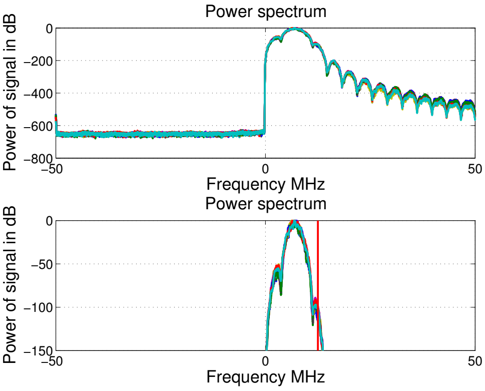


*Figure 3.17: The power spectrum of the analytic signal of 10 elements prior to Capon beamforming.*

To continue the investigation of the signal from the Capon beamformer we have plotted the power spectrum of 10 Capon beamformed lines in Figure 3.18. Luckily the beamformer has not introduced any "negative frequecies" and we can assume that the RF-data post Capon beamforming

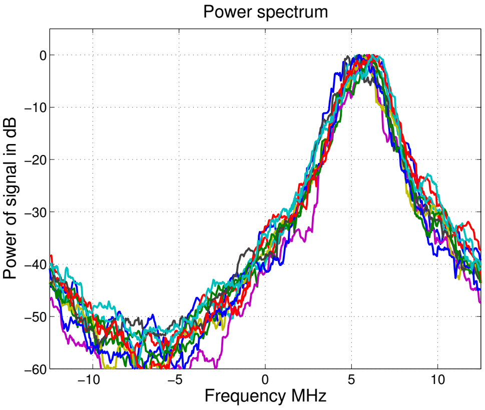


*Figure 3.18: The power spectrum of 10 Capon beamformed lines. We see that we can still assume that we have a one sided analytic signal after Capon beamforming.*

is still analytic. This has some nice aspects, like the fact that we can simply take the absolute value of the beamformed line to get the envelope, see Appendix B for details, and it will also be useful when we start estimating the strain/deformation in tissue in Chapter 5.

## 3.4 Speckle simulation

Homogeneous parts of an ultrasound image has a grainy appearance known as *speckle* (Synnevåg et al., 2007b). Research on ultrasound speckle is a continuation on the research on laser speckle and the first work on ultrasound speckle was published by Burckhardt (1978). In his article he presents speckle as the granular structure appearing when there are many scatterers occupying the same resolution cell. The waves scattered interfere with each other and create the *speckle pattern*. The appearance of speckle seems to be random, but if an object is scanned twice under exactly the same conditions the same identical speckle pattern will occur. However, if the same object is scanned under different conditions e.g. a different probe, the speckle pattern will be different. The fact that the speckle pattern is dependent on the object or tissue being imaged is the fundamental assumption in static elastography where we assume that the speckle pattern will follow tissue movement.

Since the speckle pattern appears to be random, speckle is best described in statistical terms.

### 3.4.1 Speckle statistics

Burckhardt stated that *"If the number of scatterers within one resolution element is large and the phases are distributed uniformly between 0 and $2\pi$, the amplitude $A$ obeys a Rayleigh probability density function"*. The speckle statistics was further investigated by Wagner et al. (1983) and it is now common agreement that *fully developed speckle* in ultrasound imaging is distributed as a Rayleigh distribution which has the PDF

$$f(A;\sigma) = \frac{A}{\beta^2} e^{\frac{-A^2}{2\beta^2}} \text{ with } x \geq 0 \left| \begin{array}{l} A : \text{Amplitude} \\ \beta : \text{Scaling parameter.} \end{array} \right. \tag{3.17}$$

The scaling parameter, $\beta$, can be found since we know that the variance of a Rayleigh distribution is given by $\sigma^2 = \frac{4-\pi}{2}\beta^2$. This gives us the scaling parameter as $\beta = \frac{\sigma}{\sqrt{\frac{4-\pi}{2}}}$.

The signal to noise ratio defined by $SNR = \frac{\mu}{\sigma}$ is 1.91 for a Rayleigh distribution where $\mu$ is mean and $\sigma$ is the standard deviation.

### 3.4.2 Simulated speckle

The statement*fully developed speckle* deserves a closer investigation. Wagner et al. (1983) and Burckhardt (1978) stated that if the number of scatterers within one resolution cell was large, the amplitude was distributed as a Rayleigh distribution. This was further investigated by Wagner et al. (1988), where they stated that the Rayleigh distribution was approached if there were slightly fewer than 10 scatterers per resolution cell. When the literature uses the term resolution cell they are using the one-way resolution cell. Today it is common to state that the speckle is fully developed if it is simulated more than 10 scatteres per resolution cell.

We have simulated speckle by randomly distributing $N$ point scatterers in a $15 \times 18$ mm phantom beginning at the depth of 25 mm and ending at 44 mm. We are actually just using the image from 27 mm to 42 mm to avoid *fuzzy edges*. In the lateral dimension we are simulating a 13.5 mm wide image, see Section 3.1.5, this way we are also avoiding the fuzzy edges on the sides, see Figure 3.19. The fussy edges probably arise because at the bottom there are no scattered waves interfering from beneath, and at the top no waves are scattered to interfere from the top and thus the speckle pattern will be different.

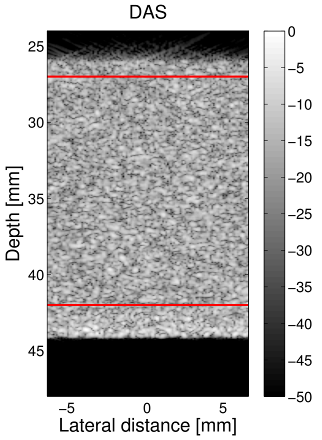


*Figure 3.19: We are just using the image from 27 to 42 mm to avoid the "fussy edges" seen in the top and bottom of this image.*

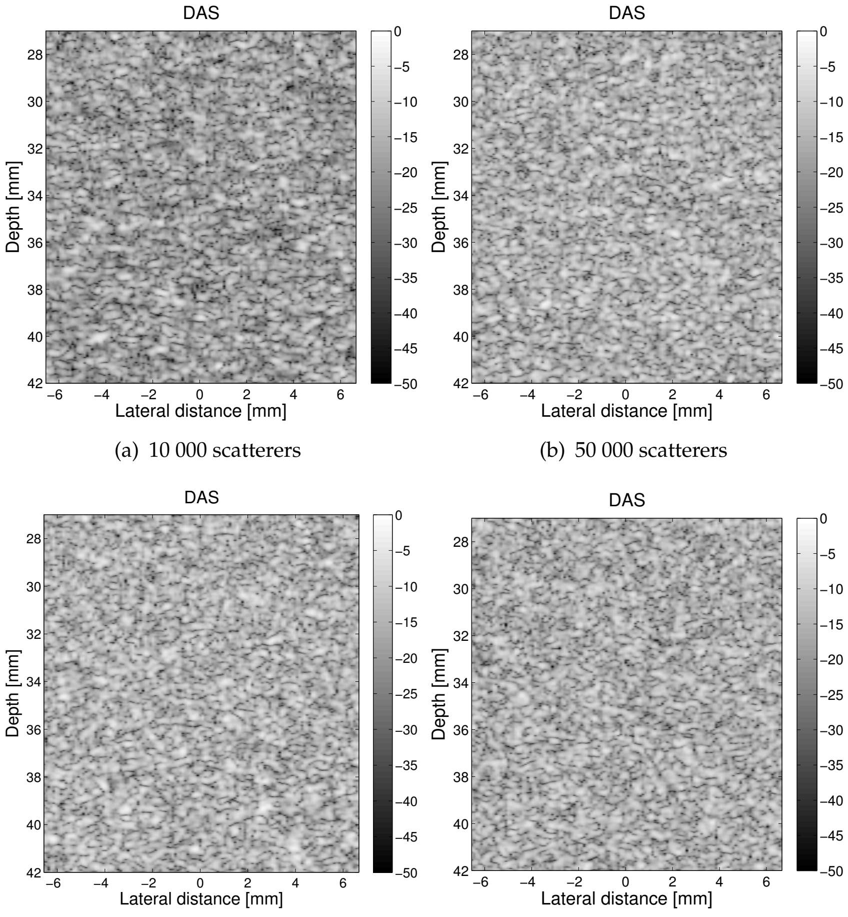
We have collapsed the elevation dimension, but given the scatterers Gaussian random amplitudes to mimic different position in the elevation dimension. Our resolution cell volume is, from Equation 3.7, 0.0554 mm$^3$ if the cell is at depth 27mm. We use depth of 27 mm because this gives the smallest resolution cell and thus the hardest requirements for the number of scatterers. Since we have collapsed the elevation dimension we end up with a two dimensional resolution cell of 0.0449 mm$^2$. We have simulated from 10 000 to 300 000 randomly distributed scatterers in the phantom. This resulting SNR is given in Table 3.2 and the resulting images are given in Figure 3.20.

From the SNR's listed in Table 3.2 we see that when the number of scatterers approximates 10 the SNR is $\approx$ 1.9, and thus the speckle is fully developed. If we investigate the images in Figure 3.20 we see that after we have reached 50 000 scatterers there is no visual difference in the contrast of the speckle pattern in the images. The 10 000 scatterers has more dark areas, while the three others seems equally distributed. This observation is confirmed by the distribution of the dB values in Figure 3.21 (b).

In Figure 3.21 we have plotted the distribution of the amplitude values of some of the simulations. We see that the simulation with 10 000 scatterers is far from its theoretical Rayleigh distribution, while the simulation with 50 000 and 100 000 scatterers is approximately equal to its Rayleigh distribution while the 300 000 scatterers simulation is very close to the theoretical. Ideally we should have used the 300 000 scatterers simulation,

| Speckle simulations | | | |
|---|---|---|---|
| Phantom size | Scatterers | ≈ Scatterers per resolution cell | SNR |
| 15x18mm | 10 000 | 2 | 1.64 |
| 15x18mm | 30 000 | 3 | 1.81 |
| 15x18mm | 50 000 | 8 | 1.87 |
| 15x18mm | 80 000 | 13 | 1.85 |
| 15x18mm | 100 000 | 17 | 1.85 |
| 15x18mm | 150 000 | 25 | 1.87 |
| 15x18mm | 300 000 | 50 | 1.92 |

*Table 3.2: SNR values calulated for different number of scatterers in the phantom when the DAS beamformer is used.*



(a) 10 000 scatterers

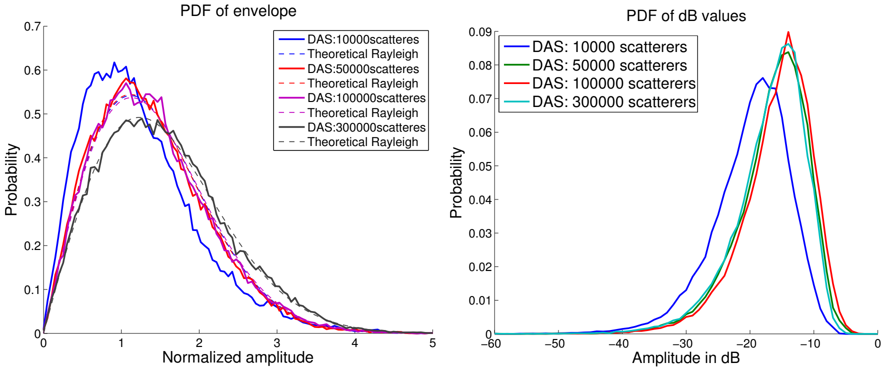
(b) 50 000 scatterers

(c) 100 000 scatterers

(d) 300 000 scatterers

*Figure 3.20: Speckle simulations with different number of scatterers. We see that there is no visible difference between 50 000 scatterers and 300 000 scatterers.*

but the time to run the simulations when using 300 000 scatterers is so high that it is much more practical to use the 100 000 scatterers - especially since we soon will oversample the speckle. The simulation with 100 000 scatterers is also Rayleigh distributed, so the speckle we investigate further is well developed.



(a) Distribution of amplitude values

(b) Distribution of decibel values

*Figure 3.21: Speckle PDF's. We see that the 50 000, 100 000 and 300 000 scatterers simulations are Rayleigh distributed (a). The distribution of dB values are very similar for 100 000 and 300 000 scatterers and since are both well developed.*

### 3.4.3 Speckle with Capon beamforming

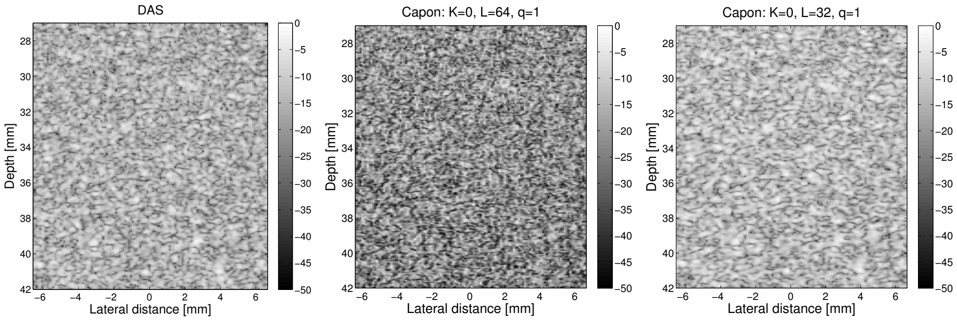
So far in the discussion on speckle we are assuming uniform DAS beamforming. When we use Capon beamforming we apply time-varying weights dependent on the spatial covariance matrix, see 2.3.2, thus resulting in a *varying resolution*. This will affect the speckle pattern created and give different statistics to the speckle (Synnevåg et al., 2007b).

| Speckle simulations with Capon beamformer | | | |
|---|---|---|---|
| Phantom size | Scatterers | $\approx$ Scatterers per resolution cell | SNR |
| 15x18mm | 10 000 | 2 | 1.09 |
| 15x18mm | 30 000 | 3 | 1.12 |
| 15x18mm | 50 000 | 8 | 1.12 |
| 15x18mm | 80 000 | 13 | 1.13 |
| 15x18mm | 100 000 | 17 | 1.13 |
| 15x18mm | 150 000 | 25 | 1.14 |
| 15x18mm | 300 000 | 50 | 1.14 |

*Table 3.3: SNR values calulated for different number of scatterers in the phantom when the Capon $K = 0, L = 64$ beamformer is used.*

The SNR of the most aggressive Capon beamformer $L = M/2$ calculated from the same simulations as in Table 3.2 is listed in Table 3.3, and thus the statistics are different for the Capon beamformer. We see from Table 3.3 that the SNR of Capon with $L = M/2$ is approximately 1.13 for what was well developed speckle for the DAS beamformer.

In Figure 3.22 we have displayed the speckle image with 100 000 scatterers from the DAS beamformer (a), Capon $L = M/2$ (b) and Capon $L = M/4$ (c). We immediately see that the most aggressive Capon beamformer creates a very different speckle pattern than DAS. The less aggressive Capon beamformer creates a pattern very similar to DAS, which is not too surprising since the less aggressive, and more robust, we make the Capon beamformer the more it is equal to a DAS beamformer with constant weights.



(a) DAS (b) Capon:$K = 1, L = M/2 = 64$ (c) Capon:$K = 1, L = M/4 = 32$

*Figure 3.22: 100 000 scatterers with different beamformers.*

The SNR of amplitude from the three different images is

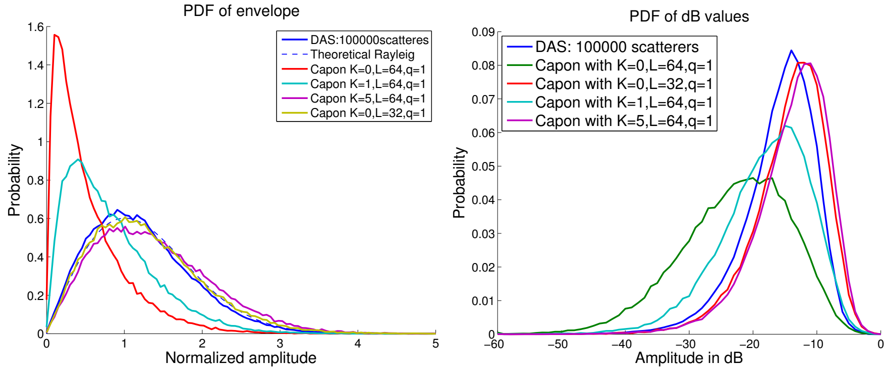
$$\begin{aligned} SNR_{DAS} &= 1.85 \\ SNR_{Capon_{K=0,L=64}} &= 1.13 \\ SNR_{Capon_{K=0,L=32}} &= 1.83. \end{aligned}$$

We see that the less aggressive Capon and DAS has approximately the same SNR. In Figure 3.23 we have plotted the distribution of the amplitude values, and as we would expect the distribution of both the less aggressive Capon, $L = M/4$, and DAS follows approximately the Rayleigh distribution. However, the aggressive Capon beamformer, $L = M/2$, has a very different distribution. The image has a much smaller and more distinct speckle pattern with clearer peaks and darker bottoms. This agrees with the findings by Synnevåg et al. (2007b). Hypothetically a more distinct pattern should result in better correlation and easier tracking of speckle and tissue movement. This is the main motivation behind this thesis, to investigate if adaptive, Capon, beamforming has benefits when we do static elastography. This will be investigated in depth in Chapters 5 and Chapter 6.

### Achieving similar speckle statistics for DAS and Capon

Synnevåg et al. (2007b) introduced the time averaging parameter $K$, see section 2.3.2, when estimating the spatial covariance matrix to get the same speckle statistics as in DAS. So far we have been using $K = 0$, meaning that we have calculated the spatial covariance matrix with one sample. With a larger $K$ we use $2K + 1$ samples to create the spatial covariance matrix. In Figure 3.23 we see that we need $K = 5$ to get the most aggressive Capon beamformer to achieve the Rayleigh distributed speckle statistics. If we once again look at the SNR's we see that as expected $K = 5$ gives the Capon beamformer the same SNR as DAS.

$$SNR_{DAS} = 1.85$$
$$SNR_{Capon_{K=5,L=64}} = 1.85.$$

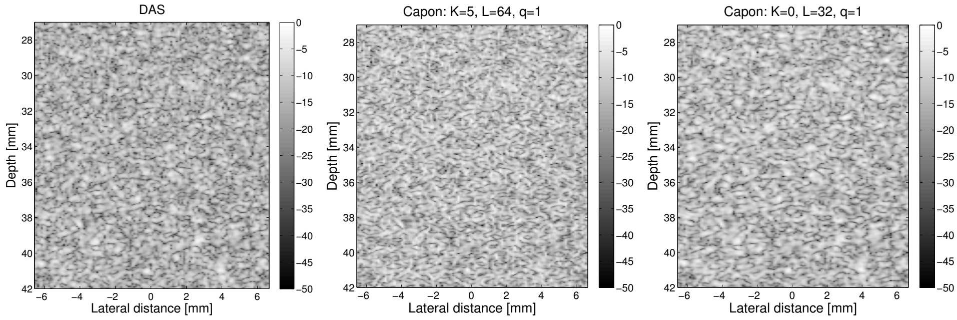


(a) Distribution of amplitude of the envelope.

(b) Distribution of dB values

*Figure 3.23: Speckle PDF's from DAS, Capon with $K = 0, L = 32$ and Capon with $L = 64$ with different $K'$s.*

When investigating the images resulting from averaging more samples, see Figure 3.24, we see that the pattern becomes more similar to the DAS patter. However, the Capon image still has a different and more distinct pattern with smaller *speckles*, motivating us to also explore this pattern when we estimate displacement in Chapter 5.



(a) DAS

(b) Capon:$K = 0, L = M/2 = 64$

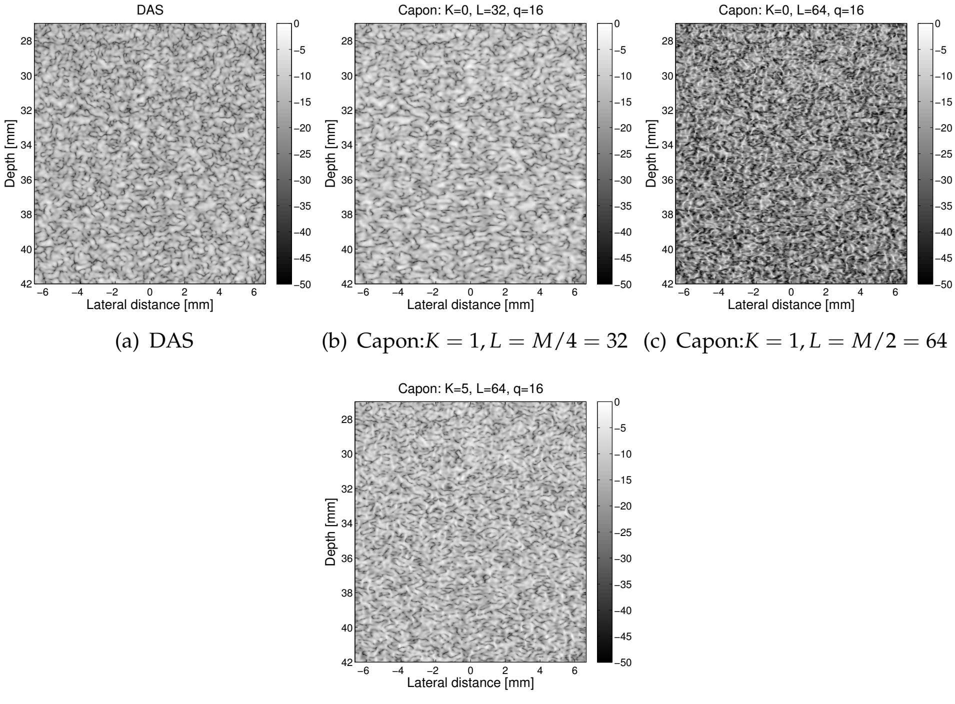
(c) Capon:$K = 0, L = M/4 = 32$

*Figure 3.24: Speckle simulations with 100 000 scatterers with different beamformers.*

### 3.4.4 Speckle with lateral oversampled beamforming

From our earlier discussion on lateral oversampling with Capon beamforming we saw that we needed an oversampling factor of $q \approx 16$ to get below 3 dB in difference between the peaks when imaging point scatterers.

Point scatter images are the worst case and point scatterers seldom occur *in vivo*. We start off by creating our speckle simulation with the lateral oversampling factor $q = 16$, the factor needed for not losing information in the scatterer images. Images created with the same beamformers as earlier are displayed in Figure 3.25. If we closely compare these images, especially image (c), to the earlier images without lateral oversampling, $q = 1$, we will see that the lateral oversampling reduced the sharp jumps that occurred in Figure 3.22 (b). This is hard to see in the printed version, but a good eye might spot it. This indicates that we might loose information with Capon beamforming when imaging speckle and motivates us to investigate this in detail.



