NTNU – Trondheim
Norwegian University of
Science and Technology

Model based estimation of speckle filter for improved suppression of multiple scattering noise with dual band ultrasound pulse complexes

Stian Solberg

Master of Science in Electronics	
Submission date:	June 2015
Supervisor:	Ilangko Balasingham, IET
Co-supervisor:	Bjørn Angelsen, ISB

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

Dual band ultrasound pulse complexes (SURF pulse complexes) composed of a overlapping low frequency (LF, e.g. 0.5 MHz) pulse and a high frequency (HF, e.g. 10 MHz) pulse is useful for improved suppression of multiple scattering noise. The received multiple scattered HF signal from such a pulse complex will vary with variations in the LF pulse, described by a variation in signal delay and a variation in the speckle of the multiple scattered signal. Transmitting several pulse complexes with variations in the LF pulse, and processing the received HF signal from these pulses allow for improved suppression of the multiple scattering noise. The processing can typically include a delay correction and a filtering to correct for the variations in the signal delay and speckle with variations of the LF pulse.

A target for this theses is to estimate the filter to correct for variations in the speckle of the multiple scattered signal, and the thesis shall examine the use of parameter estimation in models describing the variation of the speckle with variations in the LF pulse.

The work contains the following parts:

1. Describe and discuss both physically based and heuristic parametric models that potentially could be used to estimate the speckle correction filter. Describe and discuss methods of parameter estimation for adapting the models to physical measurements. Select particular models and estimation procedures to be further studied in this work, with special emphasis on observability of the parameters.

2. Develop an efficient computer program, potentially using parallel processing in GPUs, for adaptation of the models to measurements, and test the program on simulated measured signals, or potentially also on real measured signal if this becomes available during the work.

3. Derive speckle filters from the estimated model parameters, and test how such filters are able to improve suppression of multiple scattering noise on simulated measured signals, or potentially also on real measured signal if this becomes available during the work.

Abstract

In ultrasound imaging, multiple scattering noise can be severely damaging to the image quality. SURF imaging is a non-linear imaging method that can be used to suppress the multiple scattering noise, and thus improve the image quality. The suppression is done by processing methods involving a delay correction and a speckle correction. In this thesis, methods for speckle correction have been investigated on simulated signals.

The underlying effects that cause the speckle change in SURF signals have been investigated, with emphasis on an effect known as pulse form distortion. A model for describing the pulse form distortion have been tested, and proved to perform well, with the error between the model and the simulations typically well under -30dB. It was also shown a linear relationship between two of the parameters in the model, which can possibly be utilized for finding the phase of the pulse form distortion.

Three different methods for performing speckle correction have been studied, and compared to traditional methods that only uses a delay correction. Two of them yielded positive results, the average non-linear phase method and the physical speckle model. The physical speckle model is very complex however, and the performance dropped much when non-ideal model parameters were used. Therefore, the average non-linear phase method is concluded to be the most promising, although it must be tested in more realistic scenarios. The achieved improvement in signal-to-noise ratio with this method was shown to be around 3-5dB better than the traditional methods.

Sammendrag

I ultralydavbildning er multippel spredning en kilde til støy som skader bildekvaliteten. SURF avbildning er en ikke-lineær avbildningsmetode som kan brukes til å undertrykke støyen fra multippel spredning, og dermed forbedre bildekvaliteten. Undertrykkelsen utføres ved hjelp av prosesseringsmetoder som inkluderer en forsinkelseskorreksjon og en specklekorreksjon. I denne avhandlingen undersøkes metoder for specklekorreksjoner av simulerte signaler.

De underliggende effektene som skaper speckleforandringer i SURF signalene har blitt studert, med fokus på en effekt kjent som pulsformforvrengning. En modell som beskriver pulsformforvrengningen har blitt testet, og resultatene er gode. Feilen mellom modellen og simuleringene er typisk godt under -30dB. Det er også blitt vist en lineær sammenheng mellom to av parameterne i modellen, og denne sammenhengen kan mulig brukes for å finne fasen til pulsformforvrengningen.

Tre forskjellige metoder for specklekorreksjon har blitt studert, og sammenliknet med tradisjonelle metoder som kun bruker en forsinkelseskorreksjon. To av metodene gav lovende resultater. Disse metodene benytter henholdsvis en gjennomsnittlig ikke-lineær fase og en fysisk specklemodell. Den fysiske specklemodellen er derimot veldig kompleks, og resultatene ble dårlige når ikke-ideelle parametere ble brukt i modellen. Det ble derfor konkludert med at metoden med den gjennomsnittlige ikke-lineære fasen er den mest lovende, selv om det fortsatt gjenstår å teste denne i mer realistiske scenarier. Resultatene med denne metoden viser en forbedringen i signal-støy-forholdet på 3-5dB sammenliknet med tradisjonelle metoder.

Acknowledgments

First of all I would like to thank my supervisor, prof. Bjørn Angelsen for letting me join the SURF group, and for his enthusiastic guidance. It has been great fun to work with the exciting technology that is being developed in this group. A special thanks also goes to the PhD students in the SURF group, Ole Martin Brende, Johannes Kvam and Ola Myhre. Thank you very much for all your help, and for generally being great guys. I would also like to thank Dr. Anne C. Elster for letting me use the High Performance Computing Lab at IDI for my simulations. Last but not least, a big thank you goes to some fantastic friends and fellow students that took the time to proofread my report. In no special order, they are: Ali Fatemi, Bjørn Åge Tungesvik and Danial Jacobsen Hasan. Thank you!

Contents

List of Figures

List of Tables

List of Abbreviations

DCS	Delay Correction and Subtraction
DSCS	Delay and Speckle Correction and Subtraction
HF	High Frequency
LF	Low Frequency
PFD	Pulse Form Distortion
SNR	Signal-to-noise ratio

Chapter 1

Introduction

1.1 Background and motivation

A typical ultrasound image consists of three components[4, 3]: The linearly backscattered signal, X_l, the non-linearly backscattered signal, X_{nl}, and multiple scattering noise, N. The total signal can thus be written as the sum:

$$Y = X_l + X_{nl} + N \tag{1.1}$$

Depending on the application of the ultrasound image, it is often desirable to suppress all other components except the one which is relevant for the application, because the other components can obscure important information. In particular, multiple scattering is severely damaging for image quality because it creates false echoes in the image, which makes interpretation difficult. The image in figure 1.1 should justify the motivation for wanting to remove multiple scattering noise. The image shows a cross section of the carotid artery in the neck. The inside of the artery (marked in the image) is completely corrupted with multiple scattering noise, making it difficult to detect the edges of the artery and discover plaque.

This thesis will only examine simulated signals where $X_{nl} = 0$, i.e the signal consists only of a linearly backscattered component and multiple scattering noise:

$$Y = X_l + N \tag{1.2}$$

The goal is then to remove the multiple scattering noise from the signal, leaving only the linearly backscattered component.

To separate the components of an ultrasound image, non-linear imaging techniques must typically be used. An example of such a technique that is widely used today, is harmonic imaging[10, Ch. 12.5]. In terms of noise suppression however, this technique is mainly effective when the noise components are originating from shallow depths. SURF imaging is a new non-linear imaging technique which gives a better separation of the different signal components, also those originating from deeper depths. The details of how SURF imaging works will be described in chapter 2, but in order to introduce the subject of the thesis, the main concept must be known. With SURF imaging, the separation consists of two factors, a non-linear propagation delay and a speckle change. The work that has been done up to this point has mainly focused on the delay factor. Methods for estimation of the delay, and signal processing methods for suppressing the noise based on this delay have been developed, and is yielding good results[8, 5]. Even better results are theoretically achievable if the processing methods also correct for the change in speckle. Estimation of a filter for speckle correction is quite challenging, however.

1.2 Thesis topic

The topic of this thesis is the aforementioned speckle correction filter. In particular, three different estimation and correction methods will be explored. The achieved results will be compared to results where only a delay correction have been used. A special emphasis will be put on studying the effects that cause the change in speckle in SURF signals, in particular an effect known as pulse form distortion. The work will be carried out on simulated signals, where the multiple scattering noise is caused by plane reflectors.

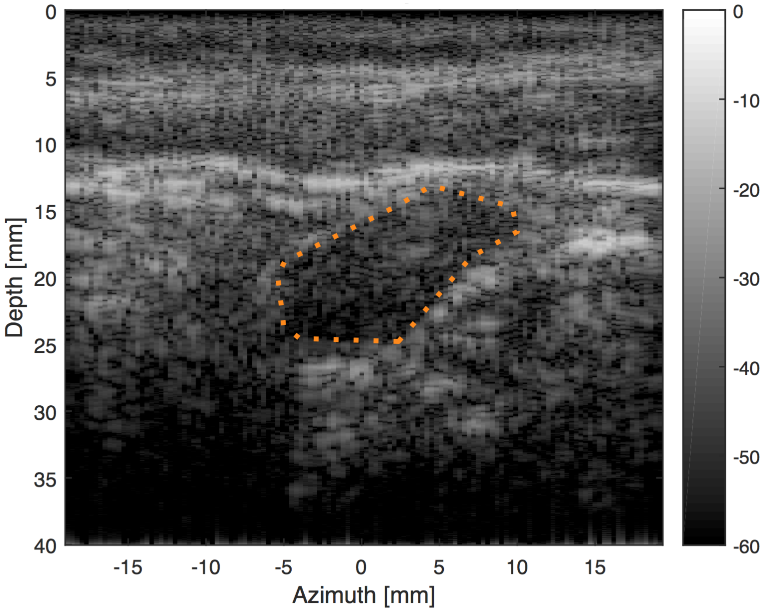


Figure 1.1: Ultrasound image of the carotid artery, corrupted with multiple scattering noise

1.3 Outline

The structure of the thesis is as follows:

Chapter 2 presents some theory related to SURF imaging and reverberations, and introduces the signal models and processing methods that form the basis of this thesis

Chapter 3 describes the methods and parameters used for simulation of ultrasound signals

Chapter 4 describes the methods used for delay estimation, and presents the achieved results using delay correction only

Chapter 5 presents a study in pulse form distortion

Chapter 6 goes trough the different methods that have been tried for estimation of the speckle correction filter, with an evaluation of the performance of each

Chapter 7 concludes the thesis, and presents some ideas that could be interesting to explore in future work

Chapter 2

Theory

2.1 Multiple scattering

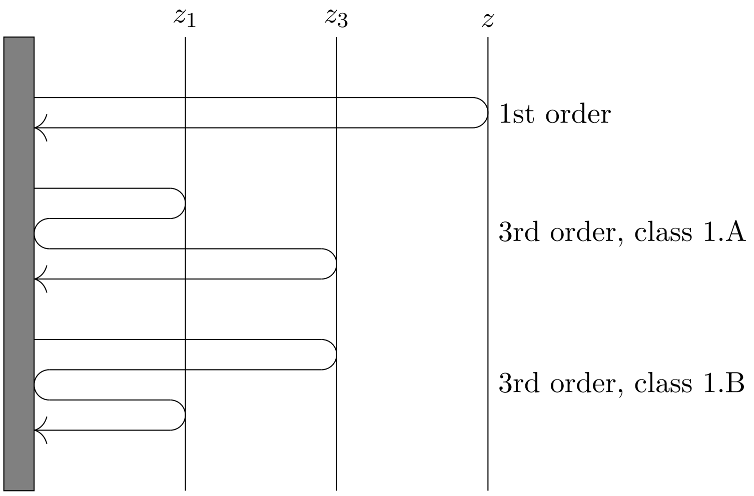


Figure 2.1: 1st order and 3rd order signal paths

Multiple scattering noise, also called reverberation noise, comes from the transmitted pulse being reflected multiple times between tissue layers, or tissue layers and the transducer surface, before being received by the transducer. This is illustrated in figure 2.1. The pulse can be reflected back and forth many times, but for each reflection the amplitude will drop. The 3rd order noise (3rd order = three reflections) will therefore be dominating, and will be the only one considered in this thesis. The reverberation noise can also be divided into different classes depending on where the reflections take place[6][1]. This thesis will only consider class 1 noise, in which the second reflection is

[1] Note that the naming conventions used in the reference have later been changed

always at the transducer surface.

With reference to figure 2.1, if there is a reflective plane at depth z, the 1st order signal (the true echo) from this plane comes from a pulse propagating directly to z and back to the transducer. If there also are planes at z_1 and z_3, where $z_1 + z_3 = z$, then the total 3rd order propagation distance between these planes, as shown in the figure, is the same as for the 1st order signal from z. The 3rd order signal will thus be interpreted as originating from z, and will create a false echo in the image. These false echoes can be severely damaging to the image quality, as was illustrated in figure 1.1.

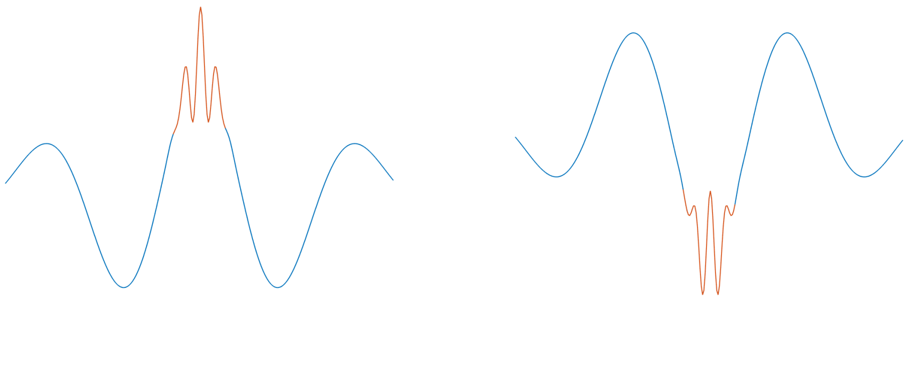
Class 1 noise can be further divided into class 1A and class 1B. In class 1A, the first reflection happens at $z_1 < z/2$, and the third reflection at $z_3 > z/2$. For class B it is the opposite. These two classes will always occur together, an important factor in surf imaging that will be discussed in the next section.

2.2 SURF imaging

The principle behind SURF imaging is that wave propagation through soft tissue is non-linear, which causes the propagation speed to be pressure dependent [10, ch. 12]. SURF imaging utilizes this phenomenon through a dual frequency pulse complex, with a high frequency (HF) imaging pulse co-propagating with an overlapping low frequency (LF) manipulation pulse that changes the propagation speed of the HF pulse[4]. The modified propagation speed caused by the LF pulse can be approximated as[4]:

$$c(p_L) = c_0(1 + \beta_p p_L) \tag{2.1}$$

where c_0 is the propagation speed without manipulation pressure, p_L is the LF amplitude and β_p a non-linearity parameter defined by the material.



(a) Positive SURF pulse

(b) Negative SURF pulse

Figure 2.2: Examples of SURF pulses

Typically three pulses are transmitted, one with positive manipulation pressure (figure 2.2a), one with negative manipulation pressure (figure 2.2b) and one without manipulation pressure. The pulses with positive and negative manipulation pressure propagates faster and slower respectively than the stand-alone HF pulse, and thus a delay between the pulses develops with depth.

The delay only develops until the first reflection however, as illustrated by the red arrows in figure 2.3. The amplitude after the first reflection has dropped so much that non-linear effects are negligible. This means that the 1st order signal and 3rd order signals will arrive at the transducer with different delays, which gives us a means of separating them. With reference to figure 2.3, the 1st order signal will have the delay $\tau_\pm(z)$, the class 1A 3rd order signal will have the delay $\tau_\pm(z_1)$ and class 1B will have the delay $\tau_\pm(z_3)$, with τ_+ denoting the delay with positive manipulation pressure and τ_- denoting the delay with negative manipulation pressure. This gives the following equation set for the total received signal for the three transmitted pulses, in the frequency domain:

$$\begin{aligned} Y_+(\omega) =& e^{-i\omega\tau_+(z)}X(\omega,z) + e^{-i\omega\tau_+(z_1)}N_A(\omega,z_1,z_3) + e^{-i\omega\tau_+(z_3)}N_B(\omega,z_1,z_3) \\ Y_0(\omega) =& X(\omega,z) + N_A(\omega,z_1,z_3) + N_B(\omega,z_1,z_3) \\ Y_-(\omega) =& e^{-i\omega\tau_-(z)}X(\omega,z) + e^{-i\omega\tau_-(z_1)}N_A(\omega,z_1,z_3) + e^{-i\omega\tau_-(z_3)}N_B(\omega,z_1,z_3) \end{aligned} \tag{2.2}$$

with $X(\omega,z)$ being the 1st order signal from the reflector at z, and $N_A(\omega,z_1,z_3)$ and $N_B(\omega,z_1,z_3)$ being the class A and class B 3rd order signals respectively, from the pair of reflectors at z_1 and z_3. The subscripts $+/0/-$ denote positive, zero, and negative manipulation pressure, respectively.

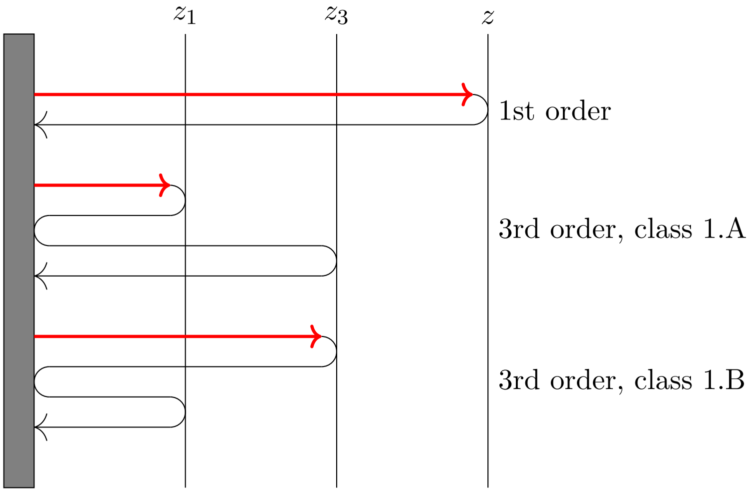


Figure 2.3: Non-linear signal paths

So far it has been assumed that the only effect of applying a manipulation pressure is a propagation delay. This would only be true if the manipulation pressure had been constant across the HF pulse, which is not the case. A variation in manipulation pressure across the HF pulse will cause different parts of the pulse to propagate at different speeds. This will cause an accumulating pulse form distortion (PFD) which will be studied in detail in chapter 5. Representing this distortion as a filter $\tilde{V}_{\pm}(\omega, z)$, equation set 2.2 will be modified as:

$$\begin{aligned} Y_+(\omega) = {} & e^{-i\omega\tau_+(z)}\tilde{V}_+(\omega,z)X(\omega,z) \\ & + e^{-i\omega\tau_+(z_1)}\tilde{V}_+(\omega,z_1)N_A(\omega,z_1,z_3) + e^{-i\omega\tau_+(z_3)}\tilde{V}_+(\omega,z_3)N_B(\omega,z_1,z_3) \end{aligned} \tag{2.3}$$

$$Y_0(\omega) = X(\omega,z) + N_A(\omega,z_1,z_3) + N_B(\omega,z_1,z_3)$$

$$\begin{aligned} Y_-(\omega) = {} & e^{-i\omega\tau_-(z)}\tilde{V}_-(\omega,z)X(\omega,z) \\ & + e^{-i\omega\tau_-(z_1)}\tilde{V}_-(\omega,z_1)N_A(\omega,z_1,z_3) + e^{-i\omega\tau_-(z_3)}\tilde{V}_-(\omega,z_3)N_B(\omega,z_1,z_3) \end{aligned}$$

Extracting an average noise delay, $\frac{\tau(z_1)+\tau(z_3)}{2}$, and writing $N_A(\omega,z_1,z_3)+N_B(\omega,z_1,z_3) = N(\omega,z_1,z_3)$, equation set 2.3 can be rewritten as:

$$Y_+(\omega) = e^{-i\omega\tau_+(z)}\tilde{V}_+(\omega,z)X(\omega,z) + e^{-i\omega(\frac{\tau_+(z_1)+\tau_+(z_3)}{2})}\tilde{L}_+(\omega,z_1,z_3)N(\omega,z_1,z_3) \tag{2.4}$$

$$Y_0(\omega) = X(\omega,z) + N(\omega,z_1,z_3)$$

$$Y_-(\omega) = e^{-i\omega\tau_-(z)}\tilde{V}_-(\omega,z)X(\omega,z) + e^{-i\omega(\frac{\tau_-(z_1)+\tau_-(z_3)}{2})}\tilde{L}_-(\omega,z_1,z_3)N(\omega,z_1,z_3)$$

with the $\tilde{L}$-filter, defined as:

$$\begin{aligned} & \tilde{L}_{\pm}(\omega,z_1,z_3) = \\ & \frac{e^{-i\omega(\frac{\tau_{\pm}(z_1)-\tau_{\pm}(z_3)}{2})}\tilde{V}_{\pm}(\omega,z_1)N_A(\omega,z_1,z_3) + e^{i\omega(\frac{\tau_{\pm}(z_1)-\tau_{\pm}(z_3)}{2})}\tilde{V}_{\pm}(\omega,z_3)N_B(\omega,z_1,z_3)}{N_A(\omega,z_1,z_3)+N_B(\omega,z_1,z_3)} \end{aligned} \tag{2.5}$$

representing a speckle filter, i.e. a change in the interference pattern between class A and class B noise. Changes in speckle will be discussed further in section 2.3.

In the ideal case, the delay development is linear with depth, $\tau(z) = az$, for some constant a. Writing z_3 as $z - z_1$, the average noise delay is then:

$$\frac{\tau(z_1)+\tau(z_3)}{2} = \frac{az_1 + a(z-z_1)}{2} = \frac{az}{2} = \frac{\tau(z)}{2} \tag{2.6}$$

i.e, the average noise delay is half the 1st order delay for any combination of z_1 and z_3.

2.2.1 Processing

The purpose of the processing is to combine the three signals in equation set 2.4 in a way that suppresses the reverberation noise, leaving only the 1st order component. Two different processing methods can be used:

Delay Corrected Sum (DCS)

With the DCS method, one only corrects for the average noise delay, ignoring the speckle filter. Defining the average noise delay as $\tau_n(z) = \frac{\tau(z_1)+\tau(z_3)}{2}$, this processing can be done as:

$$\hat{X} = Y_+ e^{i\omega\tau_{n+}} - Y_- e^{i\omega\tau_{n-}} \tag{2.7}$$

$$= (e^{-i\omega(\tau_+ - \tau_{n,+})}\tilde{V}_+ - e^{-i\omega(\tau_- - \tau_{n,-})}\tilde{V}_-)X + (\tilde{L}_+ - \tilde{L}_-)N \tag{2.8}$$

where the (ω, z) parameters have been dropped for simplicity.

Provided with a good estimate of the average noise delay, and assuming that $\tilde{L}_+$ and $\tilde{L}_-$ are almost equal, this type of processing gives a good, but not perfect suppression of the noise. As mentioned in the introduction, this type of processing has already yielded good results[8, 5], but even better results could be achieved by also correcting for the speckle filter.

Delay and Speckle Corrected Sum (DSCS)

With the DSCS method, one also corrects for the change in noise speckle. This processing can be done as:

$$\hat{X} = Y_+ e^{i\omega\tau_{n+}} \tilde{L}_+^{-1} - Y_- e^{i\omega\tau_{n-}} \tilde{L}_-^{-1} \tag{2.9}$$

$$= (e^{-i\omega(\tau_+ - \tau_{n+})}\tilde{V}_+ \tilde{L}_+^{-1} - e^{-i\omega(\tau_- - \tau_{n-})}\tilde{V}_- \tilde{L}_-^{-1})X \tag{2.10}$$

Theoretically, the noise can be completely removed using this method, but the challenge is to find good estimates for the speckle filter, which is the subject of this thesis.

2.2.2 SNR gain

The performance of the processing methods is measured by the increase in signal-to-reverberation noise ratio, or SNR gain for short, after processing relative to without processing. With simulated signals one can process the 1st order signal and the noise signal separately. In time domain, the instantaneous SNR gain can thus be calculated as

$$SNRG(t) = 20\log_{10}\left(\frac{\text{env}(x_{processed}(t))}{\text{env}(n_{processed}(t))}\right) - 20\log_{10}\left(\frac{\text{env}(x_0(t))}{\text{env}(n_0(t))}\right) \tag{2.11}$$

where $env()$ denotes the envelope of the signal.

2.3 Continuous signals and speckle

Up until now, the discussed theory has considered individual pulses. A real ultrasound signal however, is a continuous signal composed of overlapping pulses. Since ultrasound pulses have a certain length, received pulses from scatterers that are located close together will overlap and create a random interference pattern, called a speckle pattern. Using SURF imaging, the relative positioning between these overlapping pulses will change, due to the continuous delay development with depth, and the pulse shapes can be different due to PFD. This changes the speckle pattern.

To process continuous signals, they are typically divided into small intervals, such that within each interval, the signal properties are approximately stationary. Let the subscript i denote depth interval number i. Each interval can be modeled equivalently to equation set 2.4,

$$Y_{+,i}(\omega) = e^{-i\omega\tau_{+,i}}\tilde{V}_{+,i}(\omega)X_i(\omega) + e^{-i\omega\tau_{n+,i}}\tilde{L}_{+,i}(\omega)N_i(\omega) \tag{2.12}$$

$$Y_{0,i}(\omega) = X_i(\omega) + N_i(\omega)$$

$$Y_{-,i}(\omega) = e^{-i\omega\tau_{-,i}}\tilde{V}_{-,i}(\omega)X_i(\omega) + e^{-i\omega\tau_{n-,i}}\tilde{L}_{-,i}(\omega)N_i(\omega)$$

and equation 2.7 or 2.9 can be used to process the signals. The difference is that the filters $\tilde{V}_{\pm,i}(\omega)$ and $\tilde{L}_{\pm,i}(\omega)$ are more complex than their single pulse counterparts, since they now include a speckle change for all pulses that contribute within the interval.

Alternatively, the delay correction of continuous signals can be done using an interpolation scheme in the time domain without dividing the signal into intervals. The DCS processing can then be done as:

$$\hat{x}(t) = y_+(t + \tau_{n+}(t)) - y_-(t + \tau_{n-}(t)) \tag{2.13}$$

2.3.1 Noise speckle

The noise speckle is quite complex due to the huge amount of scatterer combinations that can contribute to the total noise at any depth, and due to noise pulses always occurring in pairs (class A and B). Any combination of scatterer pairs,n(z_1, z_3), for which $z_1 + z_3 = z$, will contribute to the total noise at depth z.