(a) DAS

(b) Capon:$K = 1, L = M/4 = 32$

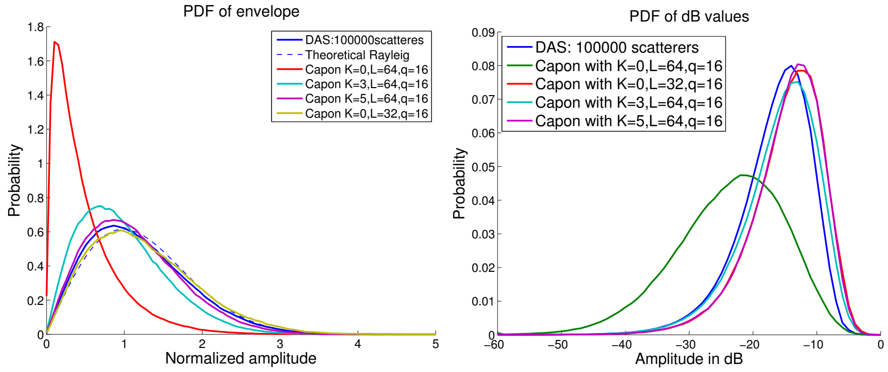
(c) Capon:$K = 1, L = M/2 = 64$

(d) Capon:$K = 5, L = M/2 = 64$

*Figure 3.25: Speckle simulations with 100 000 scatterers laterally oversampled with a factor* $q = 16$.

First we will investigate what parameters that gives us Rayleigh statistics. Our earlier result that the less aggressive Capon beamformer, $L = 32$, already has Rayleigh statistics, while the aggressive Capon beamformer, $K = 64$, needs $K = 5$ also for lateral oversampling with $q = 16$ is confirmed by the plots in Figure 3.26.

Since single point scatterers are the worst case we should expect that we can create the image of well developed speckle with $q < 16$. A lateral line from 30 mm depth from Figure 3.25 (c) is plotted in different decimated versions, thus the different $q's$, in Figure 3.27. The first thing we see is that the two versions of the DAS line do not differ much even though there is a difference in $q$ of 16. This is as expected since the lateral beam spacing with $q = 1$ is according to the Nyquist sampling criteria for DAS, see Section



(a) Distribution of amplitude of the envelope.

(b) Distribution of dB values

*Figure 3.26: Speckle PDF's. We see that also when using oversampling factor $q = 16$ a temporal averaging with $K = 5$ is sufficient for the Capon beamformer to have Rayleigh distributed speckle.*

3.1.5. If we further investigate the plot we see that the Capon line with $q = 1$ clearly misses many of the bottoms and the peaks, so much of the information is lost. In other words the image is laterally undersampled. The higher value of $q$ we use, the better the Capon lines fit the Capon line with $q = 16$, but from the plot it looks like $q = 4$ is sufficient. It is only at the very steep peak right before lateral position -6 mm that $q = 4$ doesn't fit very well.

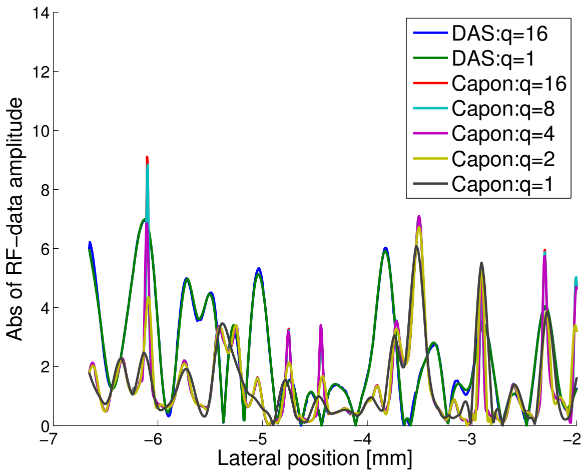


*Figure 3.27: A lateral line plotted with different oversampling factors for DAS, and Capon with $K = 0, L = 64$.*

In Figure 3.28 (a) we have plotted the sum of the absolute error of different oversampling factors $q$ compared to $q = 16$ for all the lines of the image in Figure 3.25 (c). The plot is a mean-variance plot and we have plotted the difference between DAS with $q = 1$ and $q = 16$ as reference. This plot confirms our earlier observation that $q = 4$ is sufficient. Actually $q = 4$ has less error than the DAS reference. In the bottom plot we have plotted different interpolation techniques to see if the result was dependent

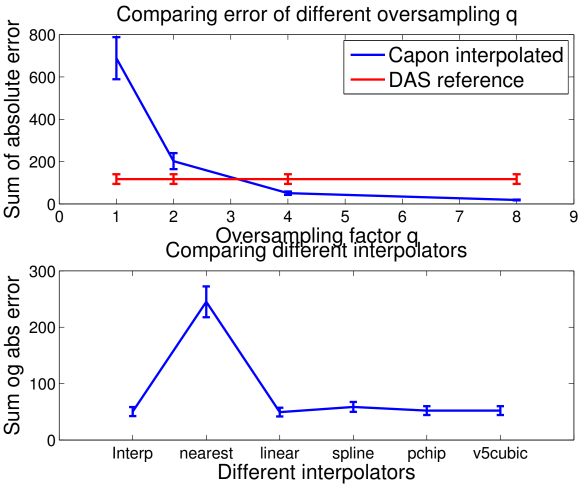


*Figure 3.28: Speckle PDF's*

on the interpolation technique. We see that as long as we use a more sophisticated technique than simple nearest neighbor interpolation it did not affect the result much.

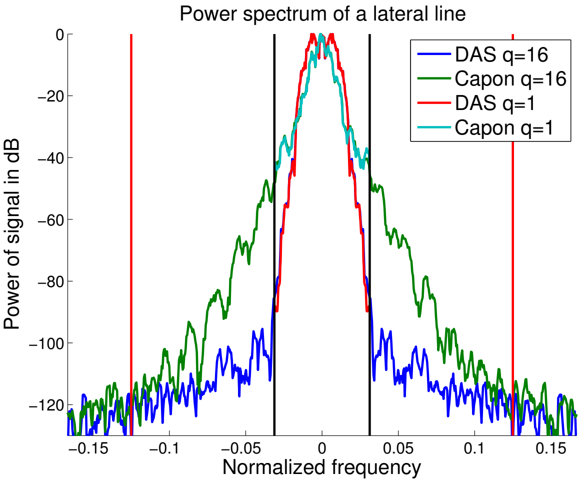
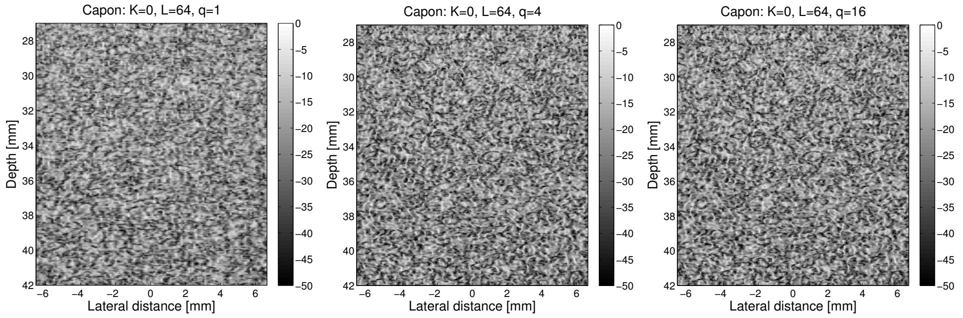


*Figure 3.29: Power frequency spectrum of a lateral line for DAS and Capon $K = 0, L = 64$ for $q = 16$ and $q = 1$.*

To understand why $q = 4$ is sufficient we can investigate the power spectrum of a lateral line, this is plotted in Figure 3.29. The black vertical lines in the plot indicate the sampling frequency if we would have used $q = 1$. The energy in the frequencies above this line will be folded back and aliased into the signal. We see that for DAS the signals outside the black lines are very small, actually less than -100 dB. This is of course because $q = 1$ is based on the Rayleigh criteria for DAS. While the higher frequencies from the Capon beamformer has much more power and thus we are actually losing information if we cut off at the black lines.

The red vertical lines in the plot indicate where we will cut off if we use $q = 4$. We see that when using $q = 4$ we get almost all the information from the higher frequencies for Capon, and the frequencies folded back into our signal has very low power, less than -120 dB. We therefore expect that by using an oversampling factor of $q = 4$ when imaging speckle with Capon beamforming we do not lose much information. This is a drastically lower oversampling factor than for point scatterers and is a much nicer value to work with when trying to get Capon to run real time on a ultrasound scanner. Since single point scatterers seldom occur *in vivo* an oversampling factor in this range should be sufficient. However, it is important to point out that these are results from one simulated probe. Åsen et al. (2014a) showed that for their setup with a phased array probe, an oversampling factor of 25 was needed when imaging point scatterers. We needed a factor of 16, and thus the oversampling factor is system dependent. Further research should be done to find what influence the choice of lateral oversampling facto for Capon.

In Figure 3.30 we have plotted Capon images of speckle from our 100 000 scatterers phantom by using $q = 1$. In (a) we can see sharp jumps in intensity between the lateral lines so we can actually see that we in fact lose information. The (b) and (c) image is created with an oversampling factor of $q = 4$ and $q = 16$ and as we can see the sharp jumps between lines is gone and there is also no visible difference between the images supporting our claim that $q = 4$ is enough for our system when imaging speckle. As stated earlier these jumps are hard to see in the printed images, but a good eye might spot them.



(a) Capon:$K = 0, L = M/2 = 64$ (b) Capon:$K = 0, L = M/2 = 64$ (c) Capon:$K = 0, L = M/2 = 64$

*Figure 3.30: Speckle simulations with 100 000 scatterers laterally oversampled with* $q = 1$ *(a),* $q = 4$ *(b) and* $q = 16$ *(c).*

## 3.5 Summary

In this chapter we have described the details of the ultrasound probe and setups we use to simulate different ultrasound images in Field II. We have investigated resolution for all dimensions, the resolution cell and the two-way resolution. The lateral sampling, beam spacing, have also been discussed.

Point scatter simulations have been investigated, and especially the lateral oversampling when doing Capon beamforming. We have seen that if we do not have sufficient lateral sampling a point scatter between two beams loose 40 dB in amplitude compared to a point scatter hit by the beam, meaning that if we do not have lateral oversampling when doing Capon we loose information. The lateral oversampling needed to avoid losing information is $q \approx 16$ when imaging single point scatterers.

We have also looked at the beampattern created by Capon and compared it to the DAS beampattern to increase our intuition and understand how the Capon beamformer works, and seen the superior lateral resolution of the Capon beamformer.

Speckle has been discussed and especially speckle statistics of both the DAS beamformer and Capon beamformer - and we have seen that by using temporal averaging the Capon beamformer gets the same Rayleigh distributed amplitudes as the DAS beamformer.

Toward the end of the chapter we looked at the lateral oversampling needed when imaging speckle, and found that for our setup when imaging well developed speckle a lateral oversampling factor of $q = 4$ was sufficient.

# Chapter 4

# Measuring Force Applied From Ultrasound Probe to Tissue

***Note:*** *This chapter breaks the flow of the thesis and can be skipped. Chapter 5 follows the results from Chapter 3. This chapter originates from the original objective of thesis, but the negative results of this chapter redefined the thesis' objective. This chapter is kept in this succession in the thesis because it fits with the timeline in which the work was done.*

***Chapter abstract:*** *The original objective of this thesis included to create a system to measure the force applied from the ultrasound probe to the tissue. If this force is measured we might get a quantitative measure on the elasticity of the tissue. This chapter describes a system using simple force sensors applied to the front of the probe. We will see that this turned out to be very inaccurate. This changed the main objective of the thesis, and leaves this chapter standing alone.*

In static elastography the deformation in the tissue is created by the ultrasound probe. To be able to get a quantitative measure of tissue elasticity, to reduce the inter operator variability, we need to know how much force that is applied to the tissue. In this chapter we will investigate if we are able to use fairly simple force sensors to measure the force applied to the tissue.

## 4.1 Force sensor

To measure the force we have decided to use a sensor called *FlexiForce®* developed by Tekscan, Boston, MA, USA. The sensor is small and thin, so one or more sensors are easily mounted on the front of the ultrasound probe by the help of our constructed *probe collar*, see Section 4.2. We are using Tekscan's standard A201 force sensor, constructed as in Figure 4.1. The sensor construction consist of two layers of flexible substrate (polyester) film. To each layer a conductive silver material is applied followed by a layer of pressure-sensitive ink. The two layers are laminated

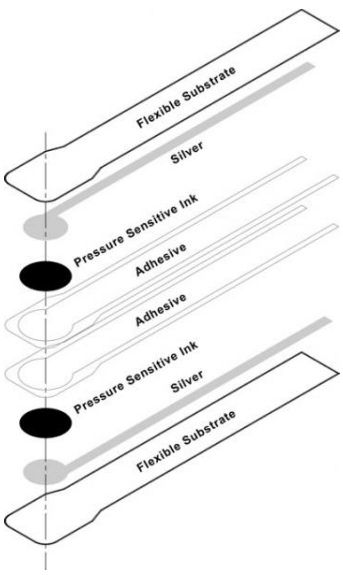


*Figure 4.1: Illustrating the construction of the FlexiForce sensor. Figure from [http://www.tekscan.com/flexible-force-sensors]*

together, leaving the active sensing area at the end of the sensor as a circle with diameter of 9.53 mm.

The sensor acts as a variable resistance. When no force is applied to the sensor, the resistance, $R$, is very high. When a force is applied to the sensor, the resistance decreases. When using sensors as measurement devices the linearity of the sensor is important. In the FlexiForce® sensors it is the conductance (1/R) that is linear.

The FlexiForce® sensor comes in three different sensing ranges with a maximum load of either 4.4 N , 110 N or 440 N. We are using 110 N sensors expecting a force of 0-10 N applied to the tissue.

### 4.1.1 Method

To use the sensor we set it up as the top resistor, $R1$, in a voltage divider, see Figure 4.2.

The equation for a voltage divider is

$$V_{\text{out}} = \frac{R_2}{R_1 + R_2} V_{\text{in}}. \tag{4.1}$$

Since we are interested in the resistance value of the sensor we can rearrange Equation (4.1) and get

$$R_1 = \frac{R_2 V_{\text{in}}}{V_{\text{out}}} - R_2 \tag{4.2}$$

as an expression for the resistance in the sensor. Since we know the value of $V_{\text{in}}$ and $R_2$, all we need to measure is $V_{\text{out}}$ to be able to calculate $R_1$. $V_{\text{out}}$ is measured by wiring the point between the resistors to an analog

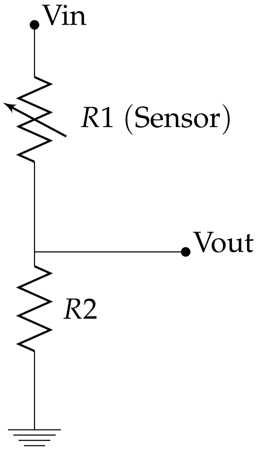


*Figure 4.2: Schematics of a voltage divider.*

pin on an Arduino Uno microcontroller[1]. The Arduino has a 8 bit analog to digital converter (ADC), so what we read is a quantized value of the voltage between 0 and 1023. The setup with two sensors and the Arduino can be seen in Figure 4.3.

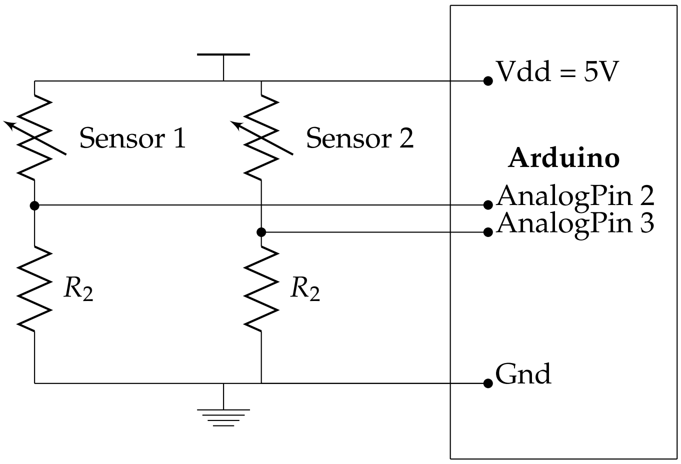


*Figure 4.3: Schematics of the two sensors in voltage dividers connected to the Arduino.*

There are several methods to measure the change of a devices resistance. We could use e.g. a Wheatstone bridge, which is basically two voltage dividers in parallel where the voltage we are interested in is the voltage difference in the two legs. This might be a more accurate approach, but is also more complex. So, for simplicity we ended up using only a single voltage divider.

### 4.1.2 Calibration

The manufacturer of the force sensors provides a guide to calibrate the sensors. First of all the manufacturer recommends to "condition" the sensor before every use. The conditioning is to place 110% or more of the

[1] See www.arduino.cc for more information on the Arduino microcontroller.

maximum weight onto the sensor for about 3 seconds, and repeat this 4-5 times.

The rest of the calibration procedure consists of 5 steps

Step 1 Place 1/3 of the test weight on the sensor and read the output.

Step 2 Place 2/3 of the test weight on the sensor and read the output.

Step 3 Place the full weight on the sensor and read the output.

Step 4 Plot the conductance (1/resistance) vs force and draw a line of best fit.

Step 5 Use the equation for the line of best fit and the sensor output to determine the force of unknown loads on the sensor.

To make sure that all the force is applied to the sensors sensing area, the circle of diameter 9.53 mm, the manufacturer recommend to use a *puck* between the weight and the sensor. For the first calibration this puck was created by cutting out a circular shape of a hard plastic material about 3 mm thick. This puck gave very unreliable results with high variance between sensors and measurements.

This could mean that the sensors are bad and inaccurate, or that the calibration is bad. The latter is most likely, and therefore, inspired by Vecchi et al. (2000), parts for a more accurate calibration was drawn in Solidworks 2013 and printed on a 3D-printer. The 3D-model of these calibration parts can be seen in Figure 4.4.

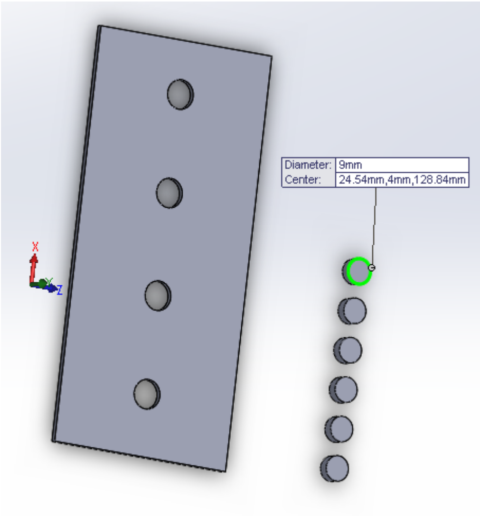


*Figure 4.4: Calibration parts for a more accurate calibration.*

The calibration parts consist of the pucks, to the right in the figure, and a board with holes to place the pucks. The pucks had a dimension of 9 mm, while holes in the board had a dimension of 10 mm. The hole in the board was placed directly above the sensing area of the sensor, and the puck was placed in the hole making sure all that the weight was placed directly on the sensor area.

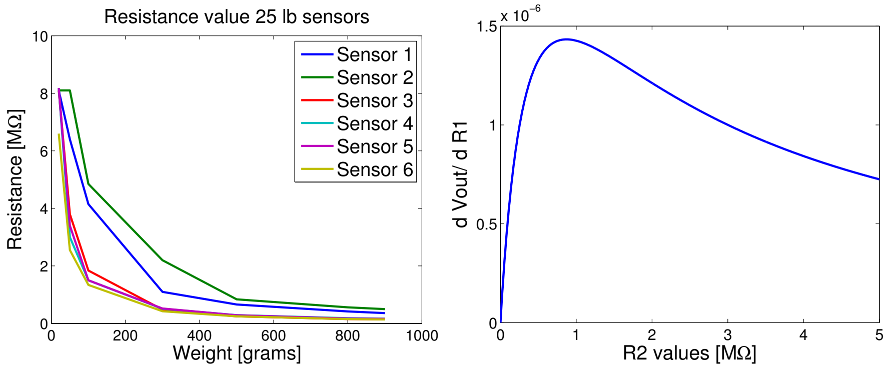
*Figure 4.5: Calibration weights*

Since we read the values from a voltage divider it is important to balance the resistance values to get the highest possible range of output values. We therefore measured the value of the sensors resistance when applying 0 to 900 grams of weight. The weight is applied using calibration weights with given weight, see Figure 4.5. This resulted in the plot in Figure 4.6 (a).

We want to have the largest possible range for the output value $V_{\text{out}}$ by choosing the best balanced value of $R_2$ compared to the sensors resistance in the working range. To find the middle of the working range we average the values of resistance $R_1$ for the different sensors and chose the average value at 300 grams. If we then differentiate equation (4.1) with regard to $R_1$ we get

$$\frac{\partial V_{\text{out}}}{\partial R_1} = \frac{V_{\text{in}} R_2}{(R_2 + R_1)^2}.$$

This equation is plotted against values of $R_2$ from 0 to 5 $M\Omega$ in Figure 4.6 (b). We see that we get a peak at approximately 1 $M\Omega$ so we chose this value for $R_2$ in our circuit giving us the largest range for the output value $V_{\text{out}}$, for $V_{\text{in}} = 5\text{V}$ and varying $R_1$, to be read by the analog pin (ADC) in the Arduino.



(a) 25 lb sensors

(b) 1 lb sensors

*Figure 4.6: Calibration curves for resistance value (a), and the derived $V_{out}$ expression vs value of $R_2$ in (b).*

The next step was to calibrate each individual sensor in the voltage divider circuit using the same calibration weights as we used to find the resistance values, thus from 0 to 900 grams. The calibration parts were used as described earlier and the measured result of the conductance (1/R) is plotted in Figure 4.7. We see from the plots that the sensors are fairly linear and there are not so much difference between each sensor except that sensors 1 and 2 are very different from the rest. This is probably because sensors 1 and 2 are from a different production batch than the rest of the sensors. If the sensors are studied carefully it looks like it is an extra layer of adhesive, see Figure 4.1, in the construction of the sensors 3-6, but the manufacturer has not given any details on this.

From the curves in the plots we calculated the best fitting line for each sensor with linear regression. The resulting function is used as each sensors calibration to convert the measured voltage into force.

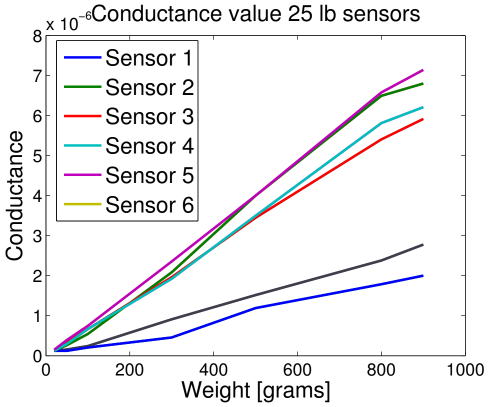


*Figure 4.7: Calibration curves for the 25 lb sensors.*

### 4.1.3 Software

In the previous section we have not given any details on how we received the values read by the Arduino. This was done using *MATLAB Support Package for Arduino (aka ArduinoIO Package)* created by *MathWorks Classroom Resources Team* available at *www.mathworks.se/matlabcentral*. This software allows very easy communication between MATLAB and an Arduino connected with USB. The standard Arduino commands can be run through this MATLAB API so the resulting values are read straight to MATLAB. A MATLAB GUI using this API was created to easily read and display the values of the sensors. A screenshot of the GUI while measuring one sensor is shown in Figure 4.8.

## 4.2 Probe collar

To be able to mount the force sensors on the front of the ultrasound probe we needed to create a *collar* to fit around the ultrasound probe. This collar was drawn in the 3D-design software Solidworks 2013 and the collar was created in two steps. First a 3D-model based on the ultrasound probe was created, see Figure 4.9 (a), then a second part, the actual collar Figure 4.9

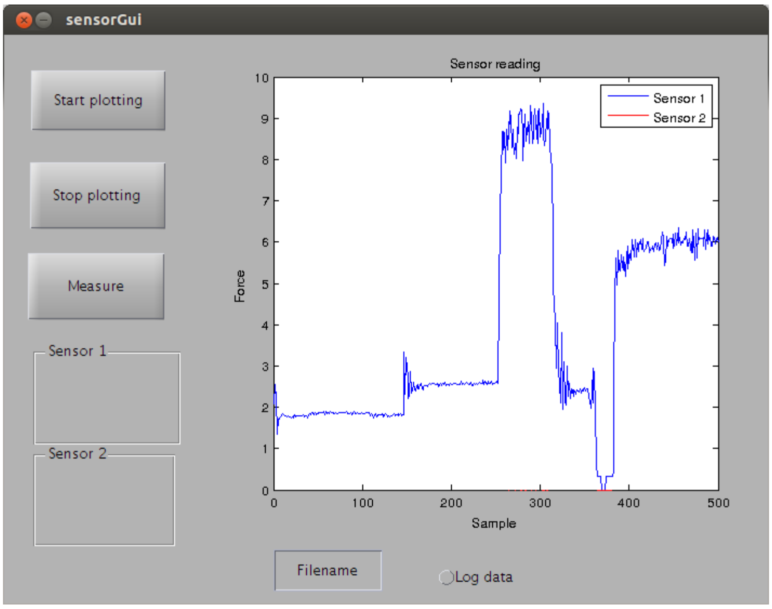
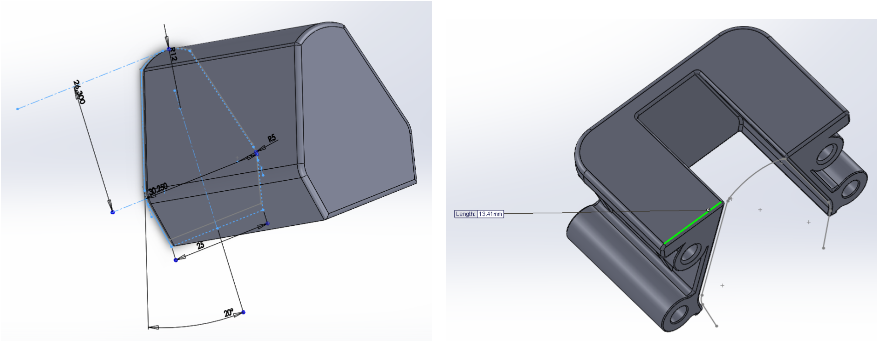


*Figure 4.8: A MATLAB GUI that reads the value of the sensors from the Arduino at real time.*

(b), was created to fit around the probe. This collar was created as two counterparts screwed together to tightly attach around the probe. The width of the side of the collar was 14 mm, giving enough space to mount the force sensors on the collar and keeping the entire sensing area of the sensor on the collar. Acknowledgement goes to associate professor Mats Høvin at the Robotics And Intelligent Systems at UiO for creating the original 3D-design, and to Yngve Hafting for 3D-printing the designs.



(a) 3D model of ultrasound probe

(b) 3D model of probe collar

*Figure 4.9*

## 4.3 Results and discussion

### 4.3.1 Verification of calibration

To verify the calibration of each sensor we once again applied the calibration weights from 0 to 900 grams and measured the calibrated value of force from each sensor. The force is fund by $F = ma$ and using $a = 9.81m/s$. The result is plotted in Figure 4.10, where we see that all the sensors except sensor 1 behave as expected. The sensors do have some variance, but it's not too bad regarding that these are very simple and cheap sensors. Sensor 1 has some bad bias towards the second half of the scale. It was only done one test and one measurement so this bias could be inaccuracy when applying the calibration weights, that the sensor was not "broken in" properly or simply that this was a bad calibration or a bad sensor. We did not pay much attention to this since we could simply choose to use the other sensors.

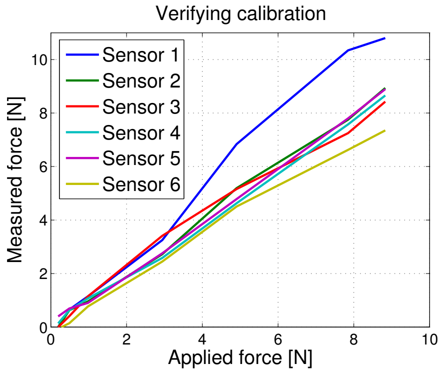
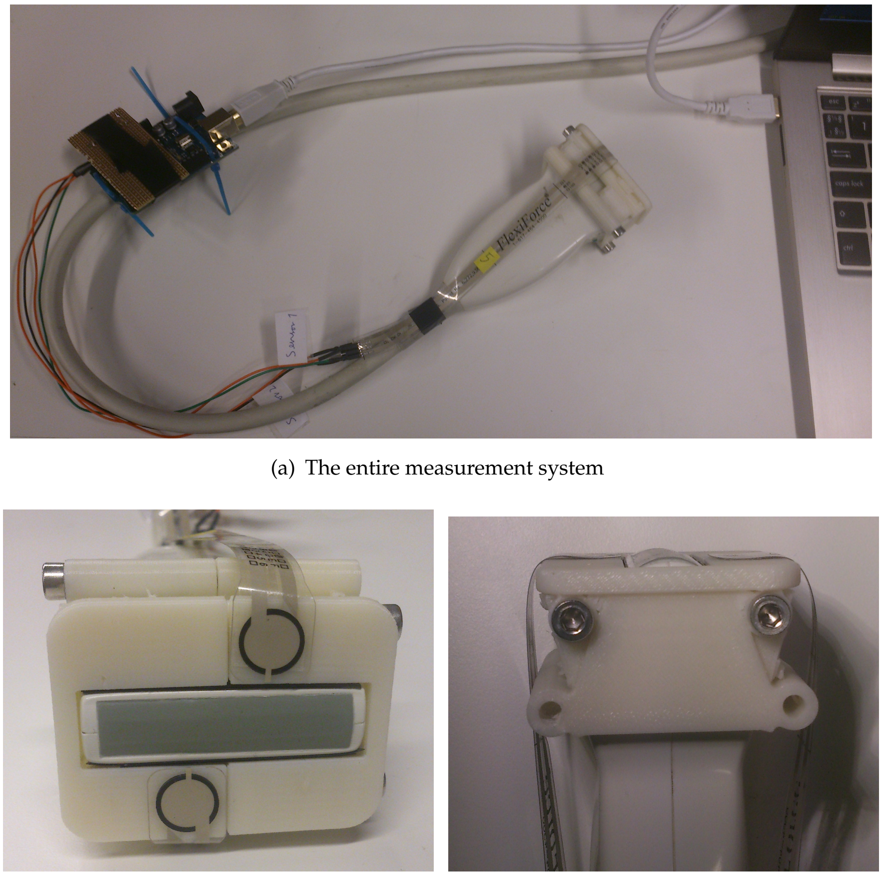


*Figure 4.10: Verifying the calibration of the sensors.*

### 4.3.2 The system setup

When putting all the parts of this force measuring system together we ended up with the system shown in the images in Figure 4.11. The system consist of the Arduino microcontroller attached to the probe cable. The two resistor divider circuits are soldered on a small prototyping board inserted directly into the correct ports on the Arduino. The two sensors are connected to the Arduino through cables soldered into the circuits allowing the sensors to be attached to the front of the probe as seen in the images.

The Arduino was attached to a computer through a USB cable which also provided the microcontroller with power. Controlling measurements of the $V_{dd}$ on the Arduino showed that the power provided through the USB was enough to keep a steady $V_{dd}$ at 5 V as long as the laptop was connected to a wall socket or had sufficient battery power.



(a) The entire measurement system

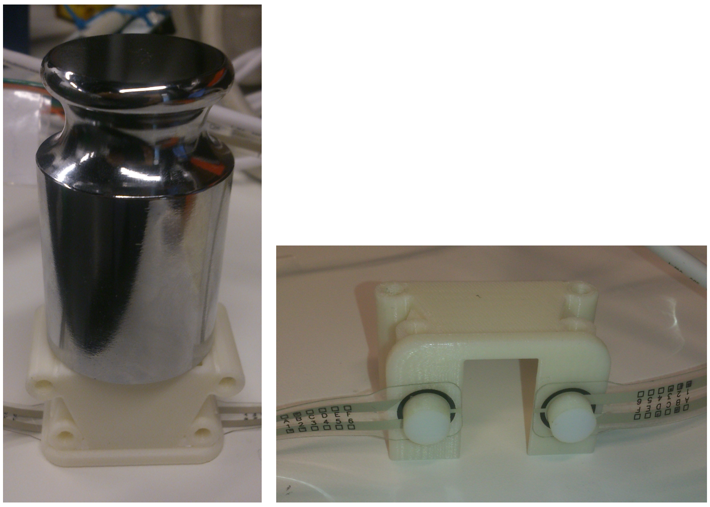
(b) Probe collar with force sensors from the front

(c) Probe collar with force sensors from the side

*Figure 4.11: Images of the measurment system setup.*

### 4.3.3 Verification of the system setup

To verify the systems capability of accurately measuring the force applied from the ultrasound transducer we used one half of the two collar counterparts, and attach the two sensors to this half. Then a known force was placed on this half and the sensors output was measured. An image of this setup can be seen in Figure 4.12 (a).



(a)

(b)

Figure 4.12: Image (a) shows the test setup and image (b) the pucks attached to the sensors.

During this test the first sign of problems arose. The assumption underlying this whole setup was that the force the sensor measured was proportional to the whole collar and probe front area. To simplify calculation for gained intuition we can assume that the sensing area of the sensor is 1 $cm^2$ while one half of the collar counterparts has an area of 5 $cm^2$. If the sensor senses 1 N it means that it is measuring a pressure of $1kPa$ since

$$P = \frac{F}{A} \quad \Bigg| \quad \begin{array}{r l} P: & \text{Pressure in Pascal} \\ F: & \text{Force in Newton} \\ A: & \text{Area in } m^2. \end{array}$$

This means that the force applied to the probe is 5 N since the pressure should be equal across the entire area of 5 $cm^2$ and thus

$$P_{\text{sensing area}} = 1kPA = \frac{F_{\text{total}}}{A_{\text{total}}} = P_{\text{total area}} \Rightarrow F_{\text{total}} = A_{\text{total}} 1kPa = 5N$$

is *applied to the probe*.

The actual area of one half of the collar counterparts is $\approx 9$ $cm^2$ and the sensing area of the sensor is $A_{\text{sensor}} = \pi(\frac{0.953}{2}cm)^2 = 0.713cm^2$. If we apply 4.95 N (500 grams) we expect the sensors to read $F_{\text{sensor}} = \frac{A_{\text{sensor}}}{A_{\text{total}}} F_{\text{total}} = 0.39N$.

When we conducted this experiment we measured $\approx 2.5N$ on each of the sensors, indicating that something in our assumption is wrong. If we closely investigate the image in Figure 4.11 (c) we see that the sensors build up about half a millimeter from the actual collar. This means that the sensors are getting more pressure/force than the area around, and thus sensing a higher force than the rest of the area of the collar. We tried two approaches to counteract this problem. First we tried to test against a softer background material and chose a stack of papers. The sensor readings were then reduced to $\approx 1.75N$ on each of the sensors. This could mean that more of the collar is in contact with the stack of papers so it is ″unloading“ some of the pressure from the sensors. We then tried to attach one layer of tape on all the area of the collar except on the actual sensors. This did not influence the measurements and we were still measuring $\approx 1.75N$ on each of the sensors. This is probably because one layer of tape was still lower than the height of the sensors so most of the pressure were still only on the sensors. When we added another layer of tape we measured $\approx 1N$ on both the sensors. From these varying measurements we see that our assumption that the force the sensors were measuring was proportional to the whole area of the collar was wrong. This is probably because the sensors are higher than the rest of the area so they experience more force than the area around. A third way to counteract this could be to cut out slits to place the sensors in the slit giving a smooth surface. The function of the sensors is that the two layers of conducting silver material is squeezed together in conductive ink. So when the force increases the resistance decreases. If the sensors were placed in a slit the layers in the sensors would not have been squeezed correctly together and we would probably encounter other errors in the measurements.

To further complicate things with our setup, Figure 4.11 (c) shows another potential problem. The tip of the ultrasound probe is slightly above the collar. This tip will therefore absorb much of the force between the collar and the tissue adding another error to our measurements. Another element that might complicate things is the ultrasound gel used between the probe and the tissue.

As a final test of the setup we attached two pucks on the sensing area of the sensors, Figure 4.12 (b). When we now added 4.95 N of force we measured the expected $\approx$ 2.5 N at each sensor. The conclusion is that the FlexiForce sensors are only accurate when all the force is applied directly on the sensing area of the sensor. This conclusion is supported by (Vecchi et al., 2000), where they created a *dome* attached to the sensing area of the sensor. They say in the article that they did this *″To create a sensor that responds to force rather than pressure ...“* but from our experience I doubt that they could get the sensor to accurately respond to pressure anyway.

## 4.4 Summary

In this chapter we have seen an approach to create a setup to measure the force exerted from the ultrasound probe to the tissue. The motivation was that if the force applied to the tissue is known, we could be able to get a quantitative measure of the tissue elasticity when doing static elastography. When the force applied is not known, as in conventional static elastography, we only measure the relative tissue elasticity. The approach chosen in this thesis was to attach simple force sensors at the front of the probe to measure the force between the probe and the tissue. The assumption was that the force measured by the flat sensors was proportional to the force applied to the probe by the operator. This assumption turned out to be an underestimation of the complexity of measuring the force since small variations in height in the front of the probe were shown to highly influence the measurements. The task of measuring the force turned out to be a more complex mechanical problem then first assumed.

An alternative approach to measure the force has been done in Gilbertson's master thesis *Handheld Force-Controlled Ultrasound Probe* (Gilbertson, 2010) from Massachusetts Institute of Technology (MIT). His approach was a much more complex mechanical approach containing a linear actuator with a six axis force sensor, see Figure 4.4. A feedback controller commands the linear actuator to move the ultrasound probe in accordance to the readings from the force sensor. In this way the system will control the probe to maintain a given force independent from the force exerted by the operator. This is a much more complex but also probably much more accurate approach - and it would be very interesting to examine static elastography where the compression force is known.

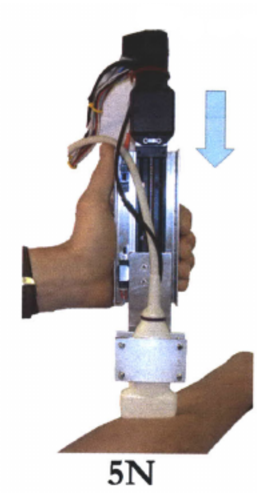


*Figure 4.13: Image of Gilbertson's Handheld Force-Controlled Ultrasound Probe Gilbertson (2010)*

# Chapter 5

# Static elastography

***Chapter abstract:*** *This chapter provides an in-depth investigation of static elastography using two different techniques for the tissue displacement estimation; crosscorrelation and pulsed-Doppler. The first part of the chapter introduces a simple approach to simulate static elastography. In the second part the crosscorrelation displacement estimation techniques is described in detail, while the third part describes the pulsed-Doppler technique. The fourth part presents two comparison criteria to compare the adaptive and conventional beamforming when the outputs of the beamformers are used for displacement estimation. The last part investigate the least square approach of estimating the strain from the displacement, and also a optimized version of this technique.*

Elastography was introduced in the second chapter, where we introduced two elastography techniques; static and shear wave elastography. In this chapter we will further investigate static elastography. Static ultrasound elastography uses two images, pre and post a compression, where the compression is caused by the ultrasound probe. This chapter continues the work done in Chapter 3 where we studied the speckle patterns that will be used to extract the tissue movement between the two images.

## 5.1 Simulating tissue compression

The thyroid, situated below the Adam's apple in the neck, is one of many organs suitable for ultrasound investigation. Nodules in the thyroid are very common and are easily seen in ultrasound images. The great majority of the nodules are benign while only 5% of the nodules are malignant (Rago et al., 2007). Malignancy can not be decided by regular ultrasound, and biopsy is the common way to investigate suspicious nodules. In recent years ultrasound elastography has shown very good results when used to predict malignancy in thyroid nodules. Studies report results with positive predictive value, the probability of being sick given positive test, of 100% and negative predictive, the probability of being sick given negative test, value of 98% (Rago et al., 2007).

These results motivated us to simulate conditions equal to thyroid imaging. We are still using the same setup as in Chapter 3 with a 7.5

MHz, 192 element linear probe using 128 active elements to image the same phantom as used in Section 3.4. We are imaging from 27 mm to 40 mm in the axial dimension, which are reasonable depths when imaging the thyroid. The thyroid also seem to have a very well defined and well developed speckle pattern, based on investigation of my own and my supervisors thyroid scanned by the VingMed Vivid FiVe system in our lab.

To simulate tissue compression we first created the pre-compression image by simulating the same phantom as in Section 3.4. This phantom had 100 000 scatterers randomly distributed in two dimensions, axial and lateral, with Gaussian randomly distributed amplitude to mimic different positions in the elevation dimension. More details are given in Section 3.4. To create the tissue compression we displaced the scatterers in the phantom, by the amount indicated by the colors in Figure 5.1, in the axial dimension. This is of course a very simple model for tissue displacement, but it supports the fundamental assumption that the speckle pattern follows the tissue movement along the axial dimension.

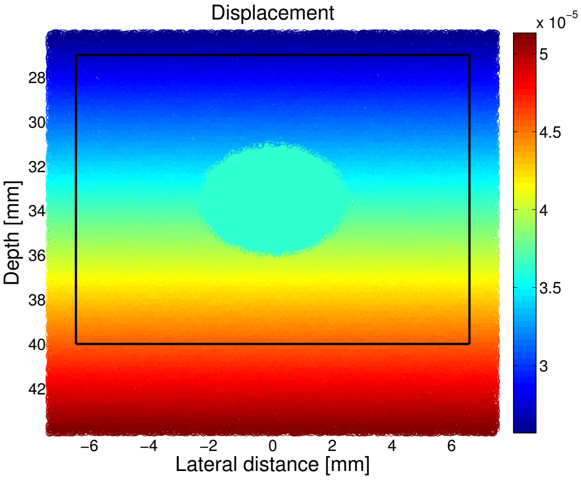
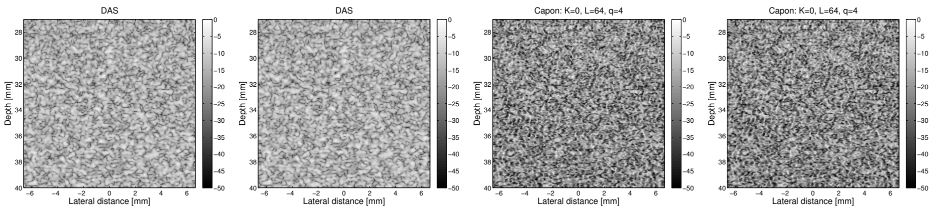


*Figure 5.1: Displacement used to simulate tissue compression. The colors indicates how much each scatterer will be displaced between the pre- and post-compression image. The black box indicates the region used to avoid fussy edges, see Section 3.4.2.*

In Figure 5.1 all the scatterers in the phantom are plotted at their position prior to compression. The colors in the image indicate how much each scatter will be displaced before we create the post compression image. The scatterers get gradually more displaced the deeper we get in to the tissue. This is because the front of the probe is the reference point for the image. This means that the tissue far from the probe will have a larger relative displacement than the tissue close to the probe (Ophir et al., 1991). In the middle of the phantom we have created a circular object where all the scatterers are displaced by the same amount - this is to mimic a hard malignant nodule in the tissue. The nodule has a diameter of 5 mm and is placed with its center at 33.5 mm depth in the center of the image.

The resulting pre- and post-compression images are displayed for both the Capon beamformer, $K = 0, L = M/2 = 64$, and the DAS beamformer in Figure 5.2. All images in this chapter are created with a



(a) DAS pre (b) DAS post (c) Capon pre (d) Capon post

*Figure 5.2: Pre- and post-compression images for both the DAS and Capon,$K = 0, L = 64$ beamformer.*

lateral oversampling factor of $q = 4$ according to the results in Chapter 3. In the particular example used in Figure 5.1 and Figure 5.2 we have displaced the furthest scatterers, at 44 mm, by $\Delta = \frac{\lambda}{4} \approx 0.05$ mm, where $\lambda$ is the wavelength of the center frequency. The scatterers closest to the probe, at 27 mm, are displaced by half the amount at 44 mm, in this example $\frac{\Delta}{2}$. The reason for this small compression is because of the fundamental limit of the pulsed-Doppler estimation which we will discuss later. We will later use different and larger amounts of displacement - and then relate it as a factor of the $\Delta$ value derived here.