Class A and B differences

In [4] it shown that for a pair of point scatterers at $(\vec{r}_1, \vec{r}_3)$, and with linear propagation, the relation between class A and class B 3rd order noise can be written as

$$N_A(\omega, \vec{r}_1, \vec{r}_3) = Q(\omega, \vec{r}_1, \vec{r}_3) N_B(\omega, \vec{r}_1, \vec{r}_3) \tag{2.14}$$

$$\tag{2.15}$$

with $Q(\omega, \vec{r}_1, \vec{r}_3)$ representing the differences in the transmit and receive beam profiles,

$$Q(\omega, \vec{r}_1, \vec{r}_3) = \frac{H_t(\omega, \vec{r}_1) H_r(\omega, \vec{r}_3)}{H_r(\omega, \vec{r}_1) H_t(\omega, \vec{r}_3)} \tag{2.16}$$

With plane reflectors, as are used in this thesis, it can be shown that $Q = 1$. This is because a plane reflector does not change the shape of the pulse, but rather acts like a mirror, extending the transmit beamprofile across the whole propagation distance. Since class A and B have equal propagation distance, they will be equal when they arrive at the transducer,

$$N_A(\omega, z_1, z_3) = N_B(\omega, z_1, z_3) = N_l(\omega, z_1, z_3) \tag{2.17}$$

The math behind this can be found in [4].

With non-linear propagation however, class A and class B will be different due to an unequal amount of self-distortion. Just as the delay with SURF imaging only develops until the first reflection, so does the self-distortion. Self-distortion will move power from the fundamental band to the harmonics, and hence cause an additional attenuation of the fundamental band [10, ch. 12]. The harmonics can be filtered away, but the additional attenuation in the fundamental band will cause class A to be stronger than class B. Representing the accumulated non-linear attenuation at depth z with the filter $\alpha(\omega, z)$, class A and B can be written as

$$N_A(\omega, z_1, z_3) = \alpha(\omega, z_1) N_l(\omega, z_1, z_3) \tag{2.18}$$

$$N_B(\omega, z_1, z_3) = \alpha(\omega, z_3) N_l(\omega, z_1, z_3) \tag{2.19}$$

Finally, there will be a difference between class A and B due to unequal delay and PFD when using SURF imaging, as described in section 2.2:

$$N_A(\omega, z_1, z_3) = \alpha(\omega, z_1) e^{-i\omega\tau_\pm(z_1)} \tilde{V}_\pm(\omega, z_1) N_l(\omega, z_1, z_3) \tag{2.20}$$

$$N_B(\omega, z_1, z_3) = \alpha(\omega, z_3) e^{-i\omega\tau_\pm(z_3)} \tilde{V}_\pm(\omega, z_3) N_l(\omega, z_1, z_3) \tag{2.21}$$

Physical speckle model

The total received noise signal as a function of fast time, $n(t)$, can be written as an integral over the noise pulses from all combinations of scatter pairs (z_1, z_3) that overlap at fast time t[3]. Fast time is the time after transmission of the initial pulse. A pulse from a specific scatter pair (z_1, z_3), has a total propagation distance of $2(z_1 + z_3) = 2z$, and will be received at the transducer after a time $\frac{2z}{c}$, where c is the propagation speed.

Let $R(z)$ define the reflection coefficients as a function of depth, and the integral can be written as

$$n(t) = \int_0^{z_{max}} dz \int_0^{z/2} dz_1 R(z_1)R(z_3) \left(n_A(t - \frac{2z}{c}, z_1, z_3) + n_B(t - \frac{2z}{c}, z_1, z_3) \right) \tag{2.22}$$

where z_3 is defined as $z - z_1$, and z_{max} is the imaging depth. The reflection coefficient at the transducer surface is a constant and is ignored in this model. Dividing the signal into intervals, interval number i can be written as

$$n_i(t) = \int_{z_{l,i}}^{z_{u,i}} dz \int_0^{z/2} dz_1 R(z_1)R(z_3) \left(n_A(t - \frac{2z}{c}, z_1, z_3) + n_B(t - \frac{2z}{c}, z_1, z_3) \right) \tag{2.23}$$

where $z_{l,i}$ and $z_{u,i}$ defines the lower and upper limit respectively, of interval number i. The integral can be written in frequency domain as

$$N_i(\omega) = \int_{z_{l,i}}^{z_{u,i}} dz \int_0^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} \left(N_A(\omega, z_1, z_3) + N_B(\omega, z_1, z_3)\right) \tag{2.24}$$

Inserting the differences between class A and class B noise with zero LF pressure, as given in equation 2.18 and 2.19 , the integral with zero LF pressure can be written as

$$N_{0,i}(\omega) = \int_{z_{l,i}}^{z_{u,i}} dz \int_0^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} N_l(\omega, z_1, z_3) \left(\alpha(\omega, z_1) + \alpha(\omega, z_3)\right) \tag{2.25}$$

$N_l(\omega, z_1, z_3)$ can be expressed by the combined transmit-receive linear beam transfer function for plane reflectors, $H(\omega, z_1 + z_3)$ and the initial pulse at the transducer, $P_t(\omega)$[3, 4], as

$$N_l(\omega, z_1, z_3) = P_t(\omega)H(\omega, z_1 + z_3) \tag{2.26}$$

Inserting this into the integral, but ignoring $P_t(\omega)$ as this is a constant, gives the final model for the noise signal with zero LF pressure as

$$N_{0,i}(\omega) = \int_{z_{l,i}}^{z_{u,i}} dz \int_0^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} H(\omega, z) \left(\alpha(\omega, z_1) + \alpha(\omega, z_3)\right) \tag{2.27}$$

Inserting the difference between class A and B noise with manipulation pressure, as given in equation 2.20 and 2.21, a model for the noise signal with LF pressure is given as

$$N_{+,i}(\omega) = \int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} H(\omega,z) \left(\alpha(\omega,z_1)e^{-i\omega\tau_+(z_1)}\tilde{V}_+(\omega,z_1) + \alpha(\omega,z_3)e^{-i\omega\tau_+(z_3)}\tilde{V}_+(\omega,z_3)\right) \tag{2.28}$$

$$N_{-,i}(\omega) = \int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} H(\omega,z) \left(\alpha(\omega,z_1)e^{-i\omega\tau_-(z_1)}\tilde{V}_-(\omega,z_1) + \alpha(\omega,z_3)e^{-i\omega\tau_-(z_3)}\tilde{V}_-(\omega,z_3)\right) \tag{2.29}$$

A model of the speckle filter in equation set 2.12, *with the average delay included*, is then found by:

$$L_{+,i}(\omega) = \frac{N_{+,i}(\omega)}{N_{0,i}(\omega)}, \qquad L_{-,i}(\omega) = \frac{N_{-,i}(\omega)}{N_{0,i}(\omega)} \tag{2.30}$$

$$L_{+,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} \left(\alpha(\omega,z_1)e^{-i\omega\tau_+(z_1)}\tilde{V}_+(\omega,z_1) + \alpha(\omega,z_3)e^{-i\omega\tau_+(z_3)}\tilde{V}_+(\omega,z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} \left(\alpha(\omega,z_1) + \alpha(\omega,z_3)\right)} \tag{2.31}$$

$$L_{-,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} \left(\alpha(\omega,z_1)e^{-i\omega\tau_-(z_1)}\tilde{V}_-(\omega,z_1) + \alpha(\omega,z_3)e^{-i\omega\tau_-(z_3)}\tilde{V}_-(\omega,z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} \left(\alpha(\omega,z_1) + \alpha(\omega,z_3)\right)} \tag{2.32}$$

Speckle correction based on these models are explored in chapter 6.

Special case

A special case arises under the following conditions:

- The PFD and the non-linear attenuation is negligible
- The delay development is linear with depth
- The magnitude of the delay is equal for both the positive LF pressure and the negative LF pressure, i.e. $\tau_+(z) = -\tau_-(z) = \tau(z)$

Under these conditions, the speckle model in equation 2.31 can be written as

$$L_{+,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}\left(e^{-i\omega\tau(z_1)} + e^{-i\omega\tau(z_3)}\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}} \tag{2.33}$$

By using the average delay defined in equation 2.6, one gets:

$$\begin{aligned} e^{-i\omega\tau(z_1)} + e^{-i\omega\tau(z_3)} &= e^{-\omega\frac{\tau(z)}{2}}\left(e^{-i\omega\frac{\tau(z_1)-\tau(z_3)}{2}} + e^{i\omega\frac{\tau(z_1)-\tau(z_3)}{2}}\right) \\ &= e^{-\omega\frac{\tau(z)}{2}} 2\cos\left(\omega\frac{\tau(z_1)-\tau(z_3)}{2}\right) \end{aligned} \tag{2.34}$$

The speckle model can thus be written as

$$L_{+,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz\ e^{-\omega\frac{\tau(z)}{2}} \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} 2\cos\left(\omega\frac{\tau(z_1)-\tau(z_3)}{2}\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}} \tag{2.35}$$

$$L_{-,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz\ e^{\omega\frac{\tau(z)}{2}} \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}} 2\cos\left(-\omega\frac{\tau(z_1)-\tau(z_3)}{2}\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}} \tag{2.36}$$

Since $\cos(x) = \cos(-x)$, the inner integral is equal for both L_+ and L_-. Therefore, by assuming that $\frac{\tau(z)}{2}$ varies little inside the interval, there is no change in speckle when changing the polarity of the LF pressure. The only difference between L_+ and L_- is then a delay.

The important point to grasp from all this is that it is the difference between class A and class B noise, as caused by PFD and non-linear attenuation, that causes the change in speckle with a positive LF pressure relative to with a negative LF pressure. This point is illustrated in figure 2.4 and 2.5. Figure 2.4 shows how class A and class B noise are delayed when applying a LF pressure. Figure 2.5 shows how the noise with a positive LF pressure overlaps with the noise with a negative LF pressure after delay correction. It can be seen that the positioning of class A relative class B is inverted with a negative LF pressure. This causes class A with positive LF pressure to overlap with class B with negative LF pressure, and vice versa. From this it can be seen that if class A and class B had been equal, perfect suppression of the noise could be achieved by subtracting the two signals. The difference between class A and B however, causes the suppression to be imperfect unless the speckle change also is corrected . Since the PFD plays an important part in making class A and class B different, it will be studied in more detail in chapter 5.

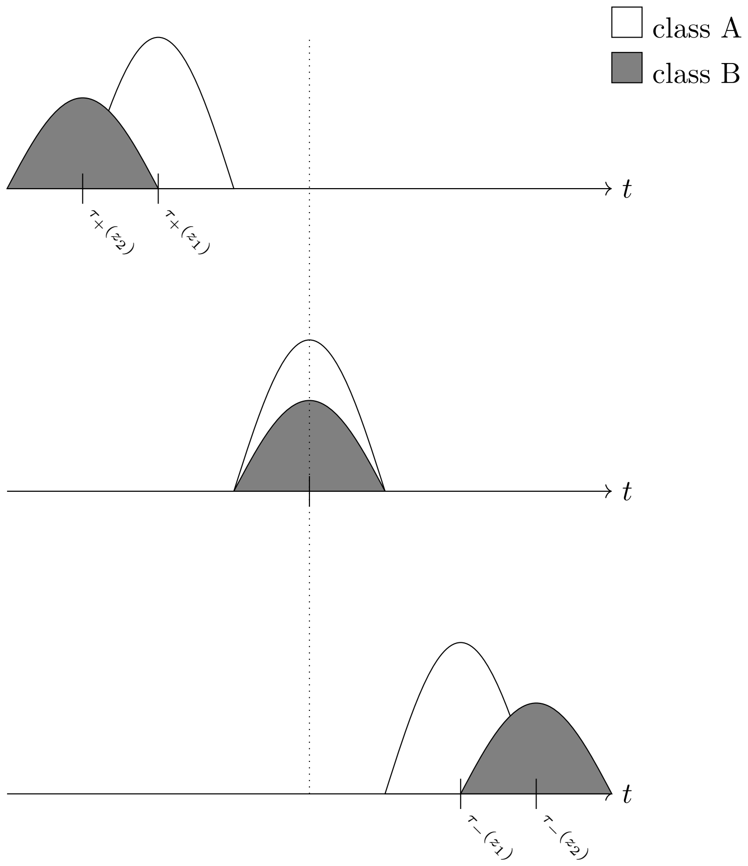


Figure 2.4: Illustration of the delays of class A and class B noise
Upper: positive LF pressure. Middle: zero LF pressure. Lower: negative LF pressure.

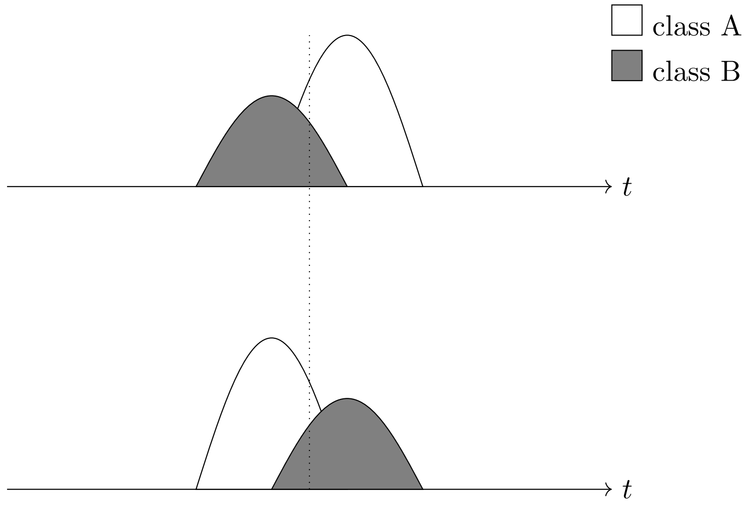


Figure 2.5: Illustration of the noise overlapping after delay correction
Upper: positive LF pressure. Lower: negative LF pressure.

Chapter 3

Simulation setup

All signals that are studied and processed in this thesis, are simulated signals, rather than real data. The benefit of using simulated signals is that one has access to the 1st order signal and the noise separately, and can hence study the effect of SURF imaging and the processing methods on each of them separately. Another advantage is that one can easily change the properties of the transducer and the transmitted pulse without being limited by physical hardware.

The signals are simulated up to a depth of 40mm, which is a realistic image depth for imaging f.ex. the carotid artery in the neck. The artery is typically located at around 20mm depth[2, ch.5]. The methods and parameters used for the simulation will be further described in the follwing sections.

3.1 Pulse simulation

To simulate ultrasound pulses, the same simulation software has been used as in the authors specialization project[9]. This software is essentially the same as described by Kvam[7], with the exception that the pulse is propagated as a compound pulse, instead of with the split method described in the reference.

The 1st order pulses are simulated by propagating a 3-dimensional pulse, initially defined at the transducer surface, non-linearly a distance z (forward propagation). Here the pulse hits a plane reflector, and is then propagated linearly a distance z (back-propagation), at which point the pulse is beamformed. Varying z from 1-40mm, with a step size of 1mm, gives a set of beamformed 1st order pulses, one pulse for each mm.

To simulate the 3rd order noise pulses, the initial pulse is first propagated non-linearly the distance to the first reflection, e.g. z_1 in figure 2.3, and then linearly a distance $z_1 + 2z_3$, before it is beamformed as if it was originating from the depth $z_1 + z_3$. This is repeated for every combination of z_1 and z_3 for which $z_1 + z_3 \in$ [1mm, 40mm], again with a step size of 1mm.

The simulations are repeated with a positive, a negative and zero manipulation pressure, resulting in three sets of pulses:

$$\text{1st order pulses: } \begin{cases} X_{+,z}(\omega) \\ X_{0,z}(\omega) & z \in [\text{1mm, 40mm}] \\ X_{-,z}(\omega) \end{cases}$$

$$\text{3rd order pulses: } \begin{cases} N_{+,z_1,z_3}(\omega) \\ N_{0,z_1,z_3}(\omega) & z_1 + z_3 \in [\text{1mm, 40mm}] \\ N_{-,z_1,z_3}(\omega) \end{cases}$$

with the subscripts $+/0/-$ denoting positive, zero, and negative manipulation pressure, respectively.

An important point to be made is that since the propagation after the first reflection is linear, the pulses can be multiplied with the reflection coefficients *after* beamforming. This means that the same simulation results can be used with different sets of reflection coefficients without running a new simulation, a feature that is utilized when creating continuous signals as described in the next section.

To extract only the HF pulse from the HF-LF pulse complex, a simulation has also been performed of a stand-alone positive and negative LF pulse. The results from this simulation have been subtracted from the pulse complex, leaving only the HF pulse.

The simulation parameters are presented in table 3.1 - 3.5.

Table 3.1: Pulse parameters

	HF	LF
Center frequency	9MHz	0.5MHz
Sampling frequency	100MHz	100MHz
Length	2 cycles	2 cycles
Amplitude	0.3MPa	1.2MPa
Shape	Gaussian	Gaussian
HF offset[1]	0°	

[1] Relative to the peak of the LF pulse

Table 3.2: Transmit transducer parameters

	HF	LF
Aperture size	7.6x5.1mm	16.2x10.5mm
Azimuth focus depth	22mm	unfocused
Elevation focus depth	22mm	unfocused [1]
Apdoization[2]	1	0
Geometry	Rectangular	Rectangular

[1] central part of the LF transducer is focused at 22mm due to the HF focusing lens

[2] Tukey window tapering factor

Table 3.3: Receive transducer parameters

	HF
Aperture size	Dynamic
Azimuth focus depth	Dynamic
Elevation focus depth	Dynamic
F-number	3.0
Apodization[1]	1
Geometry	Rectangular

[1] Tukey window tapering factor

Table 3.4: Domain parameters

Number of samples:	
Azimuth	128
Elevation	128
Temporal	1024
Spatial resolution:	
Azimuth	0.25mm
Elevation	0.25mm
Simulation depth	40mm
Step size	1mm

Table 3.5: Material parameters

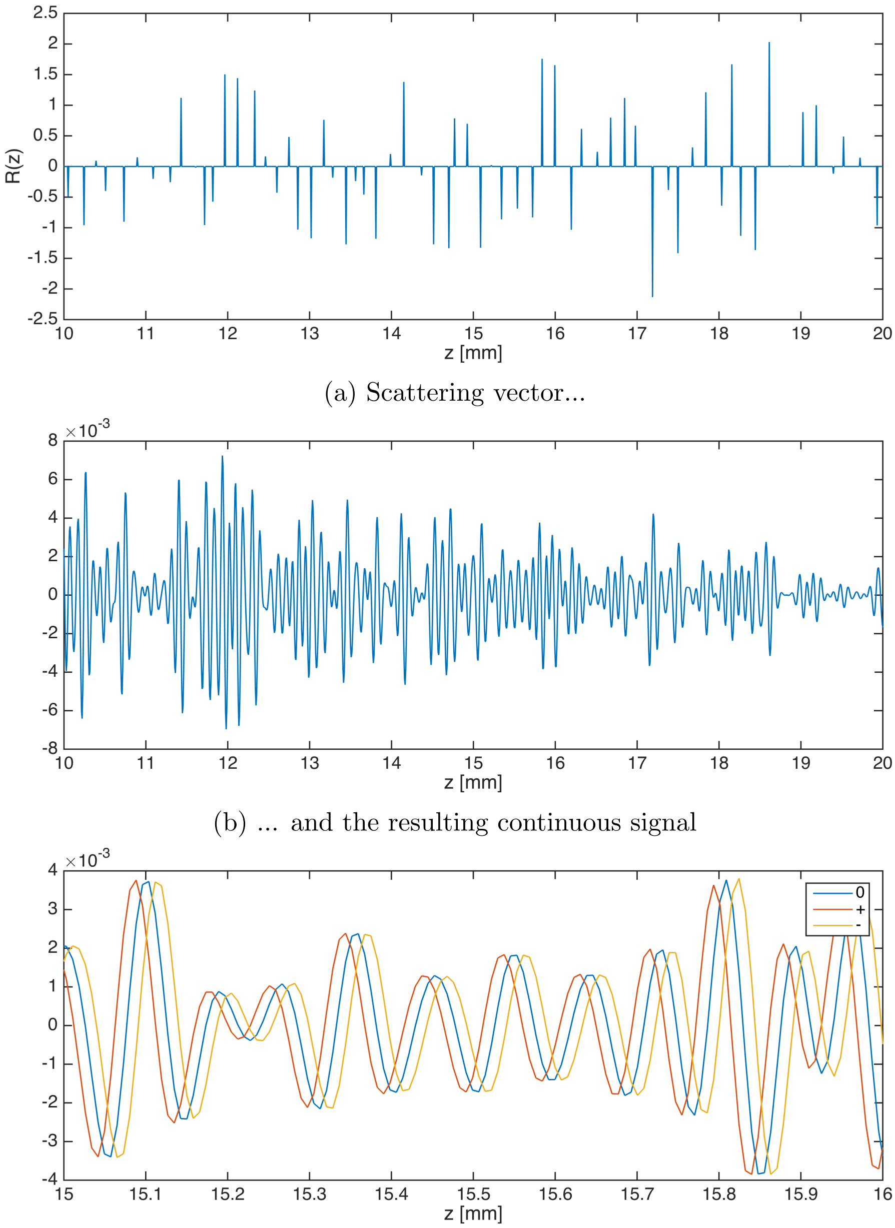
Tissue type	Muscle
Sound Speed (c_0)	$1549.9 \ \frac{\text{m}}{\text{s}}$
Compressibility (κ)	$3.9 \cdot 10^{-10} \frac{\text{ms}^2}{\text{kg}}$
Coefficient of non-linearity (β_n)	$3.9 \cdot 10^6$
Absorption power law	$0.52 \frac{\text{dB}}{\text{cm/MHz}} \cdot f^{1.1}$

3.2 Continuous signal simulation

A spacing of one millimeter between each pulse is too large for them to overlap and create a continuous signal with a realistic speckle pattern. Doing a pulse simulation with a smaller step size however, would take too too much time. So in order to create continuous signals, a method devised by PhD-student Ole Martin Brende[5] has been used. For the 1st order signal, this method convolves each of the simulated pulses with a scattering vector, $R(z)$, which defines the reflection coefficients as a function of depth. Each convolution is then multiplied with a sinusoidal weighting window, centered at the pulse simulation depth, with a width of 2mm. F.ex. the convolution of a pulse from $z = 20$mm with the scattering vector is multiplied with a window centered at 20mm, spanning 19-21mm. Finally, all the windowed convolutions are summed, resulting in a continuous signal where the signal corresponding to a specific scatterer $R(z)$ is composed of a weighting between the two closest simulated pulses.

To create the continuous noise pulses, two copies of $R(z)$ are made. One of them is multiplied with a window centered at z_1, the other with a window centered at z_3. Then these two windowed scattering vectors are convolved, creating a 3rd order scattering vector. This new scattering vector is then convolved with the simulated noise pulse from the scatterer pair (z_1, z_3). This is repeated for all combinations of z_1 and z_3 that have been simulated, and all results are summed. This will, just as with the 1st order signal, create a continuous noise signal where the signal from a specific scatter combination is composed of a weighting between the closest simulated pulses. More details and the math proving the validity of this approach can be found in[5].

The scattering vector is created with a gaussian amplitude distribution, and a poisson distribution for the distance between the scatterers. The average distance is 20 samples, approximately 0.15mm. An example of a scattering vector and the resulting continuous signal is shown in figure 3.1. To avoid random effects related to a particularly good or bad combination of scatterers when processing the signals, and rather get a impression of general trends, 30 different realizations of scattering vectors and the corresponding signals have been created, so that the processing results can be averaged over all the signals.



(a) Scattering vector...

(b) ... and the resulting continuous signal

(c) notice the delay between the signals with different manipulation pressures

Figure 3.1: Example of simulated continuous signal

Chapter 4

Delay estimation and correction

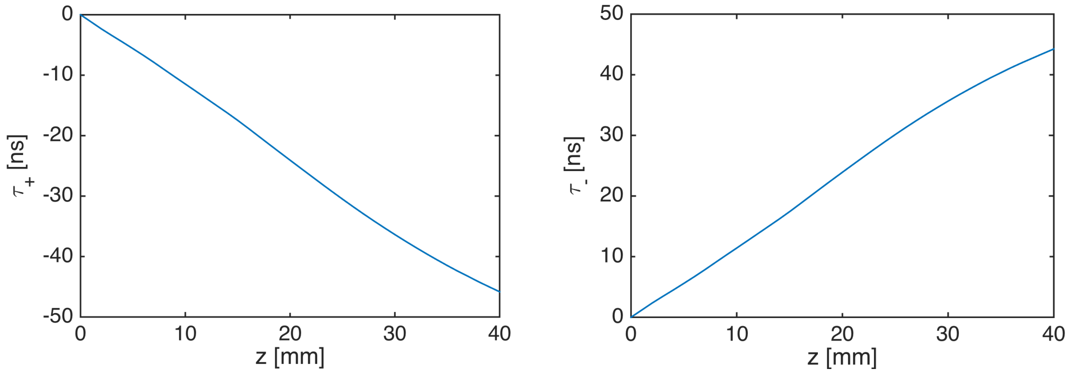
As the topic of this thesis is how noise suppression can be improved beyond using only a delay correction, by also correcting for speckle changes, no effort has been made in using realistic delay estimation methods. Instead, a perfect delay estimate have been found by doing the estimation directly on the simulated 1st order pulses. The 1st order delay is found by calculating the phase difference at the center frequency, ω_c:

$$\tau_+(z) = \frac{1}{\omega_c}\angle\frac{X_{+,z}(\omega_c)}{X_{0,z}(\omega_c)} \tag{4.1}$$

$$\tau_-(z) = \frac{1}{\omega_c}\angle\frac{X_{-,z}(\omega_c)}{X_{0,z}(\omega_c)} \tag{4.2}$$

yielding the delay curves shown in figure 4.1. These delays have then been upsampled to the same sampling frequency as the continuous signals.

As the material used in the simulation is homogeneous, the delay curves get quite linear.



(a) $\tau_+(z)$

(b) $\tau_-(z)$

Figure 4.1: Estimated delays

Therefore, the simple $\tau_n(z) = \tau(z)/2$ estimate for the noise delay have been used, as explained in 2.2.

The DCS processing has been done with a continuous delay correction (eq 2.13), implemented in matlab as follows:

Listing 4.1: DCS processing in matlab

```
% x0, n0 = 1.order and noise signal, no LF
% xp, np = 1.order and noise signal, positive LF
% xm, nm = 1.order and noise signal, negative LF
% t = time axis
% tp_corr = correction delay, positive LF
% tm_corr = correction delay, negative LF

%process 1.order and noise signal individually
x_processed  = interp1(t, xp, t+tp_corr, 'spline') ...
 - interp1(t, xm, t+ tm_corr, 'spline');

n_processed = interp1(t,  np, t+tm_corr, 'spline') ...
 - interp1(t, nm, t+ tm_corr, 'spline');

%calculate SNR gain
SNRG = 20*log10(abs(hilbert(x_processed)./hilbert(n_processed)))...
 - 20*log10(abs(hilbert(x0)./hilbert(n0)));

%smooth over 200 samples
SNRG = smooth(SNRG, 200);
```

The processing has been repeated for, and averaged over, all 30 simulated signals. The achieved average SNR gain, together with the standard deviation, is shown in figure 4.2. The results show between 20-30dB gain, depending on depth, with about 27dB in the focus. The drop in gain with depth can be attributed to an accumulation of PFD, which makes the speckle changes larger, as explained in section 2.3.1.

These results will function as a reference for the rest of this thesis, and will be compared to the results achieved with the new methods for speckle corrections, in the hope that the new methods yield even better gains.