## 5.2 Crosscorrelation displacement estimation

In their original paper Ophir et al. (1991) suggested to find the displacement difference between the pre-and post-compression axial lines by the time shift differences of the segmented lines. The straight forward way is to divide each line into windows of a certain length. Then each window is cross correlated against the window from the same position in the post compression line to find the time shift between the windows. Mathematically we can express this as (Sæbø et al., 2007):

$$\left.\begin{aligned} f_{\text{pre}}(t) &= s(t) + n_1(t), \\ f_{\text{post}}(t) &= s(t+\delta t) + n_2(t) \end{aligned}\right| \quad \begin{aligned} s(t) &: \text{Window of the signal} \\ n &: \text{Noise.} \end{aligned} \tag{5.1}$$

Actually, it is not just a time shift between the pre- and post-line, it is a stretch since the two lines are compressed versions of each other. However, for a small window of the signal the stretch will cause an apparent time delay, but some stretch will still be present (Sæbø et al., 2007). When we introduce a stretch factor $\alpha$, the two lines can be modeled as

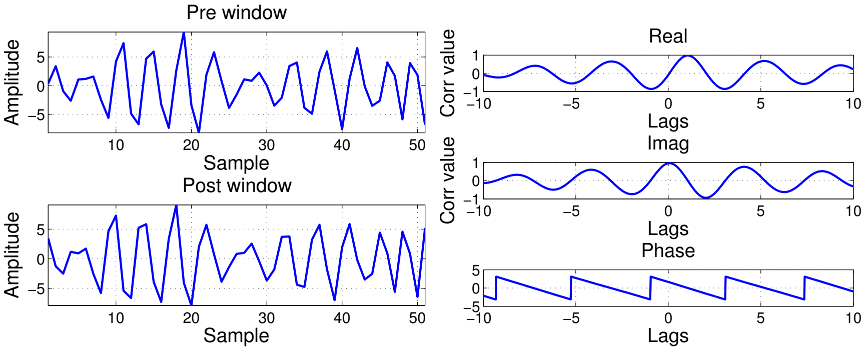
$$\left.\begin{aligned} \hat{f}_{\text{pre}}(t) &= s(t) + n_1(t), \\ \hat{f}_{\text{post}}(t) &= s\left(\frac{t+\delta t}{\alpha}\right) + n_2(t) \end{aligned}\right| \quad \begin{aligned} s(t) &: \text{Window of the signal} \\ n &: \text{Noise} \\ \alpha &: \text{Stretch factor.} \end{aligned}$$

To account for this stretch factor, we could use a so called cross-ambiguity function to estimate the displacement. This was done by Sæbø et al. (2007) to improve sidelooking sonar height estimation, which is related to the displacement estimation we are doing. For simplicity, and as we later will experience, we can assume $\alpha = 1$ in our displacement estimation and use the common crosscorrelation time delay estimation. The crosscorrelation of the two signals, from Equation (5.1), is

$$R_{f_{\text{pre}}f_{\text{post}}}(\tau) = E\{f_{\text{pre}}(t)f^*_{\text{post}}(t-\tau)\} = Rss(\delta t - \tau).$$

The autocorrelation has its maximum for $\tau = 0$, giving the crosscorrelation of $f_{\text{pre}}$ and $f_{\text{post}}$ its maximum for $\tau = \delta t$. Therefore we can find the maximum of the crosscorrelation to find the time shift between the two windows. To increase the accuracy of the estimation, we can interpolate the crosscorrelation result. Interpolating the signal introduce more quantization levels, making the displacement estimation more accurate. A derivation of the crosscorrelation time delay estimator as a maximum likelihood estimator is given in Appendix C.

To intuitively illustrate the crosscorrelation estimation we have plotted the window from the pre-compressed line, together with the window from the post-compressed line in Figure 5.3 (a). The result from the crosscorrelation is in Figure 5.3 (b).

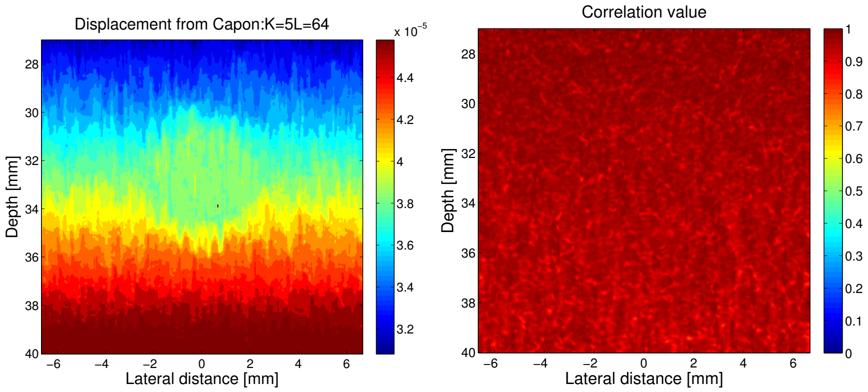


(a) Window of pre and post line.

(b) Correlation values of the crosscorrelation of the two windows.

*Figure 5.3: An example of the crosscorrelation displacement estimation. One window from the pre- and post- line is plotted in (a) where we see that the post line (bottom) is a bit shifted compared to the pre line (top). In (b) we have plotted to real (a), imaginary (b) and phase (c) of the crosscorrelation between the two windows. The top in the real, and the zero crossing of the imag and phase, indicate by which lag the two windows are separated*

The RF-signal from the Capon beamformer is, as we know from Section 3.3.3, the analytic version of the signal. In Figure 5.3 (b) we have plotted the real part, the imaginary part and the phase of the crosscorrelation of the two windows. We see that the maximum of the real part is at approximately 1 sample, while the imaginary part and the phase cross through zero at about 1 sample. The imaginary part of the analytic signal is the Hilbert transform, Appendix B, and it has been shown that when the crosscorrelation of a signal reaches a maximum its Hilbert transform passes through zero (Loupas et al., 1995). The phase of a signal $s(t)$ is defined as $\theta = \tan^{-1}\frac{Im[s(t)]}{Re[s(t)]}$, and thus the phase of the crosscorrelation will also be zero where the crosscorrelation of the Hilbert transform is zero. This allows an alternative, and maybe more efficient, implementation of the estimator; to find the zero of the Hilbert or the zero of the phase. This will later be suggested as further work. However, we have implemented finding the maximum of the interpolated version of the real part of the crosscorrelation.



(a) Displacement estimated with crosscorrelation

(b) Correlation values

*Figure 5.4: Displacement estimated with the crosscorrelation method (a), and the value of the maximum of the crosscorrelation (b).*

When the above estimation is done for every line and every sample on the pre and post compressed image we get the estimated displacement displayed in Figure 5.4 (a). It has also been shown that the SNR of the displacement estimate increase with an increasing correlation coefficient Souchon et al. (2003). Therefore a measure of the quality of the estimate is the value of the maximum of the crosscorrelation, this is plotted in 5.4 (b).

If we compare the estimated values to the model of estimation, Figure 5.1, we see that the estimation is fairly good. Notice that the color scale in the two images is slightly different since the image of the estimated displacement is just showing the region of interest, while in Figure 5.1 the black box is indicating the region of interest.

### 5.2.1 Window lengths

In the later comparison between the beamformers we will use three different windows lengths for the crosscorrelation displacement estimation. We will use 12, 25 and 50 samples in each window, this corresponds to approximately 0.34, 0.74 and 1.51 mm of length. For comparison we can remember back to the axial resolution in Equation (3.5), which we found was 0.2053 mm - actually equal to $\lambda$.

## 5.3 Pulsed-Doppler displacement estimation

We will once again return to Jensen's book (Jensen, 1996b) where we find that the further development of ultrasound imaging introduced the so called *Doppler imaging* mode. This imaging technique exploits the well known Doppler effect to detect moving objects that scatters ultrasound. This is first and foremost used for the detection of the velocity of blood flow. First it was developed a so called Continuous Wave (CW) Doppler system by Satomura (1957). The CW system transmits a continuously sinusoid into the tissue, and at the same time continuously records the backscattered waves. If the tissue is moving the backscattered sinus will be slightly shifted in frequency. This frequency shift, the Doppler frequency, is related to the velocity of the blood by

$$v = \frac{cf_d}{2f_c} \qquad \left| \qquad \begin{aligned} v &: \text{Velocity of scatterer} \\ f_d &: \text{Doppler frequency} \\ c &: \text{Wave velocity} \\ f_c &: \text{Transmitted center frequency} \end{aligned} \right. \tag{5.2}$$

derived from the Doppler equation

$$f_d = 2f_c\frac{v\cos(\theta)}{c} \qquad \left| \qquad \begin{aligned} \theta &: \text{Angle between velocity} \\ &\ \ \text{direction and ultrasound beam.} \end{aligned} \right. \tag{5.3}$$

The CW system has some drawbacks, especially the lack of range resolution. This means that for example two blood veins at two different depths can both shift the signal, so we will measure the wrong frequency. To cope with this problem a Pulsed Wave (PW) Doppler system was suggested by Baker (1970). This system transmits a short pulse of ultrasound, just as regular B-mode ultrasound, this pulse is backscattered and at least two measurments is used to find the displacement of the backscattered signal, as a consequence of the movement of blood, between the measurments. This means that the PW Doppler system is actually not using the Doppler effect at all. It is just finding the displacement of the scatterers. Finding the displacement is exactly what we want to do - so the *pulsed-Doppler* methods should be able to be used when doing static elastography. This has been done by for example Børstad (2011) in his master's thesis *Interoperative Ultrasound Strain Imaging of Brain Tumors* - where he uses PW Doppler techniques to visualize brain tumors. The main difference between our applications and his is that he used displacement in the tissue caused by the pulsation of arteries, while static elastography uses compression by the ultrasound probe.

### 5.3.1 Autocorrelation method

In 1985 Barber et al. published *A New Time Domain Technique for Velocity Measurements Using Doppler Ultrasound*. This technique was able to estimate the Doppler frequency using the autocorrelation function. The estimation is done in the time domain, allowing a very fast implementation. This technique was the basis of the first real-time blood flow imaging system demonstrated by Kasai et al. (1985). The thoroughly mathematical analysis of the technique was done by Angelsen and Kristoffersen (1983). We will not touch the details here, but the central equation is;

$$\hat{f} = \frac{\angle R(0,1)}{2\pi T_{pr}} \quad \left| \quad \begin{array}{rl} \angle R(0,1) : & \text{Phase angle of the autocorr at frame lag 1} \\ T_{pr} : & \text{Time between pulse returns.} \end{array} \right. \tag{5.4}$$

Where $R(0,1)$ is the autocorrelation function between the pre- and post-frame, which can be defined as (Børstad (2011))

$$R(\check{m}, \check{w}) = \sum_{m=0}^{M-\check{m}-1} \sum_{v=0}^{N-1} \sum_{w=0}^{O-\check{w}-1} x_a(m, v, w) x_a^*(m + \check{m}, v, w + \check{w}), \tag{5.5}$$

where

$$\begin{array}{rl} M : & \text{Depth samples (axial direction)} \\ \check{m} : & \text{Axial lag} \\ N : & \text{Lateral samples (lateral direction)} \\ O : & \text{Number of frames} \\ \check{w} : & \text{frame lag} \\ x_a : & \text{Analytic signal along the axial dimension.} \end{array}$$

If Equation (5.4) is inserted into Equation (5.2) we get the estimate of the axial velocity:

$$v_a = \frac{c \angle R(0,1)}{4\pi f_c T_{pr}} \quad \left| \quad \begin{array}{rl} \angle R(0,1) : & \text{Phase angle of the autocorr at frame lag 1} \\ T_{pr} : & \text{Time between pulse returns} \\ f_c : & \text{Center frequency.} \end{array} \right. \tag{5.6}$$

Since we only need to estimate the displacement, the time is not important, we can skip $T_{pr}$ giving us the slightly simpler axial displacement estimate

$$d_{\text{auto}} = \frac{c \angle R(0,1)}{4\pi f_c} \quad \left| \quad \begin{array}{rl} \angle R(0,1) : & \text{Phase angle of the autocorr at frame lag 1} \\ f_c : & \text{Center frequency.} \end{array} \right. \tag{5.7}$$

### 5.3.2 Modified autocorrelation method

The center frequency is attenuated proportional to the depth in ultrasound imaging. In 1995 Loupas et al. suggested to include an estimate of the center frequency, and called this the 2D autocorrelator. We will call it the modified autocorrelation method. Loupas et al. showed that the center frequency can be estimated by

$$\hat{f}_c = \frac{\angle R(1,0)}{2\pi} f_s \quad \left| \quad \begin{aligned} \angle R(1,0) &: \text{Phase angle of the autocorr at axial lag 1} \\ T_{pr} &: \text{Time between pulse returns.} \end{aligned} \right. \tag{5.8}$$

If we include this estimate in the autocorrelation method in Equation (5.7) we get the modified autocorrelation estimation of the displacement as

$$d_{\text{mod auto}} = \frac{c\angle R(0,1)}{4\pi \hat{f}_c} \quad \left| \quad \begin{aligned} \angle R(0,1) &: \text{Phase angle of the} \\ & \quad \text{autocorr at frame lag 1} \\ T_{pr} &: \text{Time between pulse returns.} \end{aligned} \right. \tag{5.9}$$

Loupas et al. reported that this method outperformed the former method, and this result was verified by Børstad (2011). However in (Blomberg, 2005) the modified autocorrelation technique only gave improved results on simulated data, not on experimental data.

### Number of samples in the estimate

The implementation of the pulsed-Doppler estimators used in this thesis is based on the implementation used by Børstad (2011). Børstad introduced to include samples from N scan lines. This gives a more accurate autocorrelation estimate at the cost of lower lateral resolution. In his implementation he uses a filter on the calculated $R(0,1)$ and $R(1,0)$ to include more samples in the estimate. We also use this filtering with a *estimation window* were we use $M = 15$ samples axially, and $N = 4$ samples laterally for our estimation window for all pulsed-Doppler estimates in this thesis. This is somewhat different from the crosscorrelation method where we only use samples from the axial lines to create the estimate and do not share information between the lines. This gives the pulsed-Doppler method an advantage compared to the crosscorrelation method, but it also gives the pulsed-Doppler estimation worse lateral resolution. An estimation window of $M = 15$ samples axially and $N = 4$ samples laterally means that the pulsed-Doppler estimation includes information from $M \times N = 60$ samples, which is about the same as the 50 samples used in the largest window of the crosscorrelation displacement estimation. In this thesis we are only using $O = 2$ frames in our pulsed-Doppler estimates, this is often referred to as a *packet size* of 2.

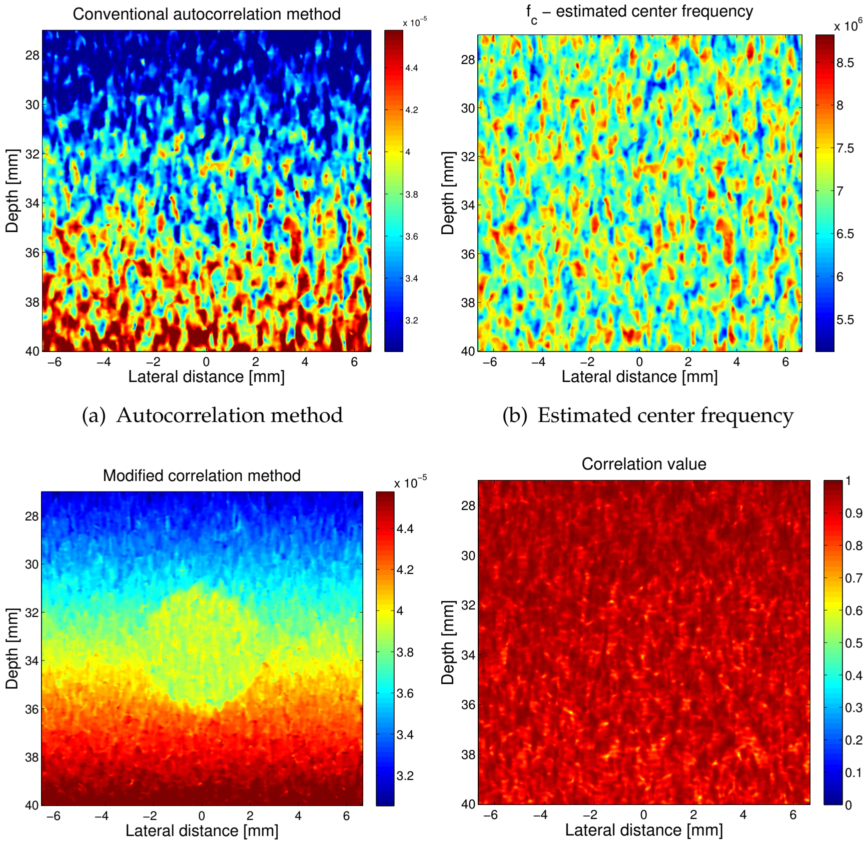
However, we are not comparing the displacement estimation methods, we are comparing the beamforming done prior to the displacement estimation - and thus we only need to use the same parameters in the displacement estimation for both beamformers. The implementation of both the autocorrelation and the modified autocorrelation is provided in Appendix A.3.

### 5.3.3 Correlation coefficients as an estimation quality indicator

Børstad (2011) suggested using the ratio of the autocorrelation at one lag over the autocorrelation at zero lag as a estimation quality indicator.

$$c = \frac{O}{O-1}\left|\frac{R(0,1)}{R(0,0)}\right|. \tag{5.10}$$

The scaling factor $\frac{O}{O-1}$, where $O$ is number of frames, ensures that $c = 1$ if the two frames compared are stationary, if it has been no compression. The factor $c$ will decrease towards zero with increasing signal decorrelation. This estimation quality indicator will be useful when we in the next chapter start to compare different parameters and beamformers.



(a) Autocorrelation method

(b) Estimated center frequency

(c) Modified autocorrelation method

(d) Correlation values

*Figure 5.5: Displacement estimation from the pulsed-Doppler methods. The conventional autocorrelation (a), the estimated center frequency (b), the modified autocorrelation method (c) and the correlation coefficients (d) suggested by Børstad (2011) are plotted.*

In Figure 5.5 (a) we have plotted the displacement estimation from the autocorrelation method, in (b) Equation (5.8) the estimation of $f_c$ is plotted, in (c) Equation (5.9) the estimation of the modified autocorrelation method is used. In (d) we have plotted the correlation coefficients suggested by Børstad (2011) as a quality indicator of the estimation. The data used is the same as in Figure 5.4.

From the plots we see that the center frequency, Figure 5.5 (b), does not attenuate proportional to the depth. This is because Field II does not simulate this attenuation. However, the modified autocorrelation method is still better than the conventional. We see that the center frequency does fluctuate a bit above and below the transmitted frequency of 7.5 MHz. This is probably because of the signal interference in the speckle pattern, and it does mean that the apparent center frequency does fluctuate a bit. The modified autocorrelation accounts for this fluctuation, which will influence the displacement estimation, and it's probably therefore this modified estimation method outperformes the conventional, even though the Field II simulations do not include frequency attenuation. When we later refer to the pulsed-Doppler displacement estimation we mean the modified autocorrelation method.

### 5.3.4 Limitations of Pulsed-Doppler techniques

The pulsed-Doppler techniques looked at so far uses the angle of the autocorrelation, and is thus limited by the uniqueness of the inverse trigonometric functions (Jensen, 1996b). The autocorrelation technique therefor has a maximum detectable velocity

$$v_{max} = \frac{c}{4}\frac{1}{f_c T_{pr}}. \tag{5.11}$$

This gives us a maximum displacement of

$$d_{max} = \frac{c}{4}\frac{1}{f_c} = \frac{\lambda}{4} = \Delta, \tag{5.12}$$

where $\lambda$ is the wavelength of the center frequency. This is what we defined as our reference displacement $\Delta$ in Section 5.1 and is why we later will not use the pulsed-Doppler displacement estimation on displacements lager than $\Delta$.

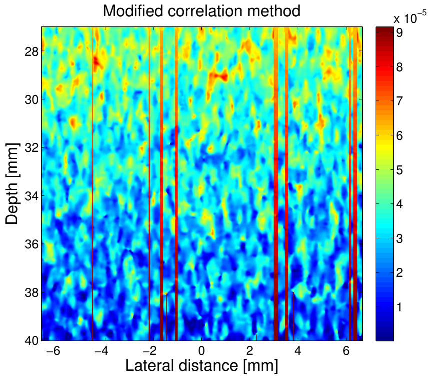


*Figure 5.6: Pulsed doppler with $2\Delta$ compression clearly showing that the pulsed doppler method gets aliasing if the compression is too large. The red vertical lines in the images are some other error artifact originating from the aliasing.*

In Figure 5.6 we have plotted the modified autocorrelation displacement estimation with a compression of $2\Delta$, and as we clearly see from the image we have aliasing causing clearly the wrong estimation. The red vertical lines in the image are some other error artifact also originating from the aliasing.

The $\Delta$ factor is only 0.05 mm for a 7 MHz center frequency. This is a very small compression. However, this limit is between two frames. A scanner usually has a frame rate of e.g. 50 Hz, and multiple frames can easily be included in the estimation. In this thesis we will stick to two frame scenarios since we are only interested in the comparison of the two different beamformers.

## 5.4 Comparison criteria

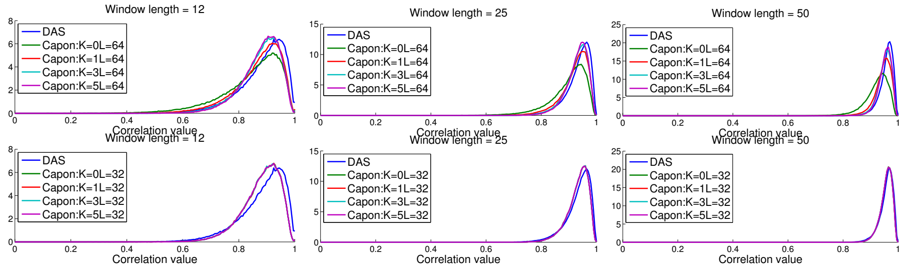
To thoroughly investigate the difference between the beamformers, the main goal of this thesis, we need quantitative comparison criteria. Therefore, we will introduce and evaluate two criteria; the correlation values, and the error between the estimated displacement result and the displacement model applied in the simulations.

### 5.4.1 Correlation values

We have already introduced the correlation values as an estimation quality indicator in the previous sections, but a closer investigation is favorable.

#### Crosscorrelation displacement estimation

For the crosscorrelation displacement estimator we use the direct correlation coefficient from the maximum of the crosscorrelation - which we know indicate the lag and thus the displacement. This value was plotted in Figure 5.4.



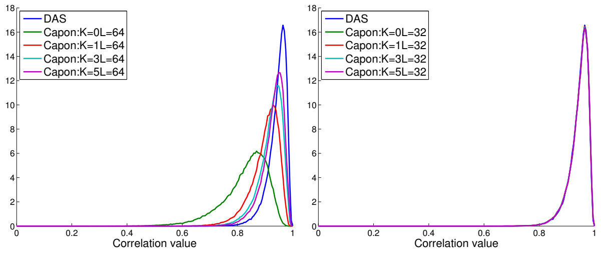
(a) Axial window length = 12 samples (b) Axial window length = 25 samples (c) Axial window length = 50 samples

*Figure 5.7: Correlation values from crosscorrelation estimation for different axial window lengths. The top plots is with the Capon beamformer with $L = 64$ for different $K'$s, while to bottom plots is for the Capon beamformer with $L = 32$ with different $K'$s - both compared to the same DAS beamformer.*

In Figure 5.7 we have plotted the distribution of the correlation values for the simulation with $\Delta$ compression for three different window lengths, 12, 25 and 50 samples, and different parameters for the Capon beamformer and for the DAS beamformer. In the plots we see that for the Capon beamformer with $L = 64$ (top plots) we get higher correlation the more samples, higher $K$, we use in the estimation of the spatial correlation matrix. When $K = 5$ we have approximately the same correlation as DAS, but DAS is still slightly better. We see that this is true for all three window lenghts. However, the longest windows give the highest correlation values. For the Capon beamformer with $L = 32$ (bottom plots) we see that the correlations are not much influenced by a larger $K$. This is in agreement with the Capon parameters giving the same speckle statistics as DAS, from Section 3.4.3. We will discuss this result in detail in the next chapter.

### Pulsed-Doppler displacement estimation

For the pulsed-Doppler displacement estimator we will use the correlation coefficient as suggested by Børstad (2011), covered in Section 5.3.3. In Figure 5.8 we have plotted the distribution of the correlation values for the simulation with $\Delta$ compression for different parameters for the Capon beamformer and for the DAS beamformer where the displacement is calculated with the pulsed-Doppler estimator. For the pulsed-Doppler estimation we are not using windows in the same sense as in the crosscorrelation estimation, see Section 5.3.2, and thus we have fever plots for this estimator compared to the crosscorrelation estimator, Figure 5.7.