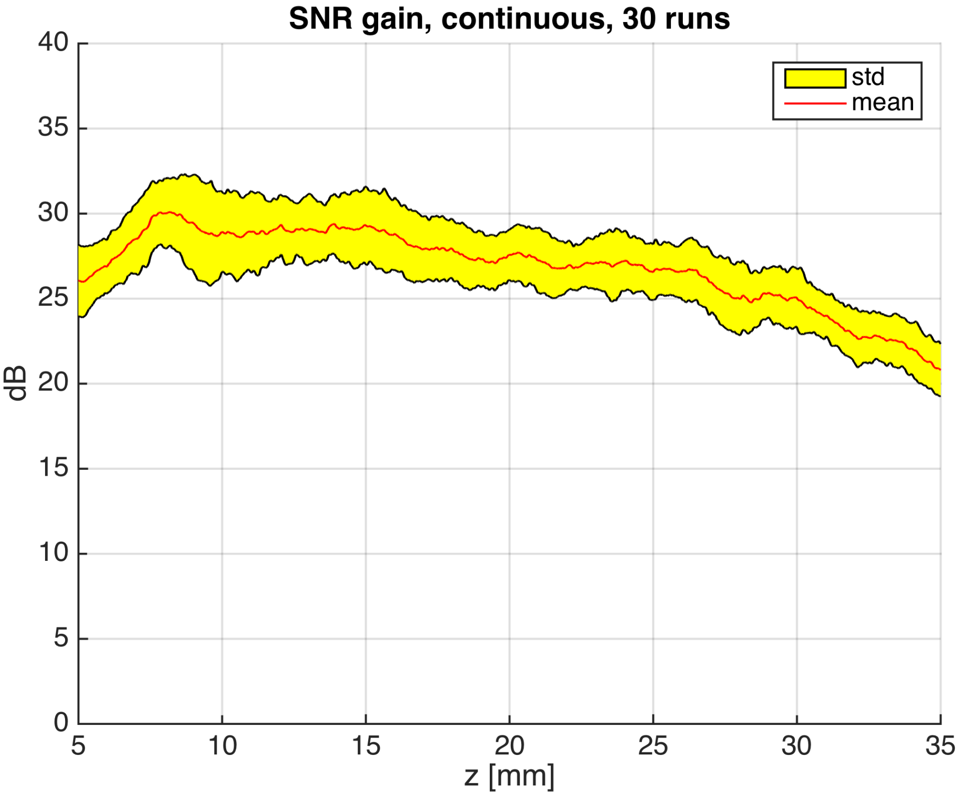


Figure 4.2: Average SNR gain and standard deviation with DCS processing

Chapter 5

Pulse form distortion

Since the PFD plays an important role in the speckle filter (eq. 2.31), it is important to have a good understanding of what it is, how it is generated, and how it can be modeled. All this will be studied in more detail in this chapter, together with a small study of how the PFD can be minimized by optimal positioning of the HF pulse on the LF pulse.

5.1 Description

As mentioned in the theory, the LF pressure is not constant across the HF pulse, causing local variations in the propagation speed of the HF pulse and a corresponding distortion of the pulse. Approximating the LF pressure across the HF pulse as a cosine, and doing a Taylor series expansion of this cosine up to the second degree polynomial, reveals the main effects the LF pulse has on the HF pulse[3]. By using retarded time, τ, defined such that $\tau = 0$ at the center of the HF pulse at any depth z, and letting $\tau_L(z)$ be the distance between the peak of the LF pulse and the center of the HF pulse, the Taylor series can be written as

$$p\cos(\omega_L(\tau + \tau_L)) = p\cos(\omega_L \tau_L(z)) - p\omega_L \tau \sin(\omega_L \tau_L(z)) - p/2(\omega_L \tau)^2 \cos(\omega_L \tau_L(z)) \tag{5.1}$$

where p is the amplitude and ω_L the center angular frequency of the LF pulse.

The first term of the taylor series is a constant. This is the term that generates the propagation delay. The second term is a linear gradient. The presence of this term means that one end of the pulse travels faster than the other. Depending on whether the gradient is positive or negative, this term will cause a compression or an expansion of the pulse. The third, quadratic term, causes the center of the pulse to travel faster than the edges. This results in a chirping of the pulse. These effects are illustrated in figure 5.1.

The two main PFD effects are hence a compression/expansion and a chirping of the pulse. The compression/expansion term is proportional to $\sin(\omega_L \tau_L)$. If the HF pulse is placed precisely on the peak of the LF pulse, so that $\tau_L = 0$, this term is zero. If the HF pulse is placed such that $\omega_L \tau_L = \pm\pi/2$, this term is at its maximum. The chirping term is proportional to $\cos(\omega_L \tau_L)$, and thus is at its maximum when the HF pulse is placed on the peak, and is zero when $\omega_L \tau_L = \pm\pi/2$.

To complicate matters further, the position of the HF pulse relative to the LF pulse is not constant, but changes with depth, i.e. $\tau_L(z)$ is a function of depth. This happens because the LF pulse experiences a phase shift as it propagates, which changes the location of the peak. This is equivalent to the HF pulse ”sliding” on the LF pulse as it propagates, as illustrated in figure 5.2. The phase shift have been measured on the simulated LF pulses, and is shown in figure 5.3. It can be seen that the phase shift is up to 50^o, meaning that both PFD terms in equation 5.1 will contribute to the total PFD as the pulse propagates, no matter where the HF pulse is placed initially.

When changing the polarity of the LF pulse ($p \to -p$ in equation 5.1), the PFD effects are opposite. For example, if the HF pulse is compressed with a positive LF, it will be expanded with a negative LF, and vice versa. Ideally, the magnitude of the terms in the Taylor series would still be equal for both polarities, but as is shown in figure 5.3, the phase shift of the negative LF is smaller than the phase shift of the positive LF. This seems to be a non-linear effect, because a similar measurement on a linear simulation shows the phase shifts to be equal. The $\omega_L \tau_L$ factors in equation 5.1 will thus not be perfectly equal when changing the polarity of the LF pulse.

To relate the PFD effects discussed so far to the PFD filter $\tilde{V}_\pm(\omega, z)$ used in the signal model in section 2.2, one can use the following property of the Fourier transform[1]:

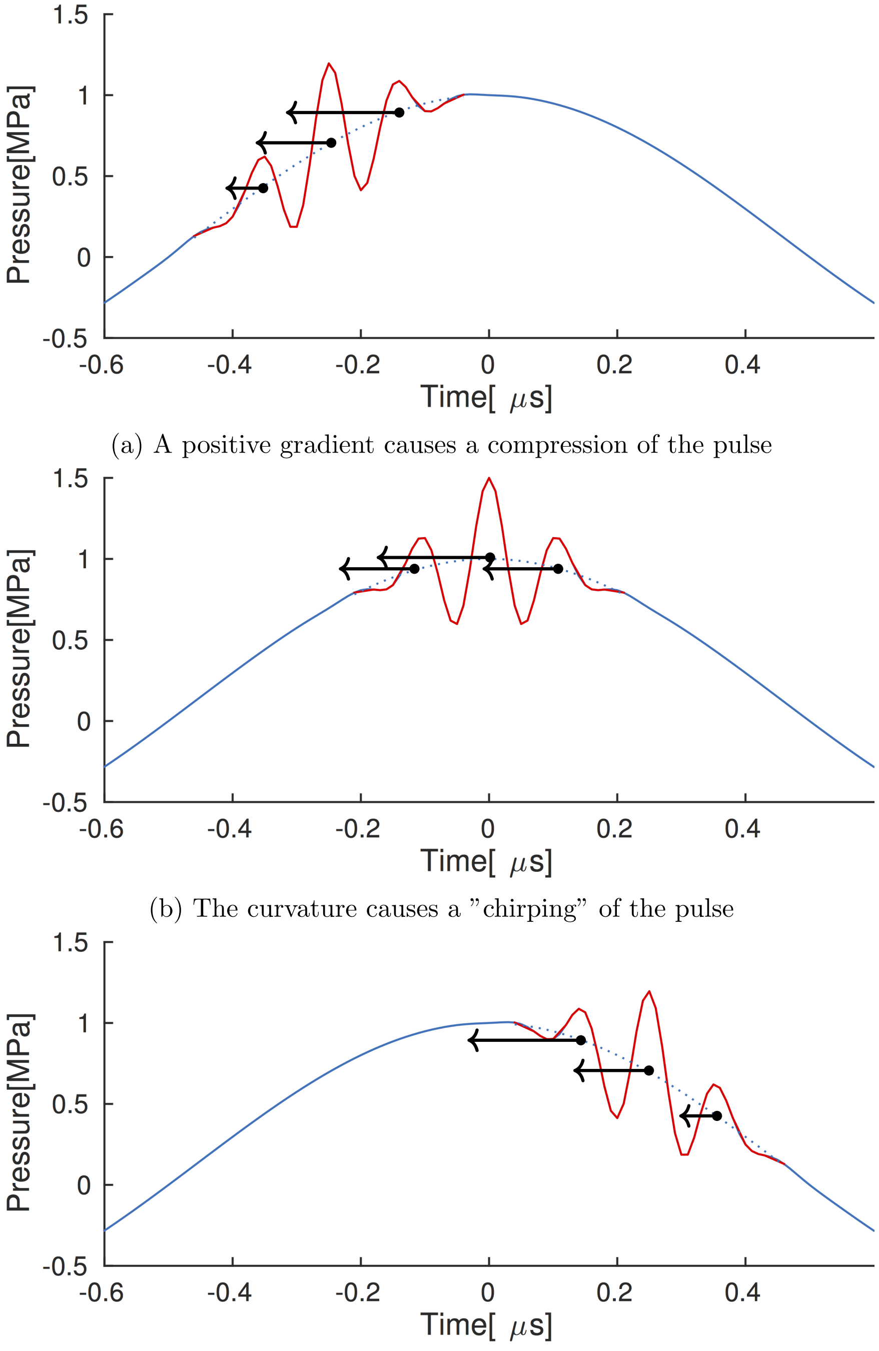
$$\begin{aligned} \mathcal{F}\{g(t)\} &= G(f) \\ \mathcal{F}\{g(ct)\} &= \frac{G(\frac{f}{c})}{|c|} \end{aligned} \tag{5.2}$$

A compression in time ($c > 1$) widens the spectrum and moves it to higher frequencies, while an expansion in time ($c < 1$) narrows the spectrum and moves it to lower frequencies. One should then expect the magnitude of the filter to boost higher frequencies and dampen lower frequencies in case of compression, and vice versa in case of expansion. Using the set of simulated pulses defined in section 3.1, the magnitude of the PFD filter can easily be calculated as:

$$|\tilde{V}_+(\omega, z)| = \left| \frac{X_{+,z}(\omega)}{X_{0,z}(\omega)} \right| \tag{5.3}$$

$$|\tilde{V}_-(\omega, z)| = \left| \frac{X_{-,z}(\omega)}{X_{0,z}(\omega)} \right| \tag{5.4}$$

These filters are plotted in figure 5.4 (each line represents one depth), together with a visualization of how the 6dB bandwidth of X_+, X_0, and X_- develops with depth. As expected, since the HF pulse slides behind the peak of the LF pulse as it propagates, X_+ gradually moves to lower frequencies and X_- to higher frequencies, compared to X_0.



(a) A positive gradient causes a compression of the pulse

(b) The curvature causes a ”chirping” of the pulse

(c) A negative gradient causes an expansion of the pulse

Figure 5.1: Local variations in propagation speed across the HF pulse

Note that the time axis is retarded time, the propagation direction is to the left. The length of the arrows represent the local propagation speed.