(a) Capon subarray length $L = 64$

(b) Capon subarray length $L = 32$

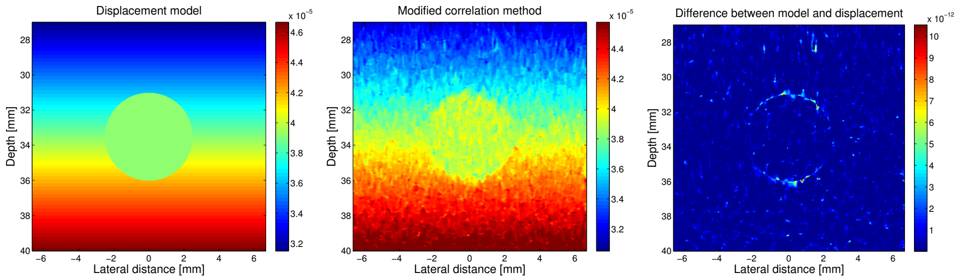
*Figure 5.8: Correlation values from pulsed-Doppler estimation. The Capon beamformer with $L = 64$ and different $K'$s is in the plot to the left (a), while the Capon beamformer with $L = 32$ for different $K'$s is plotted to the right (b).*

The trend is the same as for the crosscorrelation estimator, that the Capon beamformer with $L = 64$ gets better correlation with higher $K$ - while the $K$ does not influence much for the Capon beamformer with $L = 32$.

### 5.4.2 Error between simulated displacement model and estimated displacement

To compare the estimated displacement to the model of the displacement applied in the simulation we calculate the sum of squared difference

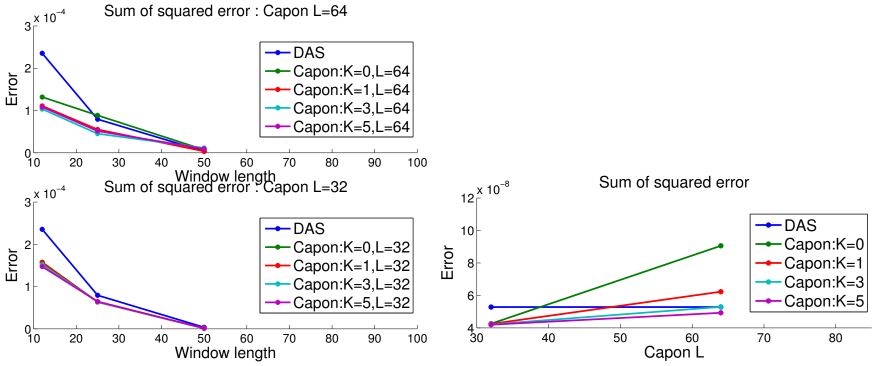
$$e = \sum_x \sum_y |(model - estimated)^2|. \tag{5.13}$$



(a) Displacement model (b) Estimated displacement (c) Error

*Figure 5.9: We have displayed the displacement model used in the simulation (a), the displacement estimated from the simulations (b) and in (c) the error between the two.*

In Figure 5.9 we have plotted the model of the displacement (a), the estimated displacement (b) and the error between the two (c). The value of the error from Figure 5.9 (c) is $e = 4.93 \cdot 10^{-8}$.

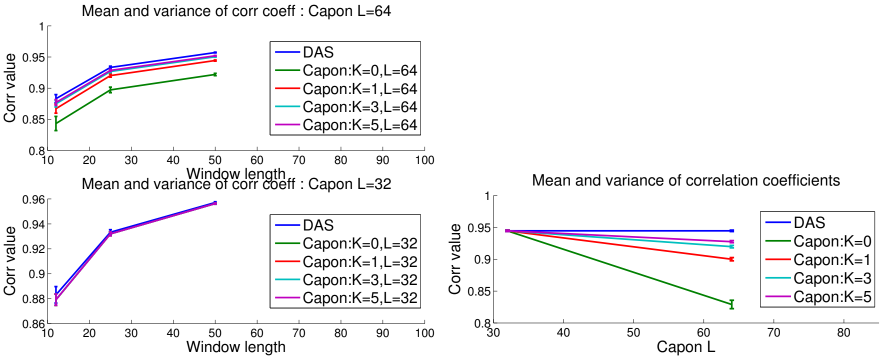


(a) Error values from crosscorrelation estimation (b) Error values from pulsed-Doppler estimation

*Figure 5.10: The error from the crosscorrelation estimation (a) and the pulsed-Doppler estimation (b).*

The error values for both beamformers with different parameters for the simulation with Δ compression is plotted in Figure 5.10. The errors from the crosscorrelation estimation are plotted in 5.10 (a) where we along the x-axis plot the three different window lengths, and the different colored lines indicate different parameters. The top plot is Capon with subarray length $L = 64$, the bottom is $L = 32$.

The plot in Figure 5.10 (b) is for the pulsed-Doppler estimation. Where the x-axis indicate if it is Capon with $L = 32$ or $L = 64$, and the different colored lines indicate what $K$ is used. The DAS is independent from the subarray, so the same value is plotted at both x-axis positions. These plots display the information about the correlation in a illustrative and intuitive manner where the lowest value is the best. Once again we see that a high $K$ for Capon $L = 64$ gives less error, while for Capon with $L = 32$ it is already quite equal to DAS.



(a) Correlation values from crosscorrelation estimation

(b) Correlation values from pulsed-Doppler estimation

*Figure 5.11: Correlation values from the crosscorrelation estimation (a), and the pulsed-Doppler esimtation (b). These plots are mean-variance plots of the same values plotted as the distribution in Figure 5.8.*

In the same sense we can compress the information from the correlation values by producing mean-variance plots of the values. This is done in Figure 5.11, where as in Figure 5.10, the crosscorrelation estimation has window length along the x-axis, Figure 5.11 (a), while the pulsed-Doppler estimation only have one window length so one plot with the Capon subarray length $L$ along the x-axis is sufficient. The plots themselves are not the only thing similar between Figure 5.10 and Figure 5.11. If we closely investigate and compare the information in the plots we will see that there is a clear trend; higher correlation values results in lower error. This is as expected, as shown in Souchon et al. (2003), and will allow us to reduce the redundant information from each simulation and only investigate the error of the estimation as the main comparison criteria in the next sections. This will prove useful in the next chapter where we will investigate quite a large number of simulations

## 5.5 Strain

So far we have looked at the displacement of the tissue. In elastography it is the strain of the tissue that is displayed in the elastogram. In Section 2.2.1 we introduced the strain as the gradient of the time delay at two different spatial locations

$$\epsilon = \Delta t / T. \tag{5.14}$$

The displacement is proportional to the time delay and we can therefor also use the estimated displacement.

### 5.5.1 Least square strain estimation

A more robust approach is to calculate the strain as the slope of the regression line of all displacement estimations

$$d[m] = am + b, \tag{5.15}$$

where the strain $\epsilon_m$ will be proportional to $a$. Equation (5.15) can be written on matrix form as

$$\begin{bmatrix} d[m_u] \\ d[m_u+1] \\ \vdots \\ d[m_l] \end{bmatrix} = \begin{bmatrix} m_u & 1 \\ m_u+1 & 1 \\ \vdots & \vdots \\ m_l & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \Leftrightarrow \mathbf{d} = \mathbf{A} \begin{bmatrix} a \\ b \end{bmatrix} \tag{5.16}$$

We have the estimated displacement matrix $\hat{\mathbf{d}}$, and can create the matrix $\mathbf{A}$. So by using the well known least-square solution

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = [\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\hat{\mathbf{d}}, \tag{5.17}$$

we can find the local strain of the tissue.

**Optimized least square strain estimations**

However, Børstad (2011) suggested a clever optimization for the least square strain estimation. The optimization builds on the fact that we only need the slope of the line, $a$, in Equation (5.15) and not the offset. Børstad then showed that the least-squares solution can be found by convolution. We will follow his arguments. We assume that $\Delta m$ is chosen so that all the velocity estimates will be used in the strain estimation. This gives us $\Delta m = n + 1$, and the matrix A from Equation (5.16) becomes

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ \vdots & \vdots \\ n & 1 \end{bmatrix}$$

giving us

$$[\mathbf{A}^\mathbf{T}\mathbf{A}]^{-1} = \frac{12}{n(n^2-1)} \begin{bmatrix} 1 & -\frac{n+1}{2} \\ -\frac{n+1}{2} & \frac{(n+1)(2n+1)}{6} \end{bmatrix}.$$

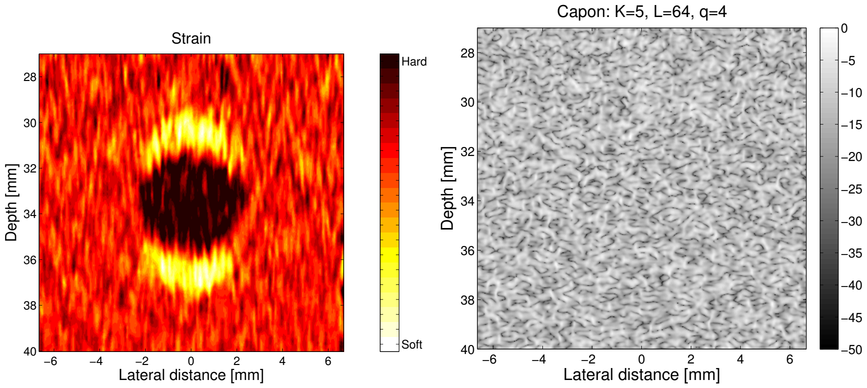
The trick is that we only need the first row of $[\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T$ to calculate $\hat{a}$. This first row is a vector, $h$:

$$\begin{aligned} h &= \frac{12}{n(n^2-1)} \begin{bmatrix} 1 & -\frac{n+1}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 1 & \dots & 1 \end{bmatrix} \\ &= \frac{12}{n(n^2-1)} \left( \begin{bmatrix} 1 & 2 & \dots & n \end{bmatrix} - \begin{bmatrix} \frac{n+1}{2} & \frac{n+1}{2} & \dots & \frac{n+1}{2} \end{bmatrix} \right) \end{aligned}$$

Now, we can create a flipped and transposed version of the vector $h$;

$$\hat{h} = \frac{12}{n(n^2-1)}\left(\begin{bmatrix} n & n-1 & \dots & 1 \end{bmatrix} - \begin{bmatrix} \frac{n+1}{2} & \frac{n+1}{2} & \dots & \frac{n+1}{2} \end{bmatrix}\right)^T.$$

This flipped and transposed version, $\hat{h}$, can be convolved with the displacement matrix, $\mathbf{d}$, to find the least square solution of $a$ for a segment of length $\Delta m = n - 1$. Børstad reports a speedup of approximately 500 using this optimized implementation, instead of the full least square implementation.



(a) Strain

(b) Ultrasound image

*Figure 5.12: The strain clearly shows the hard lesion in the middle of the image not visible in the ultrasound image. This setup is ideal, since we assume that the tissue has homogeneous reflection coefficients while the elasticity is very different.*

In Figure 5.12 (a) we have plotted the strain estimated with the optimized least square estimation, with $\Delta m = 60$, from the displacement values estimated with the modified autocorrelation method, Figure 5.5 (c). Figure 5.12 (b) displays the ultrasound B-mode image, were we see that there are no indications of difference in the tissue, while the strain image clearly shows a hard lesion in the middle of the image. This example is of course ideal, since we assume that the tissue has homogeneous reflection coefficients while the elasticity is very different. However, the example clearly illustrates the advantages of elastography.

## 5.6 Summary

In this chapter we have investigated the details of static elastography. We have seen how we can easily simulate static elastography by moving the scatterers in our simulation phantom to simulate compression. We have seen two different displacement estimation techniques, the crosscorrelation technique and the pulsed-Doppler technique. We have seen that the pulsed-Doppler can be done more sophisticated by also estimating the center frequency and called this the modified autocorrelation method and seen how the correlation values can be used as a quality indicator of the estimation. We have investigated two comparison criteria to compare the two beamformers; correlation values, and error between simulated displacement model and estimated displacement. We also saw that the correlation and error seemed to be displaying the same information, so we have chosen to use the error to compare the two beamformers in the next chapter. In the last section the least square strain estimation was introduced, and also a optimized version of the least square estimation, and we saw how elastography can display tissue characteristics not visible in regular ultrasound images.

# Chapter 6

# Results

***Chapter abstract:*** *This chapter uses the comparison criteria defined in the previous chapter to evaluate adaptive beamforming compared to conventional beamforming when applied to displacement estimation, and thus static elastography. The two beamformers are compared and evaluated before we discuss the result and suggest an explanation for the results.*

In this chapter we will reap the results from all our previous discussions and results and finally investigate our overall goal; *does adaptive beamforming have benefits when applied to static elastography.*

## 6.1 Summary of the simulations executed

In Chapter 3 we went through all the details of how we executed the simulation of ultrasound images. In Chapter 5 we presented the estimation methods and details on how we simulated the compression of tissue, and defined comparison criteria to compare the two beamformers. We saw that the correlation values and the error are highly correlated, so we will only use the error between the displacement estimation and the model of the displacement in our final evaluation. To thoroughly test our research question on whether adaptive beamforming has benefits when doing static elastography, we need to simulate as many setups and different displacements as possible. In this section we will go through all the different simulation setups we have used to tested our hypothesis.

We concluded in Chapter 3 that 100 000 scatterers in the phantom was enough to simulate well developed speckle, and that a lateral oversampling factor of $q = 4$ was sufficient for our setup when imaging well developed speckle. In Chapter 5 we introduced the $\Delta = \frac{\lambda}{4}$ amount of compression.

Each separate simulation consists of creating two images, pre- and post-compression, of the same phantom where the phantom has been compressed by an amount ($\frac{1}{2}, 1, 2$ or 4 times $\Delta$). Each image is created with both the DAS beamformer and the Capon beamformer. For the Capon beamformer we have created separate images from both $L = 32$ and $L = 64$ with $K = 0, 1, 3, 5$. In total that gives us 18 ultrasound images from each phantom, and thus 9 different static elastography calculations from each simulated compression. On these data we have used the crosscorrelation displacement estimator on all simulated compressions, while we have only used the pulsed-Doppler estimator for $\Delta$ and $\frac{\Delta}{2}$ compression, because of the aliasing problem discussed in Section 5.3.4. The crosscorrelation displacement estimation is ran with three different window lengths 12, 25 and 50 samples.

### 6.1.1 Dynamic transmit focus

In Chapter 3 we used dynamic transmit focusing when creating the speckle images. This is of course very ideal, and is hard and time consuming to do in real time. However, it is interesting to see how the beamformers compare in an ideal setting so we have used dynamic transmit focus when creating the image.

### 6.1.2 Fixed transmit focus

To also create images as they are generally created in the scanners, we have also made the same simulated compression and beamformers where we have used a fixed transmit focus at the middle of the inclusion at 33.5 mm depth.

### 6.1.3 Noise

The simulated data are ideal and noise free. This is not the case in real life, so we have also executed all the simulations where we have added noise to the signals from the elements before beamforming. The noise added was white filtered noise, where we filtered the noise with a bandpass filter approximately equal to the bandwidth of the signal. The white noise was generated with the function *randn* in MATLAB, and the filter was created as a 10'th order Butterworth filter, with lower cutoff frequency at 5 MHz and higher cutoff frequency at 10 MHz. The frequency spectrum of the signal from one element, the frequency spectrum of the noise and the frequency spectrum of the signal with added noise is in to top plot in Figure 6.1. The middle plot in the figure displays the signal and the signal with added noise. The signal to noise ratio was $\approx 0.5dB$ before beamforming. In the bottom plot we have plotted one RF-line after DAS beamforming and we see that the signal gain of the beamformer suppresses most of the noise.

### 6.1.4 Example images of displacement

Many simulations have been created, we cannot display all estimations by images, so we have chosen to display a few examples to show how the added noise and fixed transmit focus influence the estimations.

In Figure 6.2 we show an example of the crosscorrelation displacement estimation both with (b) and without (a) noise added to the element data. If we compare the two images we see that the estimation is still fairly good with added noise. The noise has influenced the estimation, it is more noisy

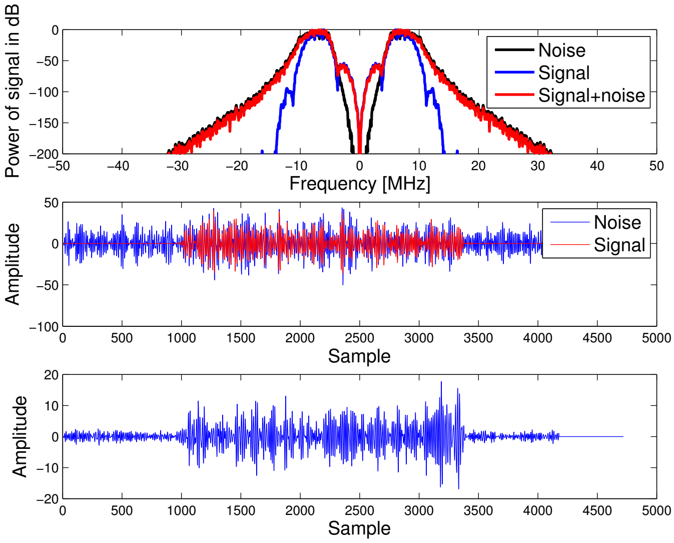
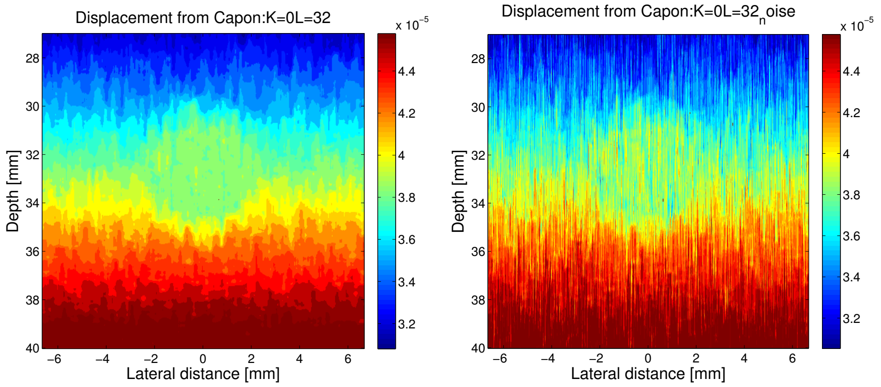


*Figure 6.1: **Top:** frequency spectrum of the noise, signal from one element and signal from one element with added noise. **Middle:** Signal from one element and signal plus noise. **Bottom:** One beamformed line.*



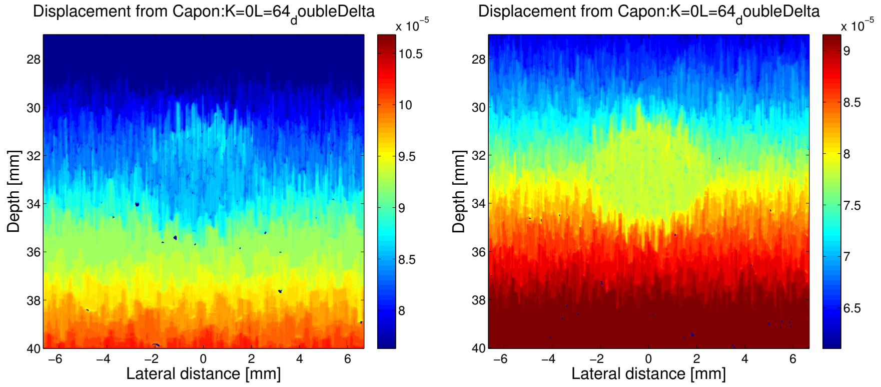
(a) Without noise

(b) With noise

*Figure 6.2: Crosscorrelation displacement estimation from dynamic focus, Capon beamforming with $K = 0$ and $L = 32$ with $\Delta$ compression simulated. To the left (a) is the estimation with no noise, and to the right (b) is the simulation with added noise.*

and thus more error has been added, but we still see the contours of a lesion in the middle with constant displacement. We will compare these differences more quantitatively in the later sections.

In Figure 6.3 we have plotted another example of the crosscorrelation displacement estimation, but now with $2\Delta$ simulated compression. We have displayed both the estimation from the ultrasound images with fixed transmit focus and the ideal dynamic transmit focus. From these images we see that the dynamic transmit focus does perform better than the fixed transmit. This is reasonable since the fixed transmit gives a better resolution throughout the whole image instead of just good resolution at one small portion of the image as fixed transmit does. However, the fixed



(a) Crosscorrelation displacement estimation from fixed transmit focus with 2Δ compression

(b) Crosscorrelation displacement estimation from dynamic transmit focus with 2Δ compression

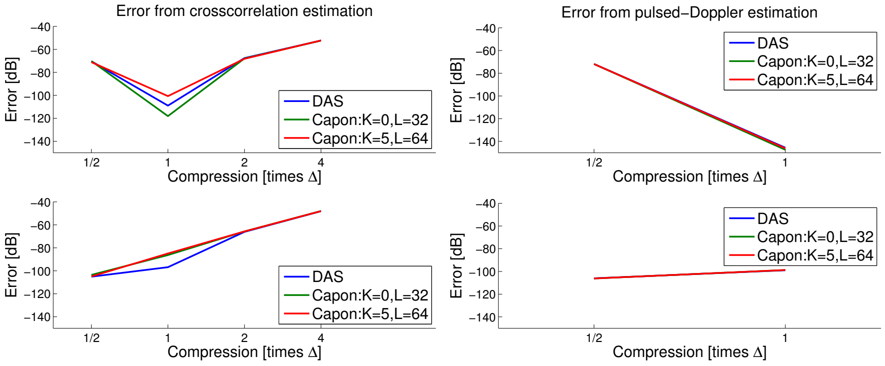
*Figure 6.3: Crosscorrelation displacement estimation from fixed transmit focus (a) and dynamic transmit focus (b) - both with Capon beamforming with $K = 0, L = 64$ and 2Δ as simulated compression.*

transmit does show the inclusion in the middle with constant displacement - so arguably the displacement estimation is still fairly good.

## 6.2 Evaluation

With all the simulations described in the previous section the amount of information created is overwhelming. A major challenge is to compress the information to easily evaluate our hypothesis. The plots, of the type in Figure 5.11 and 5.10, from each simulation displaying the mean and variance of the correlation values and the error values are placed in Appendix D. If we investigate the plots of the correlation and the error we see the trend that we pointed out in Section 5.4.1; that higher correlation gives less error. The parameters giving both the highest correlation and the lowest error for the Capon beamformer are $L = 32$ with $K = 0$ and $L = 64$ with $K = 5$. If we remember back to Section 3.4.3 this is actually the parameters that gives the Capon beamformer approximately equal speckle statistics as the DAS beamformer. We will discuss this result in detail in the next section.

The fact that these parameters generally seems to be the best for Capon, allows us to only choose these parameters for Capon when we do the final evaluation of the beamformers. In the later analysis we will find that Capon gets similar performance as DAS only with the parameters giving the best results for Capon, the analysis is therefore simpler if we only use the parameters giving the best performance for Capon. If we further investigate the plots in the appendix we will see that the longest window used for correlation, 50 samples, gives the best result. Based on these observations we can extract only these data and compare them for different compression simulations both with and without noise.