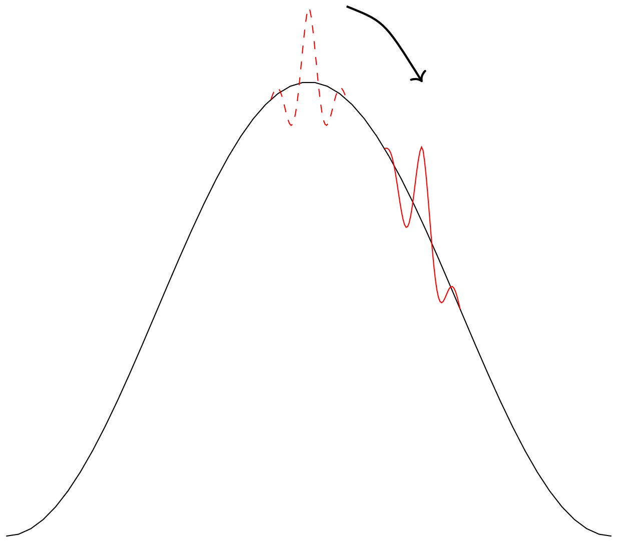


Figure 5.2: ”Sliding” of the HF pulse on the LF pulse

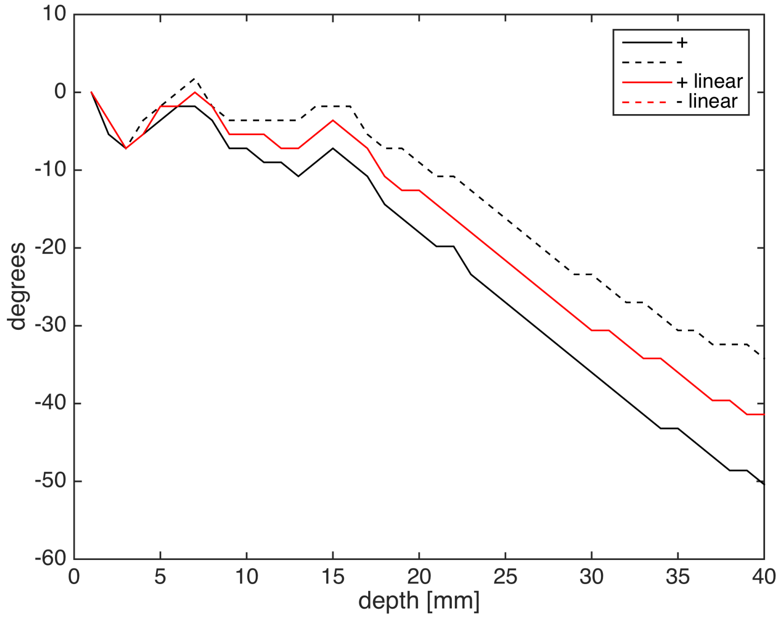
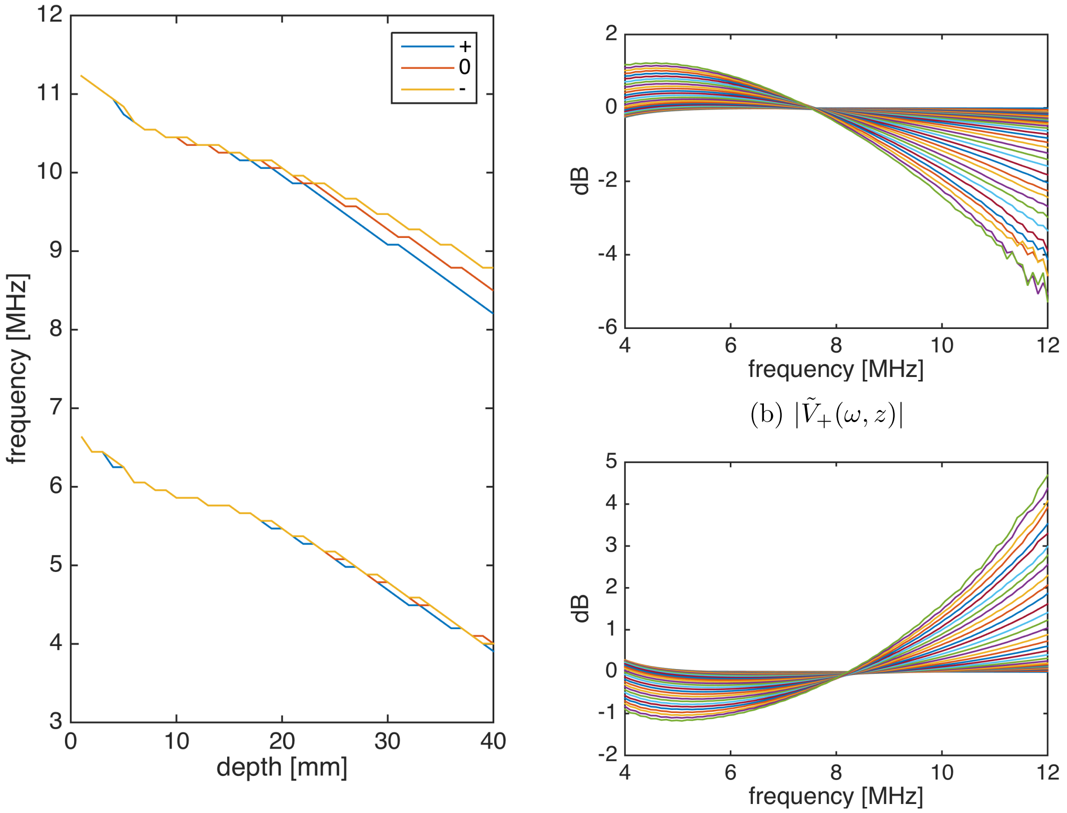


Figure 5.3: Measured phase shift of the LF pulse with depth

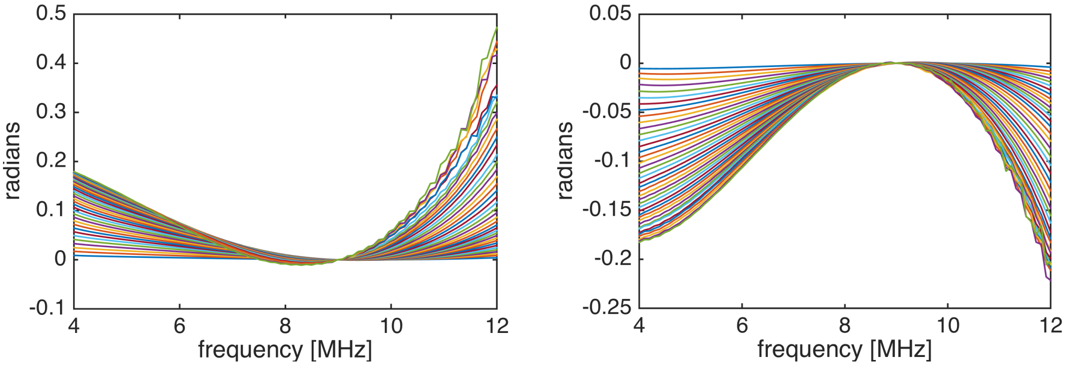


(b) $|\tilde{V}_{+}(\omega, z)|$

(c) $|\tilde{V}_{-}(\omega, z)|$

(a) 6dB bandwidth development with depth. *The distance between two lines of equal color is the bandwidth.*

Figure 5.4: Magnitude of the PFD filter



(a) $\angle\tilde{V}_{+}(\omega, z)$

(b) $\angle\tilde{V}_{-}(\omega, z)$

Figure 5.5: Non-linear phase of the PFD filter

The phase of the PFD filter is mainly caused by the chirping effect. Similar to the magnitude, the phase can easily be calculated from the simulated pulses as:

$$\angle\tilde{V}_{+}(\omega, z) = \angle\left(\frac{X_{+,z}(\omega)}{X_{0,z}(\omega)}\right) + \omega\tau_{+}(z) \tag{5.5}$$

$$\angle\tilde{V}_{-}(\omega, z) = \angle\left(\frac{X_{-,z}(\omega)}{X_{0,z}(\omega)}\right) + \omega\tau_{-}(z) \tag{5.6}$$

These filters are plotted in figure 5.5, again with each line representing one depth. The plots show an almost parabolic modification of the phase.

5.1.1 Secondary effects

Additional effects are introduced as a consequence of the primary effects discussed in the previous section, especially from the shift to lower or higher frequencies. As the attenuation of ultrasound in tissue is frequency dependent[10, ch.4], a shift to higher frequencies will increase the attenuation and lead to a higher loss of energy. A shift to lower frequencies will lower the attenuation and lead to a smaller loss of energy. The power ratios between the simulated pulses have been measured and are plotted in figure 5.6 as a function of depth. The measurements show an increase in power with a positive LF pressure, and a decrease in power with a negative LF pressure, relative to zero LF pressure. This corresponds well with measured shifts in frequency (figure 5.4).

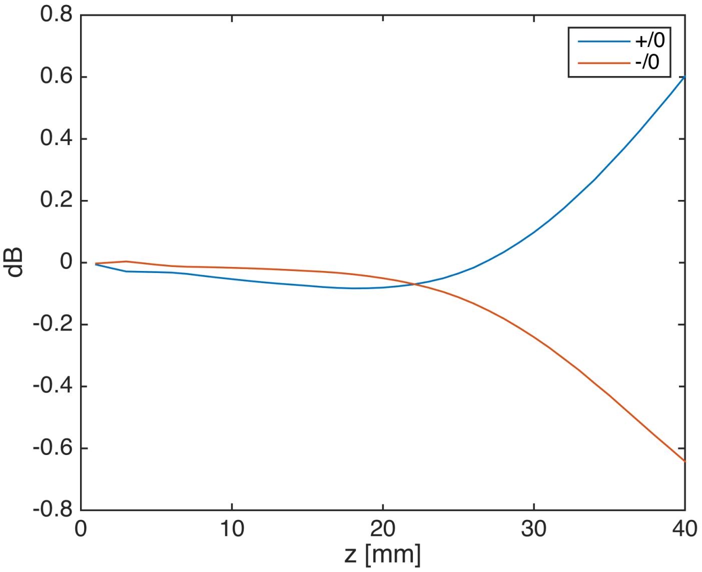


Figure 5.6: Power ratios in dB, X_{+}/X_{0} and X_{-}/X_{0}

Another secondary effect from the shift in frequency is a small change in the beam profile. The beamwidth is proportional to the wavelength[10, ch.6], and inversely proportional

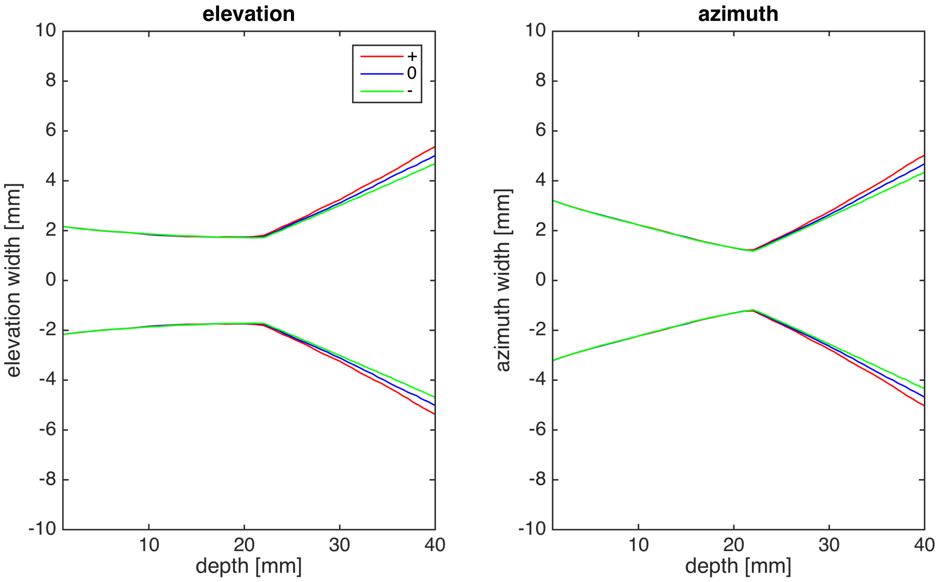


Figure 5.7: 30dB beamwidths of the HF pulses with different LF pressure

to the frequency. Higher frequencies results in narrower beam profiles. This effect has been measured on the simulated 3D pulses during forward propagation. Figure 5.7 shows a contour plot of the 30dB beamwidth in azimuth and elevation direction. The change in beamwidth is most noticeable in the diverging phase, after the focus. At 40mm, the beamwidth with a positive LF pressure is about 1.2mm wider than the beamwidth with a negative LF pressure. This corresponds well with the measurements of the magnitude of the PFD filter, which showed a shift to lower frequencies with a positive LF pressure and a shift to higher frequencies with a negative LF pressure. This change in beamwidth could lead to a small change in the received energy.

5.2 Modelling

Let the simulated pulse from depth z with zero LF pressure be defined in retarded time as $x_0(\tau, z)$, and with positive LF pressure as $x_+(\tau, z)$. A parametric model of $x_+(\tau, z)$, denoted as $\hat{x}_+(\tau, z)$, has been suggested[3] as

$$\hat{x}_+(\tau, z) = x_0\left([1 + a_1(z) + a_2(z)(\tau - \tau_+(z))](\tau - \tau_+(z)), z\right) \tag{5.7}$$

The model is given as a compressed or expanded, chirped and delayed version of the pulse with zero LF pressure. The compression/expansion is given by the parameter a_1, the chirping by the parameter a_2 and the delay by the parameter τ_+. This model has one problem, however. To properly model an expansion or compression, the amplitude should also be changed, otherwise energy would be added or removed. A compression gives a higher amplitude, an expansion gives a lower amplitude. An amplitude factor could also partially model the change in attenuation caused by the frequency shift. An amplitude factor, α, have therefore been added to the model, as

$$\hat{x}_+(\tau, z) = \alpha \cdot x_0 \left([1 + a_1(z) + a_2(z)(\tau - \tau_+(z))](\tau - \tau_+(z)), z\right) \tag{5.8}$$

The model is implemented in Matlab as:

Listing 5.1: Matlab implemantion of the model

```
function xp = nonlin_model(x0, alpha, a1, a2, tau, t)

    t_shifted = (1 + a1 + a2*(t - tau)).*(t - tau);
    xp = alpha.*interp1(t, x0, t_shifted, 'spline', 0)';

end
```

To test how well this model describes the non-linear effects, the following functional has been used:

$$E(\vec{a}) = \frac{\sum_\tau |\hat{x}_+(\tau, z) - x_+(\tau, z)|^2}{\sum_\tau |x_+(\tau, z)|^2} \tag{5.9}$$

where $\vec{a}$ denotes the model parameters, and the sum is over the pulse length. This functional gives the relative error between the model and the simulation.

To find the optimal parameters for the model, a simple gradient descent algorithm has been implemented to find the parameters that minimizes the functional,

$$\vec{a}_{opt} = \min_{\vec{a}} E(\vec{a}) \tag{5.10}$$

The implementation can be found in appendix A.1

5.2.1 Performance on on-axis pulses

The pulse simulation software described in section 3.1 also gives the possibility of extracting the pulses on the transducer axis during forward propagation. The model was first tested on these on-axis pulses. The results are shown in figure 5.8, after optimization of the model parameters. The figure shows the error in the model relative to the simulated pulse, with a comparison of the error when only the delay is accounted for. The difference in the error increases with depth as the PFD accumulates. The reduction in error when using the model with all the parameters is up to 20dB, a massive improvement. It can be concluded that the model is very good for describing the PFD during forward propagation.