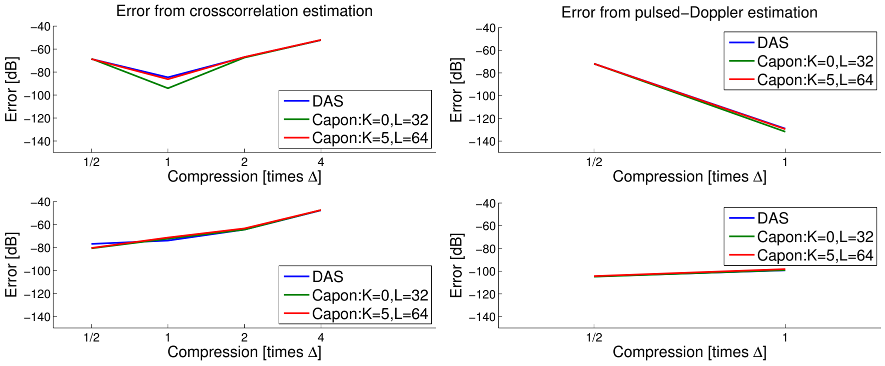


(a) Crosscorrelation: **Top:** dynamic transmit, **bottom:** fixed transmit.

(b) Pulsed-Doppler: **Top:** dynamic transmit, **bottom:** fixed transmit.

*Figure 6.4: The error from the crosscorrelation displacement estimation (a) with 50 samples window length, and the pulsed-Doppler displacement estimation in (b). The top plots are from dynamic transmit focus, the bottom plots from fixed transmit focus. These plots are from simulations without noise.*

In Figure 6.4 (a) we have plotted the error from the crosscorrelation estimation with 50 sample window length, and in (b) the error from the pulsed-Doppler estimation. The top plots are for the images created with dynamic transmit, and the bottom plots are from the images created with fixed transmit. We see that the errors are close to equal for both the beamformers, but for Δ compression the Capon $L = 32, K = 0$ beamformer is slightly better than DAS for crosscorrelation estimation when dynamic focus is used, while DAS is slight better for the crosscorrelation estimation for fixed focus. For the pulsed-Doppler estimation the results are very similar.



(a) Crosscorrelation: **Top:** dynamic transmit, **bottom:** fixed transmit

(b) Pulsed-Doppler: **Top:** dynamic transmit, **bottom:** fixed transmit

*Figure 6.5: The error from the crosscorrelation displacement estimation (a) with 50 samples window length, and the pulsed-Doppler displacement estimation in (b). The top plots are from dynamic transmit focus, the bottom plots from fixed transmit focus. These plots are from simulations with noise as described in Section 6.1.3*

In Figure 6.5 we have plotted the same simulations as in Figure 6.4, but now we have plotted the error values from the calculations where noise has been added to the data. From the plots we see that the noise has created higher error in the estimations, but it has not influenced one beamformer more than the other - the performance of the two beamformers are still very similar.

If we continue our investigation of the plots in the appendix, one thing we will notice is that the Capon beamformer seems to perform relatively best, compared to the DAS beamformer, for the crosscorrelation displacement estimation at the shortest window length, 12 samples, when noise has been added to the simulations.

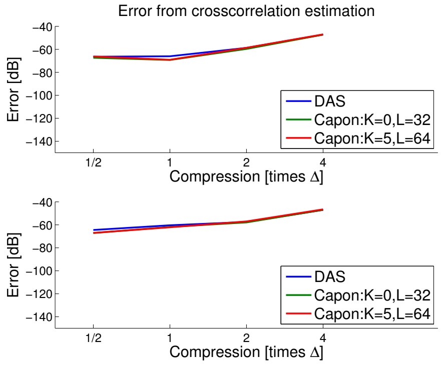
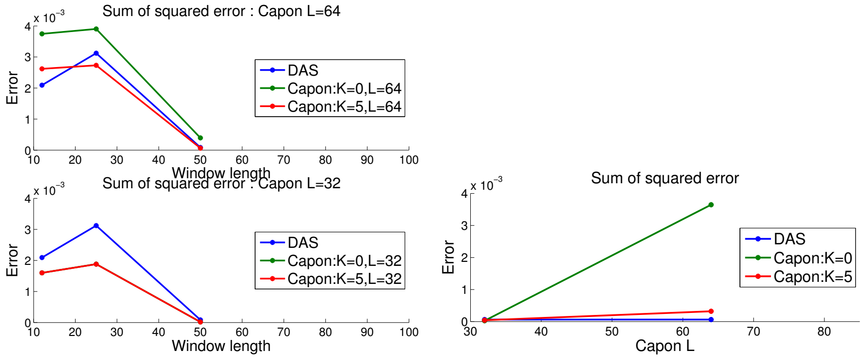


*Figure 6.6: Error from crosscorrelation estimation with noise for 12 samples windowlength.* ***Top:*** *dynamic transmit,* ***bottom:*** *fixed transmit.*

In Figure 6.6 we have plotted the error from the different compression simulations for precisely this observation. From the plot we see that indeed, the Capon beamformer gets less error than DAS - but it is just slightly and the error is, for most of the simulations, very similar.

To make sure our previous assumptions that the lateral oversampling factor of $q = 4$ is sufficient to not lose information when imaging speckle with Capon beamforming, Section 3.4.3, and that decimating the signal by a factor 4, Section 3.3.3, did not interfere with the result we did a final simulation. In this simulation we used a lateral oversampling factor of $q = 16$ and did not decimate the signal, meaning that we had 4 times the amount of data both in axial and lateral dimension. However, this information should not interfere with the results - and it did not.

In Figure 6.7 we see that this simulation follows the previous pattern of Capon with $L = 32$ and Capon with $L = 64, K = 5$ giving similar results as DAS, and actually slightly better for the crosscorrelation displacement estimation with $L = 32$. The reason for not using more and different parameters for the Capon beamformer in this simulation is the overwhelming simulation time, approximately 56 hours (see Appendix A), we get with 2048 lines in one ultrasound image.



(a) Error values from crosscorrelation estimation

(b) Error values from pulsed-Doppler estimation

*Figure 6.7: The error from the crosscorrelation estimation (a) and the pulsed-Doppler estimation (b). This if from a simulation with no decimation and with lateral oversampling factor* $q = 16$.

## 6.3 Discussion

The first result we will discuss is the interesting fact that the Capon parameters producing similar Rayleigh distributed speckle statistics as DAS, also produces the best estimation results for the Capon beamformer. This is true for both displacement estimation methods. This is interesting because our former intuition was that the more peaky and distinct speckle pattern from the Capon beamformer, should result in easier correlation and better estimation. However, our results show that it is the more *smooth* Rayleigh distribution speckle pattern that gives the best results.

One possible explanation can be derived from our previous detailed discussion of the Capon beamformer. In Section 3.3.1 we saw that the Capon beamformer adapts the beampattern to the received signal. The adaptation means that it suppresses some *unwanted* signals, and favors other signals - as we saw in the example with Sparrow's resolution limit. This also means that the Capon beamformer results in *variable resolution*. This could mean than when the Capon beamformer creates the image of one part of the tissue before and after compression, the same speckle might be imaged with two slightly different beampatterns. Then the same speckle, or part of tissue, will be imaged differently in the post-compression than in the pre-compression image, leading to less correlation and thus worse estimation. The adaptive behavior might give the result that *peaks* in the pre-compression image do not occur as peaks in the post compression image, because the adaptive behavior will favor a peak at a different position.

When we restrict the adaptive behavior of the Capon beamformer by using a smaller subarray, shorter $L$, or adding more samples in the estimation of the spatial covariance matrix, larger $K$, we probably get a more similar beampattern for the pre- and post-compression image when imaging the same *speckle* in the image.

To test this explanation we have plotted the beampattern for the Capon

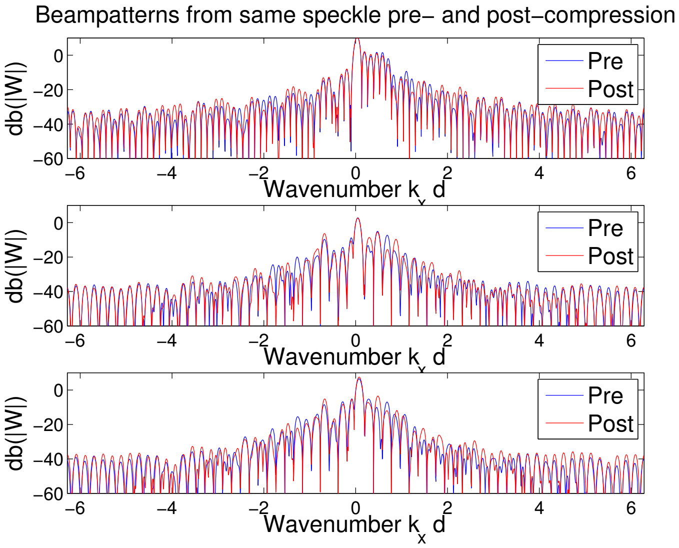


*Figure 6.8: Pre- and post-compression beampatterns for the Capon beamformer.* ***Top:*** *$L = 32, K = 0$,* ***middle*** *$L = 64, K = 5$, and* ***bottom:*** *$L = 64, K = 0$.*

beamformer with $L = 32, K = 0$ (top), $L = 64, K = 5$ (middle) and $L = 64, K = 0$ (bottom) in Figure 6.8. We have done this by taking the Fourier transform of the weights resulting from a given sample at a given depth in the pre-compression image and found how much that particular speckle has moved in the post-compression image by the crosscorrelation displacement estimation, and then plotted the beampattern from the same speckle after compression. From the plots we see that it is not much difference between the beampatterns. However, when we calculate the sum of squared difference between the pre- and post-compression beampatterns, we see a difference than can support our hypothetical explanation. The sum of squared difference is

$$
\begin{aligned}
diff_{\text{beampattern } L=32,K=0} &= 34.28 \\
diff_{\text{beampattern } L=64,K=5} &= 30.07 \\
diff_{\text{beampattern } L=64,K=0} &= 41.43.
\end{aligned}
$$

So, Capon with $L = 32, K = 0$ and $L = 64, K = 5$ has more similar pre- and post-compression beampatterns than Capon with $L = 64, K = 0$.

In Figure 6.9 we have plotted the dB of the envelope of the whole lateral line created pre- and post-compression of the same *speckle*, or at least corresponding depth, in the tissue. From the Figure we see, maybe more clearly than in the beampatterns, that the pre- and post-compression lateral lines are more similar for the Capon beamformer with $L = 32, K = 0$ (top) and $L = 64, K = 5$ (middle) than Capon with $L = 64, K = 0$ (bottom). When we calculate the sum of squared differences for the three Capon beampatterns we get numbers supporting our visual impression:

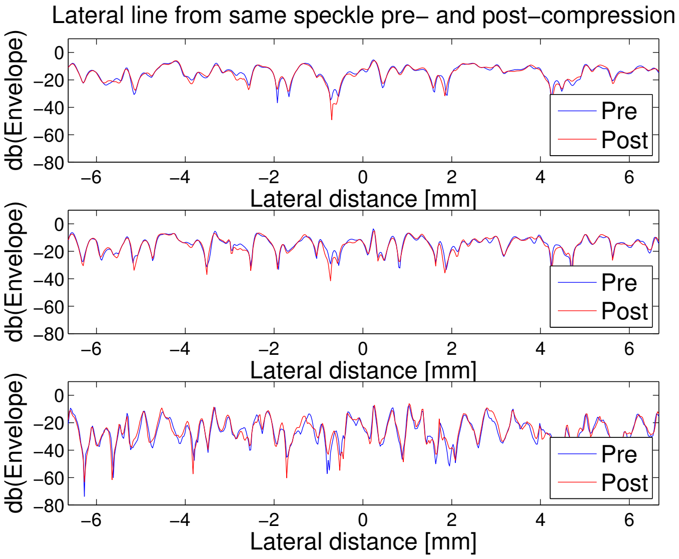


*Figure 6.9: Pre- and post-compression lateral lines for the Capon beamformer.* ***Top:*** *$L = 32, K = 0$,* ***middle*** *$L = 64, K = 5$, and* ***bottom:*** *$L = 64, K = 0$.*

$$\begin{aligned} diff_{\text{lat. line}L=32,K=0} &= 621.90 \\ diff_{\text{lat. line}L=64,K=5} &= 752.23 \\ diff_{\text{lat. line}L=64,K=0} &= 1589.50, \end{aligned}$$

where $L = 64, K = 0$ by far has the largest difference. This result further supports our hypothetical explanation of the results.

Our observation motivates an iterative approach when using the Capon beamformer. One approach could be to first save the weights used to create the beampattern for the pre-compression image. Secondly we estimate the displacement of the tissue as usual, but after this first iteration of finding the displacement we can apply the same weights used to create the beampattern for the pre-compression image when we create the post-compression image for the second time. Then we rerun the displacement estimation where we now use the same Capon beampattern for the same tissue both pre- and post- compression. This will potentially give a better displacement estimation by exploiting the superior resolution of Capon, but with no added adaptation differences between the images.

## 6.4 Summary

In this chapter we have summarized all the simulations executed to test our hypothesis, and have seen that the information created in all these simulations were overwhelming - so to do our final evaluation we extracted the most crucial information. From the evaluation and the discussion we saw that Capon beamforming gave similar performance for axial correlation for displacement estimation as conventional DAS beamforming and thus similar accuracy when doing static elastography. It was in fact the parameters producing similar Rayleigh distributed speckle statistics as DAS, that produced the best estimation results for the Capon beamformer. In the discussion we suggested one possible explanation for this result; that the adaptive behavior of the Capon beamformer slightly changes the image of the same tissue, same speckle, between the pre- and post-compression images and thus making it harder to correlate the images. When we restrict the adaptability of the Capon beamformer by using a smaller subarray, shorter *L*, or add more samples to the estimation of the covariance matrix, larger *K*, we get a more similar beampattern for the pre- and post-compression images of the same speckle - giving us the same estimation performance as the DAS beamformer. This explanation was supported by investigating both the lateral lines and the beampattern created pre- and post-compression.

# Chapter 7

# Conclusion and further work

## 7.1 Conclusion

Adaptive beamforming (Capon) does create a more distinct speckle pattern, smaller speckles, than the conventional (DAS) beamformer. The statistical distribution of the speckle created with the Capon beamformer follows a different distribution than the Rayleigh distributed speckle statistics of the DAS beamformer when we are imaging well developed speckle. However, the speckle statistics of the Capon beamformer do get the same Rayleigh distribution as the DAS beamformer if we use a smaller subarray, $L$, or use more axial samples, larger $K$, when we estimate the spatial covariance matrix used in the Capon beamforming calculations.

Lateral oversampling, denser spacing of the transmit beams, is necessary for single frame scenarios when using Capon beamforming. For single scatterers we needed an oversampling factor of $q \approx 16$, while when imaging well developed speckle a lateral oversampling factor of $q = 4$ was sufficient for our particular setup.

Our main research question was to investigate if the more distinct speckle pattern of the Capon beamformer has benefits when applied to static elastography. To test this hypothesis we thoroughly compared the conventional DAS beamformer to the adaptive Capon beamformer when they were applied to simulated static elastography. We used two different displacement estimations; a crosscorrelation method and a pulsed-Doppler method. Both the error between the estimate and the simulated model of the displacement, and the correlation coefficients resulting from the estimations was used to investigate the hypothesis. Multiple rates of compression and many different parameters for the Capon beamformer and the displacement estimation methods were simulated.

The final conclusion based on the research and results in this thesis is; the adaptive beamformer have similar performance for axial correlation for displacement estimation as conventional beamforming, and thus similar accuracy when doing static elastography. It was in fact the Capon parameters producing similar Rayleigh distributed speckle statistics as DAS, and not the Capon parameters creating the most distinct speckle pattern, that produced the best estimation results for the Capon beamformer. We presented an explanation for our results by arguing that the adaptability of the Capon beamformer resulted in slightly different pre- and post-compression images of the same part of the tissue. Since the displacment estimation is estimating the shift between the pre- and post-compression images, differences between the two images is disadvantageous. Therefore, restricting the adaptability of the Capon beamformer by increasing the number of samples, higher $K$, in the estimation of the spatial covariance matrix, or by using smaller subarrays, lower $L$, in the calculations for the Capon beamformer, gave the best performance for the Capon beamformer.

## 7.2 Future work

This section will suggest some interesting topics for further work, found during the work of this thesis.

### Lateral correlation

We have seen that the axial correlation is similar for the Capon and DAS beamformers. However, it is first and formost in the lateral direction the adaptive beamformer produces better resolution. It is therefore very interesting to investigate if the lateral correlation is improved by the Capon beamformer. If the lateral correlation is improved this possibly also means that block matching speckle tracking is also improved by Capon beamforming.

### Åsen's phase rotation method for linear array imaging

In section 3.2.5 we mentioned the phase rotation method for linear array imaging suggested by Åsen et al. (2014b). However this was used for phased array imaging, where the ultrasound beams are moved by an angle. To use the same approach for linear imaging we probably need to create a new steering angle for every sample at every line. This approach should be investigated to create a simple way of providing lateral oversampling for linear array imaging, which we have shown is necessary when using Capon beamforming for ultrasound imaging.

Åsen's phase rotation method is an approximation to the true beams, so the effect of this method needs to be investigated and compared to actually creating the beams. When his approach is used for linear array imaging we probably also use larger steering angles for the closer parts of the image, this might give worse approximations than what Åsen found for his very small angular shifts.

It is also interesting to see if the lateral oversampling factor of $q = 4$ for speckle imaging is sufficient with this method.

### Larger amounts of displacement

Simulating larger amounts of displacement could be interesting. We have only simulated two frames with a relatively small compression between the images. The effect of using larger displacement, and for example find where the crosscorrelation displacement estimation method breaks down.

### Displacement estimation with the cross-ambiguity function

Sæbø et al. (2007) showed that the cross-ambiguity function gave a significant improvement compared to the conventional crosscorrelation function when doing sidelooking sonar height estimation. This is a similar problem to the displacement estimation done in this thesis. It would be interesting to investigate if the cross-ambiguity function gives similar improvements for displacement estimation, and investigate if the performance of the two beamformers are still similar.

The cross-ambiguity function should also allow larger amounts of displacement, and since it finds the actual stretch it introduces a new approach to find the strain.

### Iterative displacement estimation with the Capon beamformer

In Section 6.3 we suggested to try an iterative displacement estimation when using the Capon beamformer. If we use an iterative approach we can use the same beampattern for the pre- and post-compression images and possibly exploit the superior resolution of the Capon beamformer, but with no added adaptation difference between the images.

Alternatively we could save the data used for calculating the spatial covariance matrix for the pre-compression image, and use this data combined with the data for the post-compression image to use the information from both images when creating the covariance matrix and thus the weights used for the second iteration and the final pre- and post-compression images.

# Appendix A

# Code

In this appendix we will display the most important MATLAB code used in this thesis. For this thesis is has been written 6844 lines of code, most of which is for the simulation of ultrasound images in Field II. As much as possible of the code has been parallelized with the parallelized for-loops *parfor* in MATLAB. The simulation of one line in one ultrasound image took about 20 minutes when running on the *oskikkanlegur* server at the University of Oslo. Meaning that the images with 512 lines took about $\frac{512 \text{ lines}}{12 \text{ workers}} \times \frac{20 \text{ minutes}}{60 \text{ minutes/ hour}} \approx 14$ hours, and thus 28 hours for both the pre- and post-compression images, since MATLAB generously (ironic) gives us only 12 parallelization workers, while *oskikkanlegur* actually had 48 cores. The largest images created, with oversampling factor $q = 16$, had 2048 lines and thus took about 56 hours to complete. Finding a bug after the simulation had completed was therefore very frustrating.