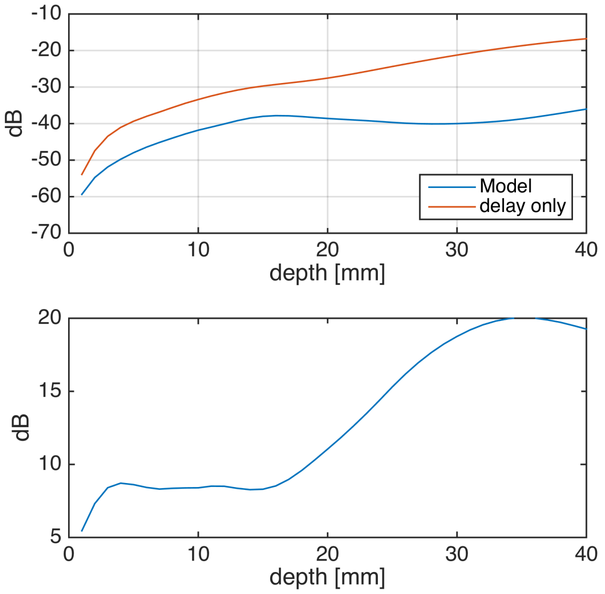


Figure 5.8: Model error on on-axis pulses
The lower figure shows the difference between the plots in the upper figure

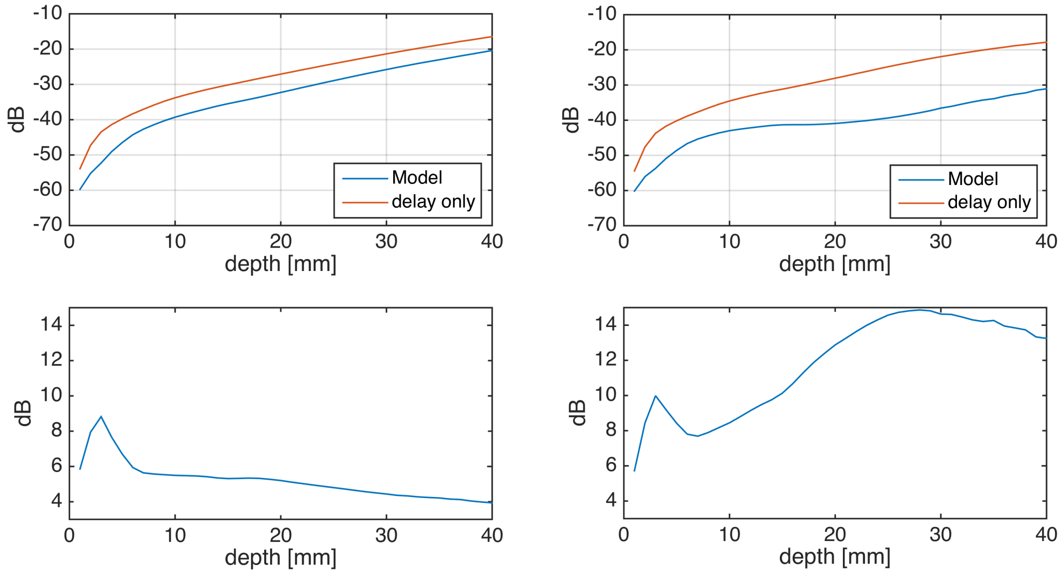
5.2.2 Performance on beamformed pulses

The model has then been tested on the beamformed pulses in the same way. The results from this test are shown in figure 5.9a. While the model still gives a reduction in error compared to only accounting for delay, the improvement is only 4-5dB. This is a massive decrease in performance compared to the results achieved with the on-axis pulses from forward propagation. It was hypothesized that a possible cause for this could be that the beamformed pulses have propagated twice as far, and thus been attenuated twice as much due to absorption. Since the absorption is frequency dependent it results in a gradual shift to lower frequencies. The model may perform worse at these lower frequencies.

As shown in table 3.5, the absorption in the simulation software is given by the following power law:

$$\frac{0.52\text{dB}}{\text{cm}/\text{MHz}} \cdot f^{1.1} \tag{5.11}$$

with f being the frequency in MHz. A filter that compensates for the absorption during back propagation has been applied to the beamformed pulses. The performance of the model when used on the absorption compensated pulses is shown in figure 5.9b. The error between the model and the simulation is now almost the same as with the on-axis pulses, with up to 15dB reduction in error compared to accounting only for the delay. For the best model performance on the beamformed pulses, it can therefore be concluded that it necessary to compensate for the absorption during back propagation.



(a) without absorption compensation

(b) with absorption compensation

Figure 5.9: Model error on beamformed pulses
The lower figures show the difference between the plots in the upper figures

Since the model is giving such good results, it may be used to model the PFD in the speckle filter, f.ex. in equation 2.31. It can also be used as a tool for measuring the different PFD effects. The optimized model parameters can give a way of parametrically quantifying the PFD. This will be used in section 5.3.

5.2.3 Relation between the delay and the chirping

The optimized model parameters can be seen in figure 5.13 for different offsets of the HF pulse on the LF pulse. The use of different offsets will be explained in section 5.3. Of special interest is the likeness between the chirping parameter, a_2, and the delay, τ_+. A relationship between the two have been suggested[3] as:

$$a_2(z) = -\frac{\omega_L^2}{2}\tau_+(z) \tag{5.12}$$

i.e. a_2 is proportional to τ_+ through the center frequency of the LF pulse. It is very interesting if this is true, because then a good delay estimate also gives a good chirping estimate for free. A plot of $a_2(z)$ and $-\frac{\omega_L^2}{2}\tau_+(z)$ is shown in figure 5.10. It can be seen that $-\frac{\omega_L^2}{2}\tau_+$ overestimates a_2 a bit, but that the shape of the curves are quite similar. A slightly adjusted estimate, given as $-0.43\omega_L^2\tau_+$ is also plotted in the figure. It can be seen that this estimate almost perfectly agrees with a_2. This implies that there is indeed a linear relationship between the delay and the chirping, which will make the non-linear phase of the PFD filter much easier to estimate.

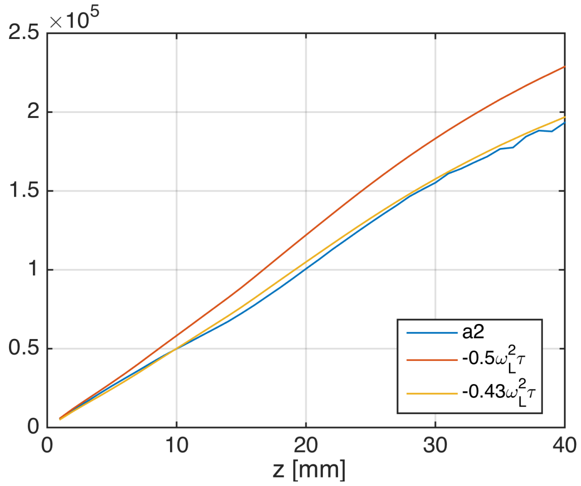


Figure 5.10: Optimal a_2 compared to estimates

Another interesting observation is how linear the development of a_2 is with depth. This also implies a linear development of the non-linear phase of the PFD filter. This in turn may make it possible to extract an average non-linear phase from the speckle filter. This idea will be explored in section 6.2.

5.3 Best pulse positioning for minimum PFD

Changing the initial position of the HF pulse relative to the peak of the LF pulse when transmitting the pulse complex, will also change how the PFD develops with depth. An interesting problem is then to find the optimal initial position that minimizes the PFD, or rather, minimizes the difference in PFD between the pulses with positive and negative LF pressure, and to see how this influences the noise suppression. A hypothesis is that by starting with the HF pulse slightly in front of the LF peak, the compression caused by the positive gradient at this position will cancel out the expansion caused by the negative gradient after the pulse slides to the back of peak, thus resulting in less distortion at larger depths.

To find the answer to these questions, several pulse simulations have been performed with different HF offsets, but all other parameters are the same as given in chapter 3. The tested offsets are in the range $0^o - 16^o$ in front of the peak, with a step size of 4^o, see figure 5.11. The same offsets are used with both the positive and the negative LF pressures.

To measure the PFD, one can look at the parameters obtained with the model described in the previous section, for each of the offsets. The results are shown in figure 5.13. The parameter that is most affected by a change in the offset is a_1, the parameter describing expansion/compression. As expected, with 0^o offset, the a_1 parameter develops in the same direction the whole time. Starting slightly in front of the peak, the a_1 parameter develops first in one direction, and then turns and develops in the opposite direction, resulting in less distortion at larger depths. The smallest difference in the a_1 parameter with positive LF and with negative LF is hence achieved with small offsets for shallow depths, and larger offsets for larger depths.

The amplitude parameter, α , is also affected much by the offset, but there are so many effects that contribute to this parameter that it is difficult to analyze. In terms of having the smallest difference in the α parameter between the positive and negative LF, it seems that the 16^o offset is best, except for the largest depths (>30mm).

The "chirping" parameter, a_2, is minimally affected by the offset except at larger depths, >20mm. There seems to be a bit more distortion the larger the offset is with a positive LF, and a bit less distortion the larger the offset is with a negative LF. The difference with a positive and a negative LF is thus more or less constant.

The effect of the different offsets and PFD developments on noise suppression with DCS processing only, is shown in figure 5.12. This figure shows the SNR gain achieved by using a simple continuous DCS processing (eq. 2.13) on the simulated continuous signals (section 3.2), with the noise delay given as half the 1st order delay. The general trend seems to be that the smaller the offset, the better are the results, except at very large depths, >30mm, where a $4^o - 8^o$ offsets seems to be equally good. An explanation for this can be that the noise at any depth z is composed of pulses that have propagated non-linearly, and thus accumulated PFD, over a distance *less* than z. Minimizing the PFD in the noise at any depth z is thus a question of minimizing the PFD development *before* z. Looking at figure 5.13 it can be seen that the compression/expansion parameter, a_1, which is most affected by different offsets, is smallest in the area 0-20mm with 0^o offset. Even though a larger offset yields less PFD development at larger depths, it is the development at shallower depths that is important for the noise. It can thus be concluded that, at least for the transducer configuration and pulse parameters used in this thesis, starting with the HF pulse as close to the LF peak as possible gives optimal noise suppression. A 0^o offset will be used in the rest of this thesis.

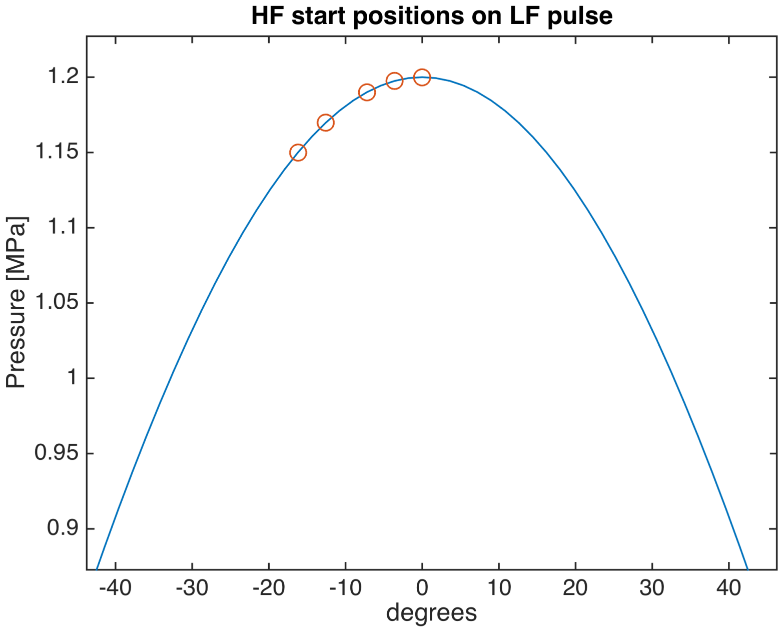


Figure 5.11: Tested start positions of the HF pulse on the LF pulse
Note that the propagation direction is to the left, and the HF pulse "slides" to the right during propagation

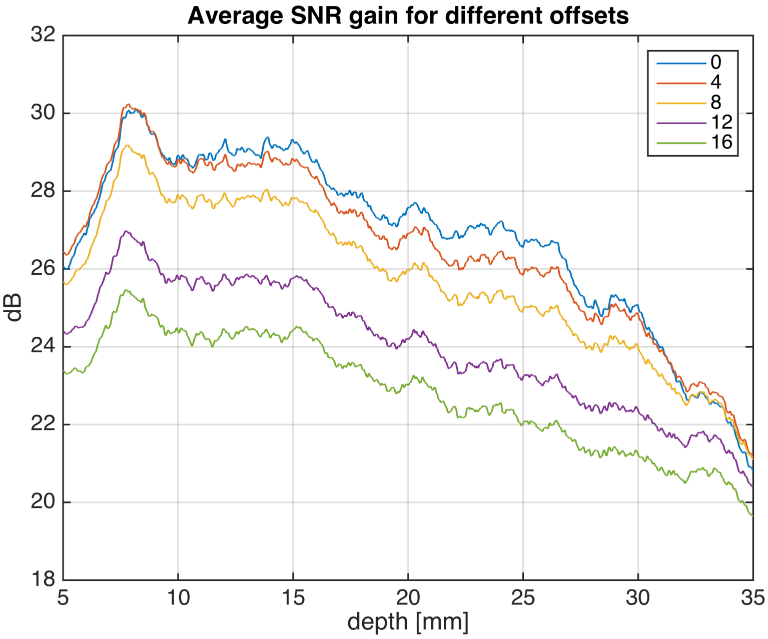


Figure 5.12: SNR gain for different offsets
Continuous DCS processing, with noise delay = half the 1st order delay. Averaged over 30 runs.