## A.1 Implementation of the Capon beamformer

The implementation of the Capon beamformer is written with emphasis on intuition - not speed. The code can be written much more efficient. When we call this function we call it from a *parfor* loop in MATLAB, this allows us to calculate each line in parallel giving quite a speedup.

```
function [z,w] = myCaponCubeByLine(rf_data,regCoef,...
                                   L,K,doForwardBackward)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Beamform one axial line from M elements using
%% the Capon beamformer
%%
%% This implementation is written with focus of
%% intuition, not speed.
%%
%%@Input:
%%     rf_data  : RF-data from the elements
%%     regCoef  : Diagonal loading constant
%%     L        : Subarray size
%%     K        : Temporal averaging factor
%%     doForwardBackward : Forward backward averaging
%%              (This is never used in this thesis)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[N M] = size(rf_data);
z = zeros(1,N); %Beamformed output
w = zeros(L,N); %Weights calculated
I = eye(L);     %Identity matrix

%For every sample
for k = 1:N
    if sum(rf_data(k,:) == 0) %Ignore data with zeros
        z(k) = 0;
    else
        %Estimate spatial covariance matrix
        R_sub = zeros(L,L);
        for l = 1:M-L+1
            if k-K < 1
                R_sub=...
                    R_sub+rf_data(1:k+K,l:l+L-1)'...
                              *rf_data(1:k+K,l:l+L-1);
            elseif k+K > N
                R_sub=...
                   R_sub+rf_data(k-K:end,l:l+L-1)'...
                            *rf_data(k-K:end,l:l+L-1);
            else
                R_sub...
                   =R_sub+rf_data(k-K:k+K,l:l+L-1)'...
```

```
                                   *rf_data(k-K:k+K,l:l+L-1);
            end
        end
        %Average the subarrays
        R_sub = R_sub/(M-L+1);

        %Forward backward averaging
        if(doForwardBackward)
            R_sub = 0.5*(R_sub+rot90(conj(R_sub),2));
        end

        %Invert matrix with added diagonal loading
        Ri = inv(R_sub + I*(regCoef/L)*trace(R_sub));

        %Create weights
        a = ones(L,1); %Steering vector
        w_mv = (Ri*a)/(a'*Ri*a);

        %Multiply weights with sub parts of data
        %We are using amplitude Capon
        z(k) = 0;
        for j = 1:M-L+1
            z(k) = z(k) + w_mv'*rf_data(k,j:L+j-1)';
        end
        %Average output
        z(k) = z(k)/(M-L+1);
        %Return weights
        w(:,k) = w_mv;
    end
end; end
```

## A.2 Crosscorrelation displacement estimation

The central parts of the implementation of the crosscorrelation displacement estimation.

```
%Parallelize each line
parfor line = 1:M
    disp(['Line␣',num2str(line)]);
    for i = 1:N-window_length
        %Divinding RF-data into windows
        pre_window = pre_rf(i:i+window_length,line);
        post_window = post_rf(i:i+window_length,line);

        %Crosscorrelate the windows
        [c,lags] = xcorr(pre_window,post_window,...
                                        max_lags,'coeff');
        %Interpolate the result
        lags_interp=interp(lags,interpolating_factor);
        c_interp = interp(c,interpolating_factor);

        %Find the maximum peak of the real part
        [C_real(i,line),I_real] = max(real(c_interp));

        %Calculate the displacement
        displacement_real(i,line) =...
                    p.c*lags_interp(I_real)/(p.fs*2);
```

## A.3 Pulsed-Doppler displacement estimation

The implementation used for the pulsed-Doppler displacement estimation is based on the implementation presented by Børstad (2011).

```
function [d_2,d_1,f_hat,fc_hat,C] = ...
 pulsedDopplerDisplacementEstimation(X,P,PARAMETERS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    pulsedDopplerDisplacementEstimation
%
%    This function calculates the pulsed-doppler
%    displacement estimation. It calulcates both
%    the atocorrelation and the modified
%    autocorrelation method. The implementation is
%    highly influenced by Borstads implementation
%    from his master's thesis at NTNU.
%
%@input
%    X : RF-data from multiple frames. Here two frames
%    P : Different constants
%    PARAMETERES : Parameters for this estimation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c   = P.c;                        %Wave velocity
fs  = P.fs;                       %Sampling frequency
fc = P.f0;                        %Central frequency
U = PARAMETERS.rangeGate;         %Axial gate  : 15
V = PARAMETERS.lateralGate;       %Lateral gate: 4
O = PARAMETERS.numberOfFrames;%Number of frames used

X_conj = conj(X); %Complex conjugate of X

%Autocorrelation between frames lag
R_0_1 = sum(X(1:end-1,:,1:end-1).*...
                X_conj(1:end-1,:,2:end),3);
%Estimation window filtering
```

```
R_0_1 = conv2(R_0_1, ones(U,1), 'valid');
R_0_1 = conv2(R_0_1, ones(1,V), 'valid');

%Autocorrelation time lag
R_1_0 = sum(X(1:end-1,:,:).*X_conj(2:end,:,:),3);
%Estimation window filtering
R_1_0 = conv2(R_1_0, ones(U,1), 'valid');
R_1_0 = conv2(R_1_0, ones(1,V), 'valid');

%Estimated doppler frequency
f_hat = abs(unwrap(angle(R_0_1))/(2*pi));
%Estimated centeral frequency
fc_hat = abs(unwrap(angle(R_1_0))/(2*pi*1/fs));

%Correlation coefficient estimation qualityindicator.
C = sum(X(1:end-1,:,:).*X_conj(1:end-1,:,:),3);
C = conv2(C, ones(U,1), 'valid');
C = conv2(C, ones(1,V), 'valid');
C = (O/(O-1))*abs(R_0_1)./C;

%Autocorrelation method
d_1 = abs(c*f_hat/(2*fc));
%Modified autocorrelation method
d_2 = abs(c*f_hat./(2*fc_hat));
end
```

# Appendix B

# The Hilbert Transform

The Hilbert transform is used excessively through this thesis, and therefore deserves a closer description. The Hilbert transform is an operation that shifts the phase of a signal $x(t)$ by $-\frac{\pi}{2}$. In time domain the phase shifted signal $\hat{x}(t)$ is given by the convolution

$$\hat{x}(t) = \frac{1}{\pi t} * x(t). \tag{B.1}$$

In frequency domain the Hilbert transform acts as a filter with transfer function

$$H(f) = \begin{cases} -i = e^{+\frac{i\pi}{2}} & ,f > 0 \\ 0 & ,f = 0 \\ i = e^{-\frac{i\pi}{2}} & ,f < 0 \end{cases} \tag{B.2}$$

And this is where the interesting stuff happens. This means that the negative frequencies will be shifted by a phase of $-\frac{\pi}{2}$ and the positive frequencies with $+\frac{\pi}{2}$.

In this thesis we are using the Hilbert transform for two reasons. To create the analytic signal, and to find the envelope of a signal.

## B.1 Analytic signal

The analytic signal

$$x_a[n] = x[n] + j(\hat{x}[n]) \mid \hat{x}[n]\text{: Hilbert transform of } x[n]. \tag{B.3}$$

is the one-sided representation of a real-valued signal. The idea is that the negative frequency components in the Fourier spectrum of a real-valued signal is redundant and can be discarded without any loss of information. This gives us a complex signal, which in many cases is beneficial as it facilitates many signal processing techniques. In this thesis it allows us to have complex weights for the Capon beamformer 2.3.2 allowing a non-symmetric beampattern, and allows the pulsed-Doppler techniques. As long as the signal processing techniques does not introduce negative frequencies we can convert back to a real signal by just discarding the imaginary part.

In Figure B.1 we demonstrate what we just discussed.

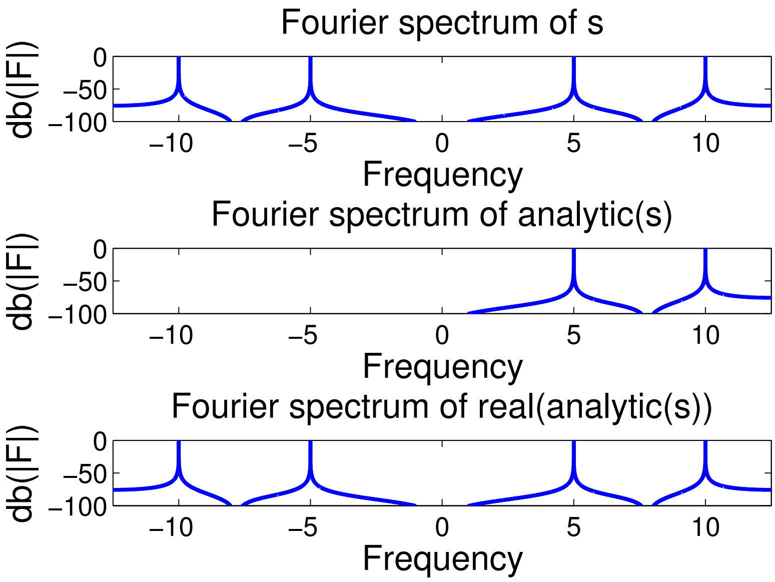


*Figure B.1: The top is the Fourier spectrum of a signal consisting of two sinuses with frequency 5 and 10. The signal is real so the first spectrum is two-sided. The middle plot is the spectrum of the analytic signal as defined above, and the bottom plot is the Fourier spectrum of just the real part of the analytic signal. We see that as expected no information is lost in the signal.*

## B.2 Envelope of signal

The envelope of the signal can easily be found by taking the magnitude of the analytic signal $A[n] = |x_a[n]| = \sqrt{x[n]^2 + \hat{x}[n]^2}$ this follows from the fact that the Hilbert transform is a $\frac{\pi}{2}$ phase shift of the signal, see Figure B.2 where *signal* is $x[n]$ and *Hilbert(s)* is $\hat{x}[n]$, so the square root of the sum of squares of the two signals will intuitively be the envelope.

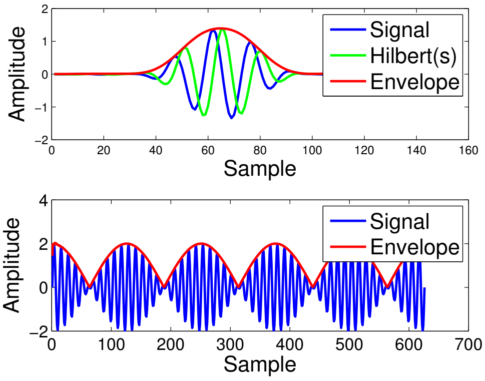


*Figure B.2: The top is displaying the signal, the Hilbert transform of the signal and the envelope of the signal. The bottom shows the signal and the envelope of the signal.*

# Appendix C

# Crosscorrelation time delay estimator

To derive the cross correlation time delay estimator we can follow the arguments in the lecture notes, lecture #14, in the course *EECE 522 Estimation Theory* at Binghamton University.

The time delay problem can be described by a transmitted signal $s(t)$ and the received signal $s(t-\tau_0)$ where $\tau_0$ is the time delay we want to find. This received signal is often buried in noise, so we receive $x(t) = s(t-\tau_0) + w(t)$ where we will assume that $w(t)$ is Gaussian noise.

The sampled version of the signal can be expressed as

$$x[n] = \begin{cases} w[n] & 0 \leq n \leq n_0 - 1 \\ s[n-n_0] + w[n] & n_o \leq n \leq n_0 + M - 1 \\ w[n] & n_o + M \leq n \leq N - 1 \end{cases}$$

To estimate the $\tau_0$, or its sampled version $n_0$, we can do a Maximum likelihood estimation (MLE). To do this we need the probability density function (PDF) of the signal. Since all the three subintervals has Gaussian distribution, and because they are independent, we can write the PDF as a product of three different PDFs. One for each subinterval:

$$p(x;n_o) =$$

$$\prod_{n=0}^{n_0-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2[n]}{2\sigma^2}} \cdot \prod_{n=n_o}^{n_0+M-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x[n]-s[n-n_0])^2}{2\sigma^2}} \cdot \prod_{n=n_0+M}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2[n]}{2\sigma^2}}.$$

Now, since $(x[n]-s[n-n_0])^2 = x^2[n] - 2x[n]s[n-n_0] + s^2[n-n_0]$ - the last part cancel. The products can also be moved inside the exponent, since that is the only part dependent on $n$, leaving us with

$$p(x;n_o) =$$

$$= (\frac{1}{\sqrt{2\pi\sigma^2}})^N e^{\left(-\frac{\sum_{n_0}^{N-1} x^2[n]}{2\sigma^2}\right)} \cdot e^{\left(-\frac{1}{2\pi\sigma^2}\sum_{n=n_0}^{n_0+M-1}(-2x[n]s[n-n_0]+s^2[n-n_0])\right)}.$$

Our goal was to estimate $N_0$, so since the first part does not depend on $n_0$ we only need to maximize the negative of

$$\begin{aligned}
&2\sum_{n=n_0}^{n_0+M-1}(x[n]s[n-n_0]+s^2[n-n_0]) \\
&=2\sum_{n=n_0}^{n_0+M-1}(x[n]s[n-n_0])+\sum_{n=n_0}^{n_0+M-1}s^2[n-n_0].
\end{aligned}$$

The second part does not depend on $n_0$ since the summand moves with the limits as $n_0$ changes. So we only need to maximize

$$\sum_{n=0}^{N-1}x[n]s[n-n_0].$$

This is a familiar expression - the crosscorrelation. This means that the MLE can be implemented as finding the maximum of a crosscorrelation between the received signal $x[n]$ and the transmitted signal $s[n]$.

# Appendix D

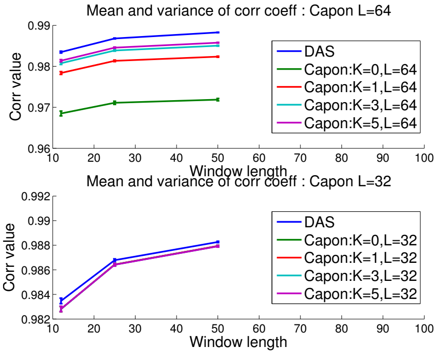
# Plots from all simulations

## D.1 Simulations with dynamic transmit focus

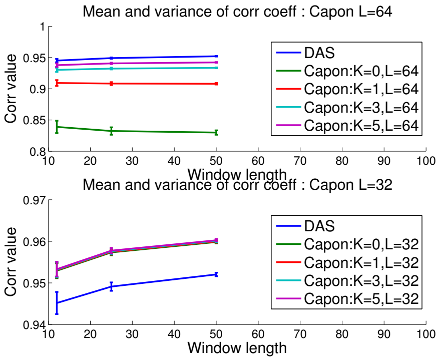
### D.1.1 Compression = $\frac{\Delta}{2}$

#### Cross correlation estimation

#### Correlation values

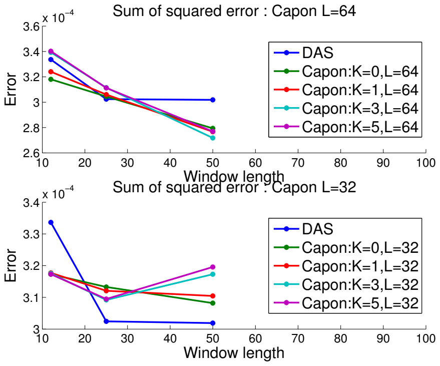


(a) Without noise

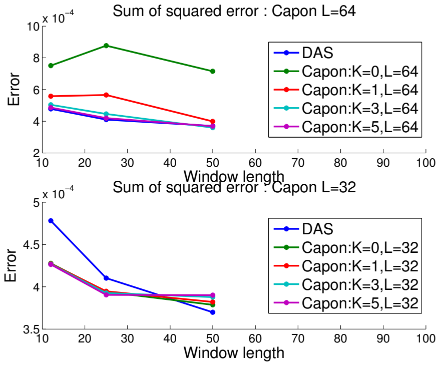


(b) Noise

#### Error values



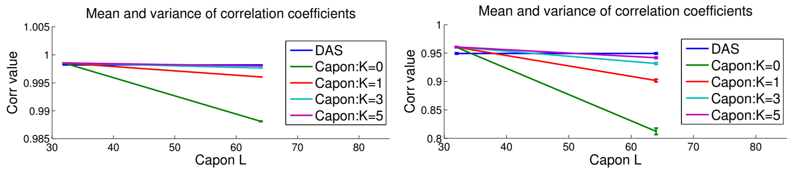
(c) Without noise



(d) Noise

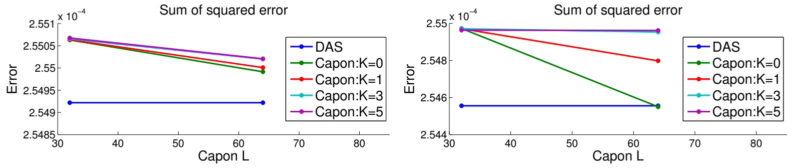
### Pulsed-Doppler estimation

### Correlation values



(e) Without noise (f) Noise

### Error values

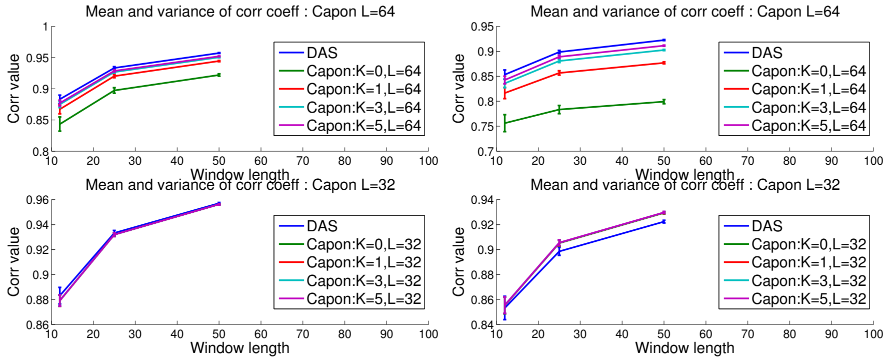


(g) Without noise (h) Noise

## D.1.2 Compression = Δ

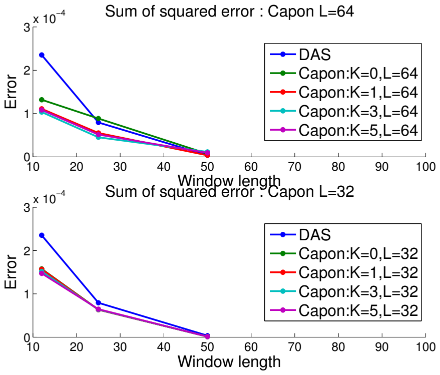
### Cross correlation estimation

### Correlation values

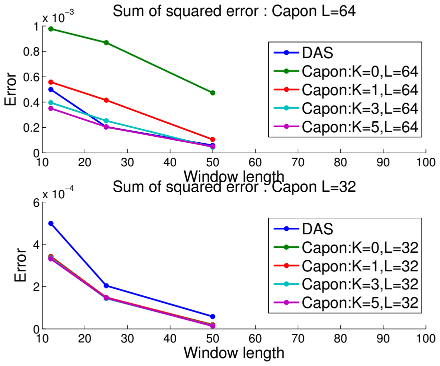


(i) Without noise (j) Noise

## Error values



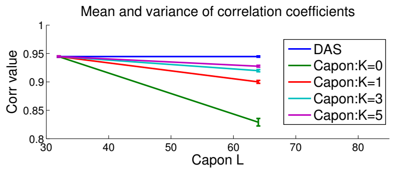
(k) Without noise



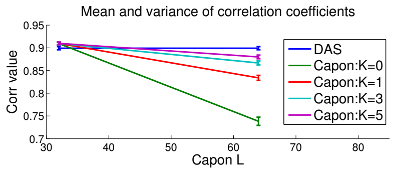
(l) Noise

## Pulsed-Doppler estimation

## Correlation values

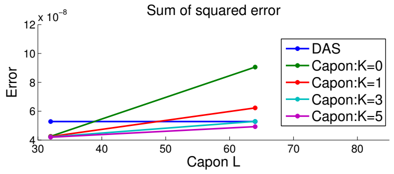


(m) Without noise

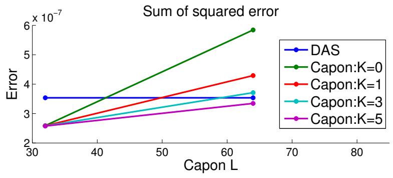


(n) Noise

## Error values



(o) Without noise

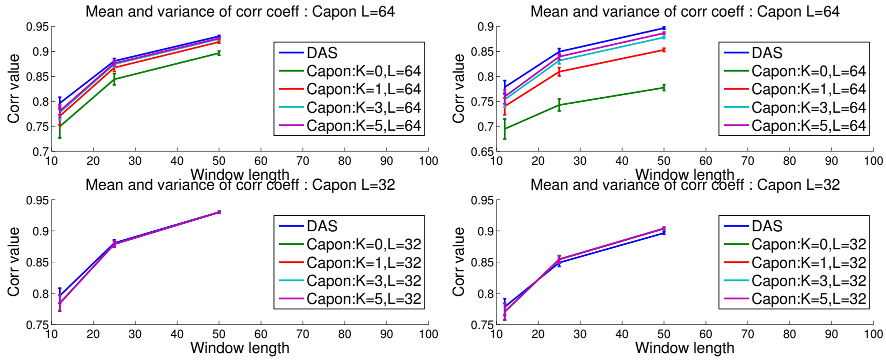


(p) Noise

### D.1.3 Compression = 2Δ

## Cross correlation estimation

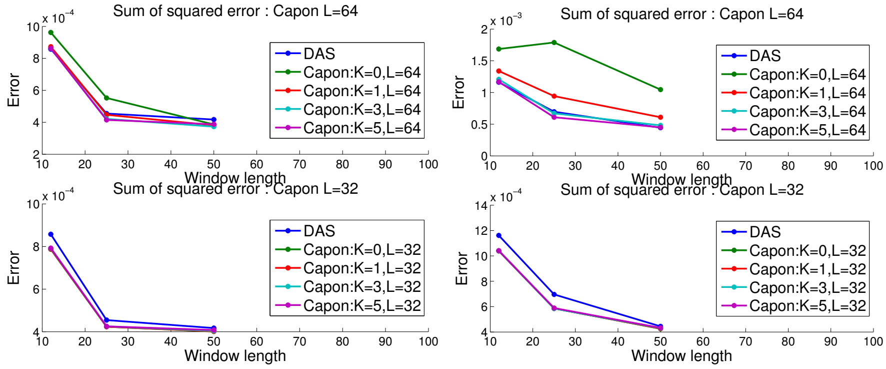
## Correlation values



(q) Without noise

(r) Noise

## Error values



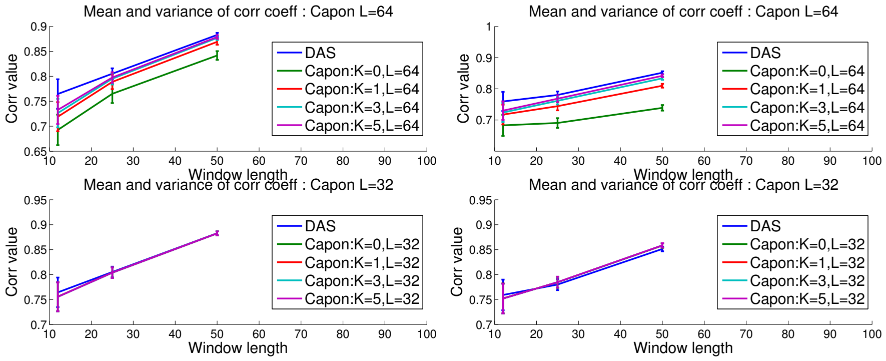
(s) Without noise

(t) Noise

## D.1.4 Compression = 4Δ

### Cross correlation estimation

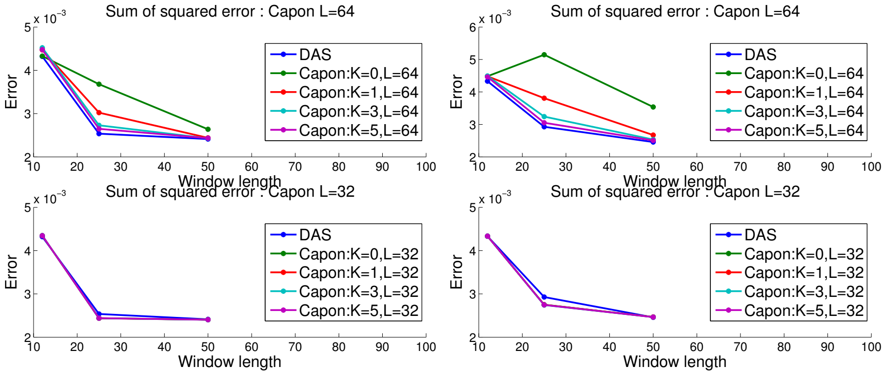
### Correlation values



(u) Without noise

(v) Noise

### Error values



(w) Without noise

(x) Noise

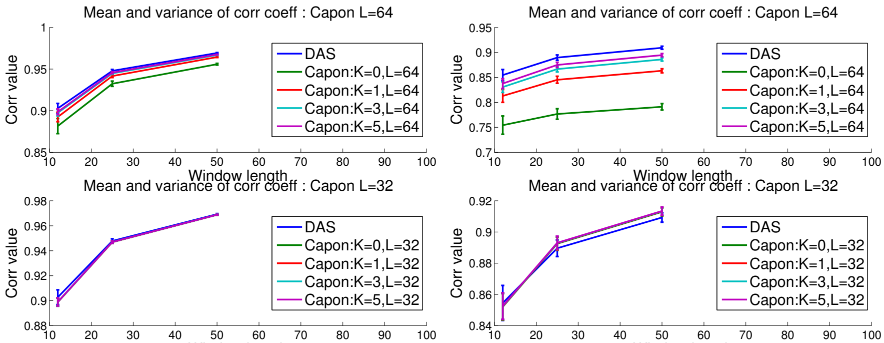
*Figure D.1*

# D.2 Simulations with fixed focus transmit

## D.2.1 Compression = $\frac{\Delta}{2}$

### Cross correlation estimation

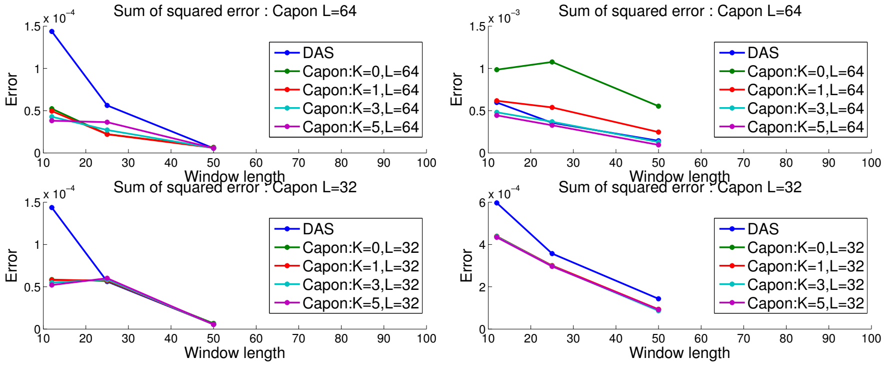
### Correlation values



(a) Without noise

(b) Noise

### Error values

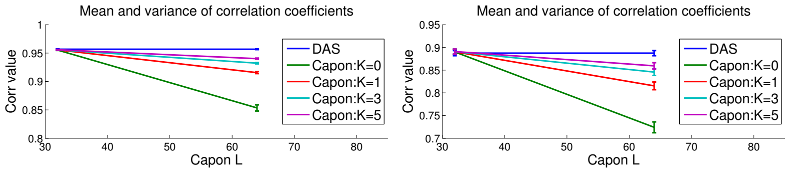


(c) Without noise

(d) Noise

## Pulsed-Doppler estimation

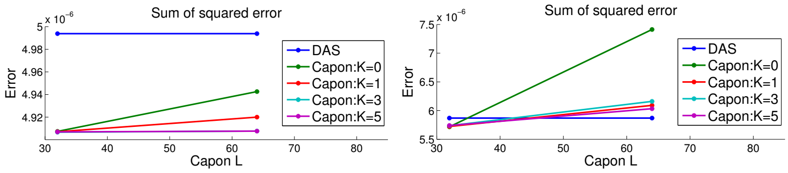
## Correlation values



(e) Without noise

(f) Noise

## Error values



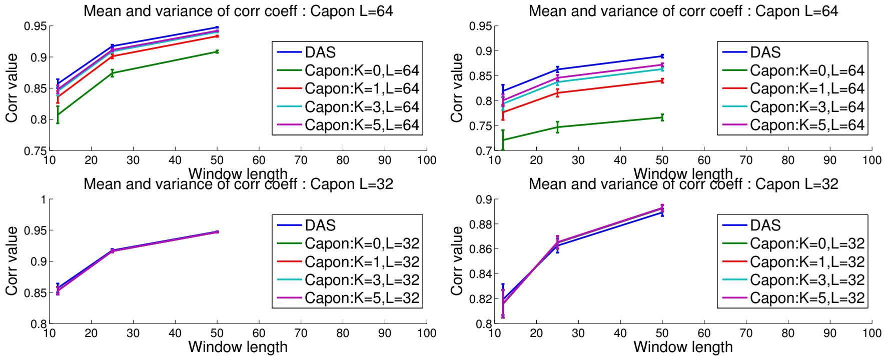
(g) Without noise

(h) Noise

## D.2.2 Compression = Δ

## Cross correlation estimation

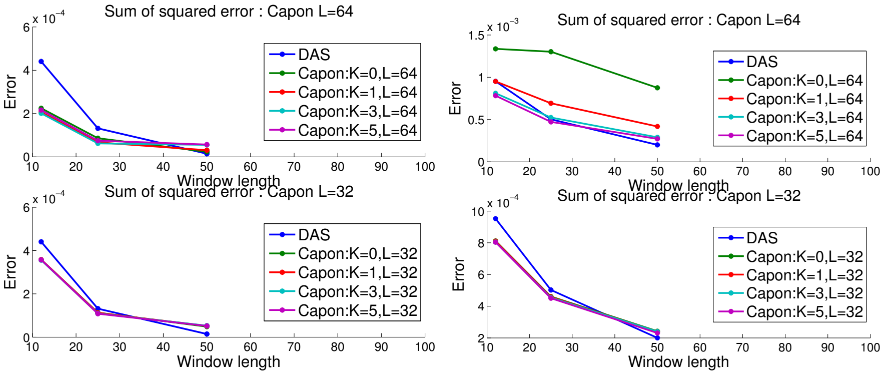
## Correlation values



(i) Without noise

(j) Noise

## Error values

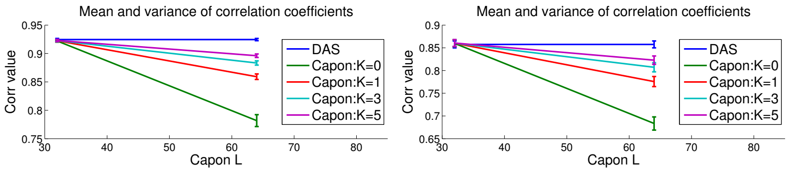


(k) Without noise

(l) Noise

## Pulsed-Doppler estimation

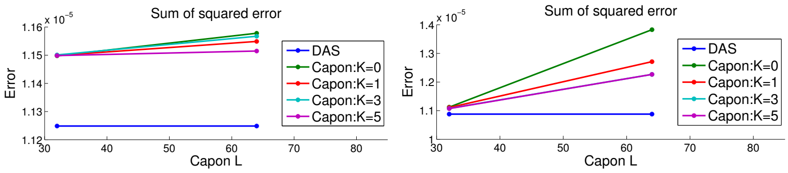
## Correlation values



(m) Without noise

(n) Noise

## Error values



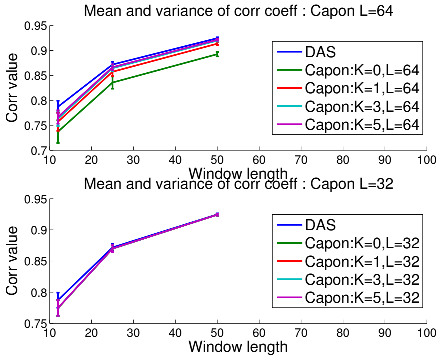
(o) Without noise

(p) Noise

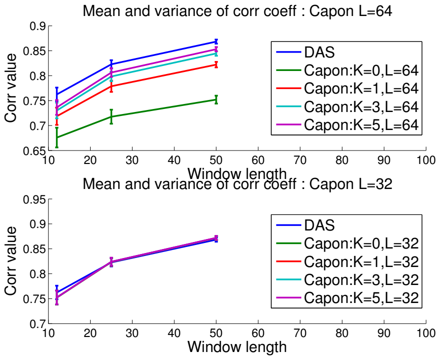
### D.2.3 Compression = 2Δ

## Cross correlation estimation

## Correlation values

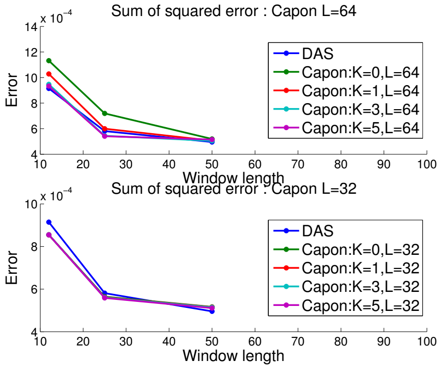


(q) Without noise

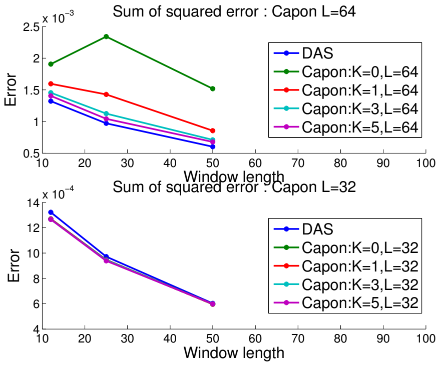


(r) Noise

## Error values



(s) Without noise

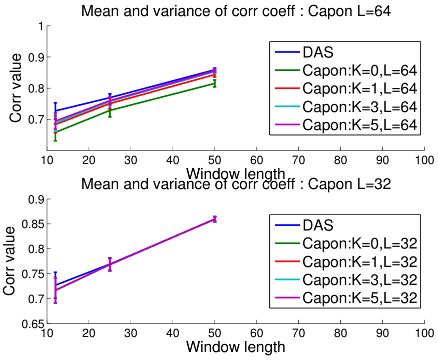


(t) Noise

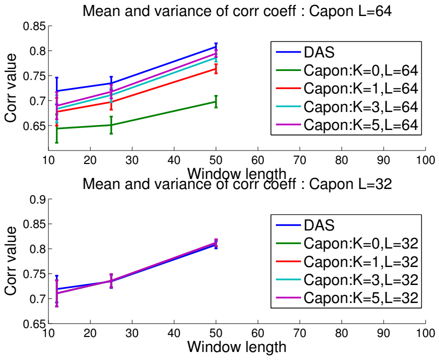
### D.2.4 Compression = 4Δ

## Cross correlation estimation

## Correlation values

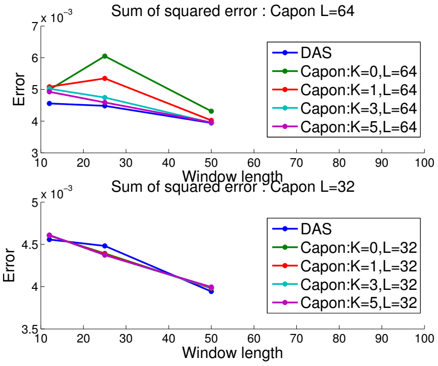


(u) Without noise

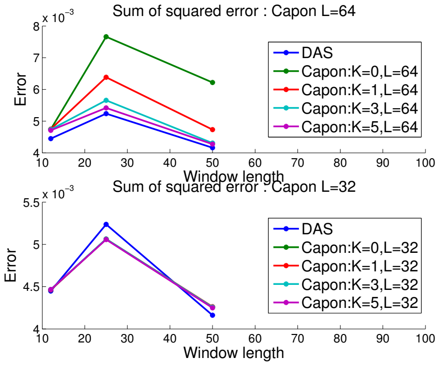


(v) Noise

## Error values



(w) Without noise



(x) Noise

# Appendix E

# Abstracts submitted to the 2014 IEEE International Ultrasonics Symposium

## Understanding Contrast Improvements from Capon Beamforming

Ole Marius Hoel Rindal[1], Jon Petter Åsen[2], Andreas Austeng[1] and Sverre Holm[1]

### Background, Motivation and Objective

It is common to state that Capon beamforming applied to ultrasound images increases both contrast and resolution. However, the increased contrast is mainly a result of the increased resolution giving sharper edges in e.g. a cyst. Recently it has also been shown that lateral oversampling is needed to achieve lateral shift-invariance between image frames when using Capon beamforming (Åsen et al IEEE UFFC 2014). This is also essential for single frame scenarios, and especially when considering contrast. This work aims at understanding the contrast improvements obtained with Capon beamforming, and how to image point scatterers and speckle without loss of information.

### Statement of Contribution/Method

We have simulated a 196 element linear probe with center frequency at 7.5 MHz in Field II. Speckle statistics from conventional and Capon beamforming are investigated with emphasis on the lateral oversampling factor needed to avoid loss in information when imaging speckle with Capon beamforming. Cylindrical cysts with different radii were included in the simulations and the contrast was investigated by calculating the CNR

[1] University of Oslo
[2] Squarehead Technology AS

(contrast-to-noise ratio) between regions of different sizes inside the cyst and a region of speckle.

We have also simulated brighter inclusions together with dark cysts and discussed how this influence the dynamic range and how the different beamformers influence the visibility of the inclusions. We demonstrate that Capon beamforming gives sharper edges and examine the beampatterns for the different beamformers at the edge of the cyst to explain why Capon beamforming produces sharper edges

### Results, Discussion and Conclusions

We show that lateral oversampling is needed when using the Capon beamformer, but the factor of oversampling is smaller when imaging speckle than point scatterers. If the Nyquist requirement for beam spacing is used for Capon beamforming a point scatterer placed between two beams has a 40 dB lower amplitude than a point scatterer directly on the beam. We demonstrate that Capon beamforming does not increase the contrast in general. The same contrast as Capon is achieved with DAS with Hamming apodization, but Capon improves the edges and thus gives better contrast for smaller cysts. Compared to non-weighted DAS, a cyst with 2.5 mm radius has 33 % better CNR for the Capon beamformer with $K = 5$ temporal averaging and $L = 32$ subarray averaging, when the CNR calculation is based on a circular region with radius of 1.25 mm in the center of the cyst. The CNR for DAS with Hamming apodization is actually 1 % better for the same region compared to the Capon beamformer. When the radius of the area from which the CNR is calculated is increased and approaches the radius of the cyst, the Capon beamformer gives higher CNR compared to the DAS beamformers i.e. Capon beamforming provides contrast improvements near edges because of the improved lateral resolution.

# Comparing Conventional and Adaptive Beamforming for Static Elastography

Ole Marius Hoel Rindal[1], Andreas Austeng[1] and Sverre Holm[1]

## Background, Motivation and Objective

Static elastography consists of creating a pre- and post-compression ultrasound image where the tissue being imaged has been compressed between the images. The displacement of tissue is calculated along the axial dimension based on the assumption that speckle pattern follows tissue movement. Tissue strain, indicating the stiffness of tissue, can then be found from the displacement of the tissue. Speckle statistics and the speckle pattern are different for images created with conventional and adaptive (Capon) beamforming. The speckle pattern created with adaptive beamforming has a smaller and more distinct pattern because of the improved resolution by adaptive beamforming. Hypothetically a more distinct pattern should result in better correlation and thus better displacement estimation.

Recently it has been shown that lateral oversampling is needed to achieve lateral shift-invariance between image frames when using adaptive beamforming (Åsen et al IEEE UFFC 2014). Shift-invariance between frames is especially important for elastography since the displacement estimate is based on correlation between two nearly identical frames. We analyze the lateral oversampling needed when imaging point scatterers and well developed speckle before we investigate if the difference in speckle statistics and pattern of adaptive beamforming has benefits when doing static elastography.

## Statement of Contribution/Method

Two speckle images are created from Field II simulations based on the same scatter phantom where the scatterers have been displaced axially to create pre- and post-compression ultrasound images. The images are created with the conventional beamformer and the adaptive beamformer with different parameters. In the middle of the phantom a circular object has constant displacement to mimic a hard malignant nodule in the tissue.

Two methods to estimate the displacement of tissue are implemented and investigated, the crosscorrelation method and a pulsed- doppler method. Multiple parameters for the two methods are tested on different rates of displacement both with and without added noise.

## Results, Discussion and Conclusions

We show that lateral oversampling is necessary for single frame scenarios when doing adaptive beamforming and to achieve shift- invariant imaging

[1] University of Oslo

of speckle. We have measured and compared the correlation values and the sum of squared error between the displacement estimate and the displacement model. The speckle pattern from adaptive beamforming is more distinct and does have a wider frequency spectrum, but our research shows that this gave similar performance for axial correlation for displacement estimation as conventional beamforming and thus similar accuracy when doing static elastography.

# References

Bjørn A. J. Angelsen and Kjell Kristoffersen. Discrete time estimation of the mean Doppler frequency in ultrasonic blood velocity measurements. *IEEE Transaction on Biomedical Engineering*, (4), 1983.

Jon Petter Åsen, Andreas Austeng, and Sverre Holm. Capon Beamforming and Moving Objects - An Analysis of Lateral Shift-Invariance. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, to appear in, 2014a.

Jon Petter Åsen, Jo Inge Buskenes, Carl-Inge Colombo Nilsen, Andreas Austeng, and Sverre Holm. Implementing Capon Beamforming on a GPU for Real-Time Cardiac Ultrasound Imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Controll*, 61(1), 2014b.

Stephane Audiere, Yassine Mofid, Maurice Charbit, Elsa Angelini, Veronique Miette, and Laurent Sandrin. Fibroscan® practice improvement with a real-time assistance ultrasound tool: a premiminary study. In *2009 IEEE International Ultrasonics Symposium*, 2009.

Andreas Austeng, Carl-Inge Colombo Nilsen, Are Charles Jensen, Sven Peter Nasholm, and Sverre Holm. Coherent plane-wave compounding and minimum variance beamforming. In *2011 IEEE International Ultrasonics Symposium*, 2011.

Donald W. Baker. Pulsed ultrasonic Doppler blood-flow sensing. *IEEE Transactions on Sonics and Ultrasonics*, 17(3), 1970.

Richard Barakat. Application of Apodization to Increase Two-Point Resolution by the Sparrow Criterion. I. Coherent Illumination. *Journal of the Optical Society of America*, 52(3), 1962.

William D. Barber, Jeffrey W. Eberhard, and Steven G. Karr. A new time domain technique for velocity measurements using Doppler ultrasound. *IEEE Transaction on Biomedical Engineering*, 32(3), 1985.

Jérémy Bercoff. ShearWave ™ Elastography. Technical report, SuperSonic Imagine, 2008.

Jérémy Bercoff, Mickaël Tanter, and Mathias Fink. Supersonic shear imaging: a new technique for soft tissue elasticity mapping. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 51(4), 2004.

Ann Elisabeth Albright Blomberg. *Evaluation of a modified autocorrelation method when applied to echocardiographic strain rate imaging*. Master's thesis, University of Oslo, 2005.

Thomas Kristoffersen Børstad. *Intraoperative Ultrasound Strain Imaging of Brain Tumors*. Master's thesis, Norwegian University of Science and Technology, 2011.

Finn Bryn. Optimum Signal Processing of Three-Dimensional Arrays Operating on Gausssian Signals and Noise. *The Journal of the Acoustical Society of America*, 34(3), 1962.

Cristoph B. Burckhardt. Speckle in Ultrasound B-Mode Scans. *IEEE Transactions on Sonics and Ultrasonics*, 25(1), 1978.

J. Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8), 1969.

Henry Cox. Resolving power and sensitivity to mismatch of optimum array processors. *The Journal of the Acoustical Society of America*, 54(3), 1973.

Matthew Wright Gilbertson. *Handheld Force-Controlled Ultrasound Probe*. Master's thesis, Massachusetts Institute of Technology, 2010.

Fredric J. Harris. On the Use of Windows for Harmonic Analysis. *Proceedings of the IEEE*, 6(1):51–83, 1978.

Torbjørn Hergum, Tore Bjåstad, Kjell Kristoffersen, and Hans Torp. Parallel beamforming using synthetic transmit beams. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 54(2), 2007.

Jørgen Arendt Jensen. Field: A program for simulating ultrasound systems. *Medical & Biological Engineering & Computing*, 34, 1996a.

Jørgen Arendt Jensen. *Estimation of blood velocities using ultrasound: a signal processing approach*. Cambridge University Press, Cambridge, 1996b.

Jørgen Arendt Jensen and Niels Bruun Svendsen. Calculation of Pressure Fields from Arbitrarily Shaped, Apodized, and Excited Ultrasound Transducers. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 39(2), 1992.

Don H. Johnson and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Prentice-Hall signal processing series. P.T.R. Prentice-Hall, 1993.

Chihiro Kasai, Koroku Namekawa, Akira Koyano, and Ryozo Omoto. Real-Time Two-Dimensional Blood Flow Imaging Using an Autocorrelation Technique. *IEEE Transactions on Sonics and Ultrasonics*, 32(3), 1985.

Hamid Krim and Mats Viberg. Two Decades of Array Signal Processing Research: The Parametric Approach. *IEEE Signal Processing Magazine*, 13 (4), 1996.

Jian Li, Petre Stoica, and Zhisong Wang. On Robust Capon Beamforming and Diagonal Loading. *IEEE Transactions on Signal Processing*, 51(7), 2003.

Thanasis Loupas, J.T. Powers, and Robert W. Gill. An Axial Velocity Estimator for Ultrasound Blood Flow Imaging, Based on a Full Evaluation of the Doppler Equation by Means of a Two-Dimensional Autocorrelation Approach. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 42(4), 1995.

J. Ophir, I Cespedes, H Ponnekanti, Y Yazdi, and X Li. Elastography: a quantitative method for imaging the elasticity of biological tissues. *Ultrasonic imaging*, 134(13), 1991.

J Ophir, SK Alam, and BS Garra. Elastography: imaging the elastic properties of soft tissues with ultrasound. *Journal of Medical Ultrasonics*, 29, 2002.

T. Rago, F. Santini, M. Scutari, A. Pinchera, and P. Vitti. Elastography: New Developments in Ultrasound for Predicting Malignancy in Thyroid Nodules. *The Journal of Clinical Endocrinology & Metabolism*, 92(8), 2007.

T.O. Sæbø, R.E. Hansen, H.J. Callow, and B. Kjellesvig. Using the Cross-Ambiguity Function for Improving Sidelooking Sonar Height Estimation. In *2nd International Conference & Exhibition on "Underwater Acoustic Measurements: Technologies & Results"*, 2007.

Shigeo Satomura. Ultrasonic Doppler method for the inspection of cardiac functions. *The Journal of the Acoustical Society of America*, 29(11), 1957.

Rémi Souchon, Olivier Rouvière, Albert Gelet, Valérie Detti, Seshadri Srinivasan, Jonathan Ophir, and Jean-Yves Chapelon. Visualisation of HIFU lesions using elastography of the human prostate in vivo: preliminary results. *Ultrasound in Medicine & Biology*, 29(7), 2003.

Bernhard D. Steinberg. *Principles of Aperture and Array System Design*. John Wiley & Sons, 1976.

Johan-Fredrik Synnevåg, Andreas Austeng, and Sverre Holm. Adaptive Beamforming Applied to Medical Ultrasound Imaging. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 54(8), 2007a.

Johan-Fredrik Synnevåg, Carl-Inge Colombo Nilsen, and Sverre Holm. Speckle Statistics in Adaptive Beamforming. In *IEEE Ultrasonics Symposium*, 2007b.

Johan-Fredrik Synnevåg, Andreas Austeng, and Sverre Holm. Benefits of Minimum-Variance Beamforming in Medical Ultrasound Imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 56(9), 2009.

Shan Tie-Jun, Mati Wax, and Thomas Kailath. On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(4), 1985.

Fabrizio Vecchi, Cinzia Freschi, Silvestro Micera, Angelo M. Sabatini, Paolo Dario, and Rindaldo Sacchetti. Experimental Evaluation of Two Commercial Force Sensors for Applications in Biomechanics and Motor Control. *5th Annual Conference of the International Functional Electrical Stimulation Society*, 2000.

Robert F. Wagner, Stephen W. Smith, John M. Sandrik, and Hector Lopez. Statistics of Speckle in Ultrasound B-Scans. *IEEE Transactions on Sonics and Ultrasonics*, 30(3), 1983.

Robert F. Wagner, Micahel F. Insana, and Stephen W. Smith. Fundamental Correlation Lengths of Coherent Speckle in Medical Ultrasonic Images. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 35(1), 1988.

T. A. Whittingham. Medical diagnostic applications and sources. *Progress in Biophysics and Molecular Biology*, 93, 2007.

J. Nelson Wright. Image Formation in Diagnostic Ultrasound. In *IEEE International Ultrasonics Symposium*, 1997.