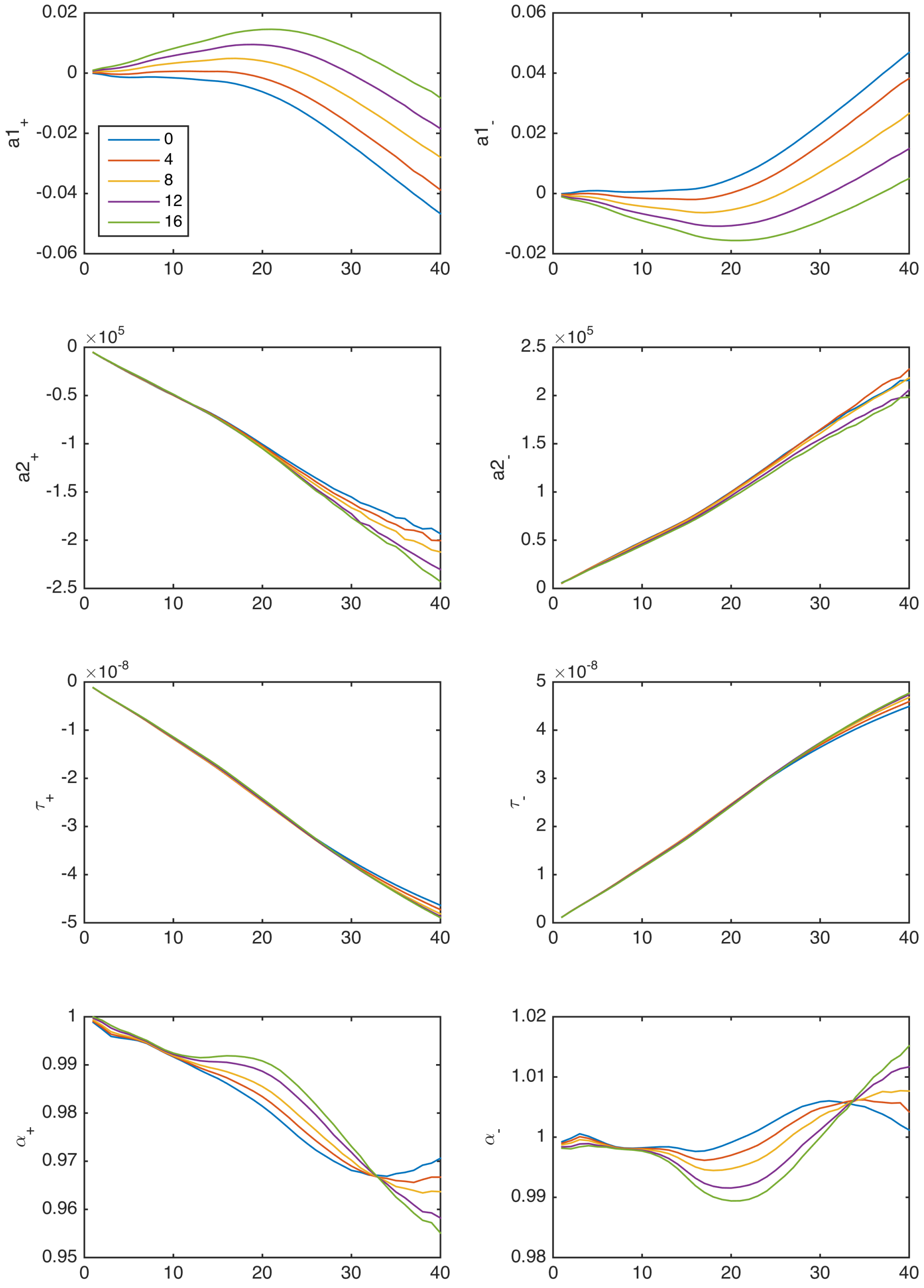


Figure 5.13: Model parameter development with depth for different offsets

Left: with positive LF. Right: with negative LF. The x-axis shows depth in mm.

Chapter 6

Speckle correction

This chapter will present the different methods that have been tried for estimation of the speckle correction filter, and evaluate the achieved performance.

6.1 Solving the equation set

Signal model 2.12 is repeated here for convenience:

$$Y_{+,i}(\omega) = e^{-i\omega\tau_{+,i}}\tilde{V}_{+,i}(\omega)X_i(\omega) + e^{-i\omega\tau_{n+,i}}\tilde{L}_{+,i}(\omega)N_i(\omega) \tag{6.1}$$

$$Y_{0,i}(\omega) = X_i(\omega) + N_i(\omega)$$

$$Y_{-,i}(\omega) = e^{-i\omega\tau_{-,i}}\tilde{V}_{-,i}(\omega)X_i(\omega) + e^{-i\omega\tau_{n-,i}}\tilde{L}_{-,i}(\omega)N_i(\omega)$$

To simplify the notation, the linear phase can be merged into the speckle filter:

$$V_{\pm,i}(\omega) = e^{-i\omega\tau_{\pm,i}}\tilde{V}_{\pm,i}(\omega) \tag{6.2}$$

$$L_{\pm,i}(\omega) = e^{-i\omega\tau_{n\pm,i}}\tilde{L}_{\pm,i}(\omega) \tag{6.3}$$

Dropping the ω parameter and the i subscript for further simplicity, the equation set can then be written as:

$$Y_+ = V_+X + L_+N \tag{6.4}$$

$$Y_0 = X + N$$

$$Y_- = V_-X + L_-N$$

The first method that has been tried is simply to solve this equation set for X, and thus circumventing the need for estimating a speckle filter. Initially there are three equations and six unknowns, which is not a solvable equation set. However, if one can find a relation between L_+ and L_-, and V_+ and V_-, it will be reduced to four unknowns. Furthermore, assuming that $V_\pm$ is easier to estimate than $L_\pm$ with other methods, $V_\pm$ can be assumed known. That reduces the equation set to three equations and three unknowns, which is apparently solvable.

An approximate relation between L_+ and L_- have been found experimentally as:

$$L_- \approx L_+^* \tag{6.5}$$

That is, L_- is approximately the complex conjugate of L_+. This can be seen in figure 6.1, which shows an example of L_+ and L_- *after* delay correction in an interval of length 128 samples ($\approx$ 1mm) around the focus (22mm), averaged over 30 signals. The difference in magnitude is quite small, and the phase of L_- is inverted compared to L_+.

An approximate relation between V_+ and V_-, exemplified with the same interval in figure 6.2, is found to be:

$$V_- \approx \frac{1}{V_+} \tag{6.6}$$

It should be noted that when looking on individual signals instead of averaging over several, there can be larger deviations from these relations.

Inserting these relations into equation set 6.4 gives:

$$\begin{aligned} Y_+ &= V_+X + L_+N \\ \\ Y_0 &= X + N \\ \\ Y_- &= \frac{1}{V_+}X + L_+^*N \end{aligned} \tag{6.7}$$

It then seems simple to solve for X with the following combinations:

$$N = Y_0 - X \tag{6.8}$$

$$L_+ = (L_+^*)^* \tag{6.9}$$

$$\frac{Y_+ - V_+X}{N} = \frac{Y_-^* - (V_+^{-1})^*X^*}{N^*} \tag{6.10}$$

$$\frac{Y_+ - V_+X}{Y_0 - X} = \frac{Y_-^* - (V_+^{-1})^*X^*}{Y_0^* - X^*} \tag{6.11}$$

which reduces to:

$$((V_+^{-1})^* - V_+)|X|^2 + (V_+Y_0^* - Y_-^*)X - ((V_+^{-1})^*Y_0 - Y_+)X^* - (Y_0^*Y_+ - Y_0Y_-^*) = 0 \tag{6.12}$$

simplified as:

$$A|X|^2 + BX - CX^* - D = 0 \tag{6.13}$$

This equation cannot be solved analytically, but if the absolute value of the function has a clearly defined minimum it may be that some kind of search algorithm can find the answer. In order to see if it is possible to find such a minimum under ideal conditions, the equation has been tested on a signal set where Y_0 and Y_+ are as simulated, but Y_- has been created synthetically such that the relations 6.5 and 6.6 are exact. Calculating the functional:

$$J(X) = \left|A|X|^2 + BX - CX^* - D\right| \tag{6.14}$$

for different values of X in a search area around the *true* value of X will show if there is a uniquely defined minimum that can be found by a search algorithm. Each frequency component in each depth interval can be solved individually. Examples of $J(X)$ (in dB) for two different frequency components in the same depth interval, are shown in figure 6.3. The red star denotes the true value of X, and the pink square shows an initial estimate of X given by delay correction only, which can be used as the starting point for a search algorithm. The search area in these figures is $\pm25\%$ of the true value. The function does seem to have a unique minimum corresponding to the real value of X, although it is located within a very narrow "valley", and could be tricky to find.

Before actually implementing a search algorithm, it should be tested if there also exists a minimum in a more realistic case, using the actual simulated signals, and not the synthetic idealized signals. There will then be a deviation from the approximated relations in equation 6.5 and 6.6. If the minimum of $J(X)$ is still located close to the true value of X with these deviations present, this method of solving the equation set can be said to be robust.

Examples of the results from this test are shown in figure 6.4, with the same depth interval and frequency components as in the ideal case. The minimums of the functional are marked with a circle. The exact values of L_+, L_-^*, V_+ and $1/V_-$ in these examples are shown in table 6.1. 6MHz is an example of a small deviation, 9MHz is an example of a large deviation. Unfortunately, in both examples, the minimums are located far away from the true value of X, giving a far worse estimate of X than with only the delay correction. This method seems to be too sensitive to deviations from the ideal relations between L_+ and L_- to be of any use. Since the results do not show much promise, no more time has been spent working on this method.

Table 6.1: Exact values for speckle filter

	L_+	L_-^*	V_+	$1/V_-$
6MHz	$0.921e^{0.023i}$	$0.900e^{0.030i}$	$0.941e^{0.645i}$	$0.964e^{0.646i}$
9MHz	$0.857e^{-0.106i}$	$0.872e^{-0.016}$	$1.008e^{0.744i}$	$1.039e^{0.741i}$

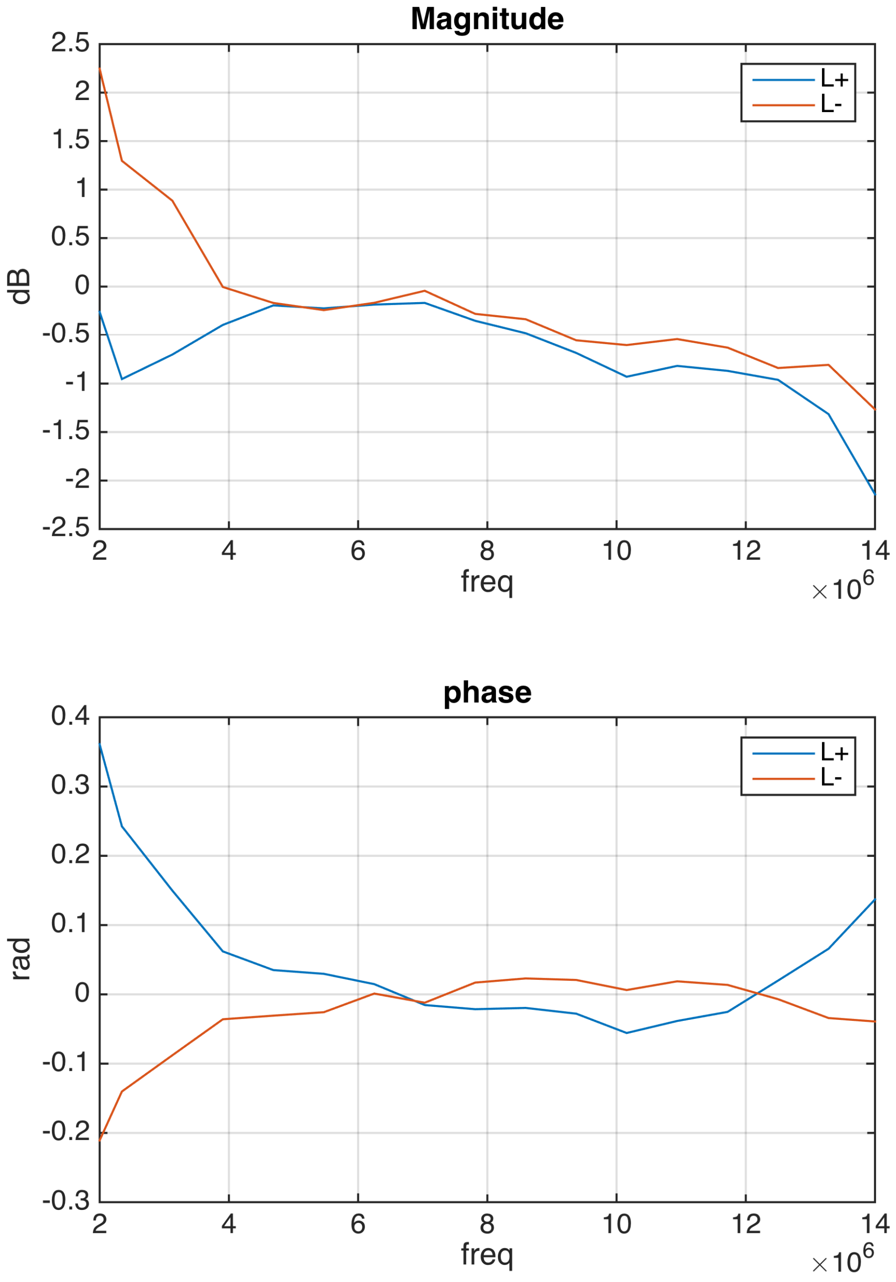


Figure 6.1: Comparison of L_+ and L_-

After average noise delay correction, at a depth of 22mm, interval length 128 samples, weighted with a hanning window, averaged over 30 signals

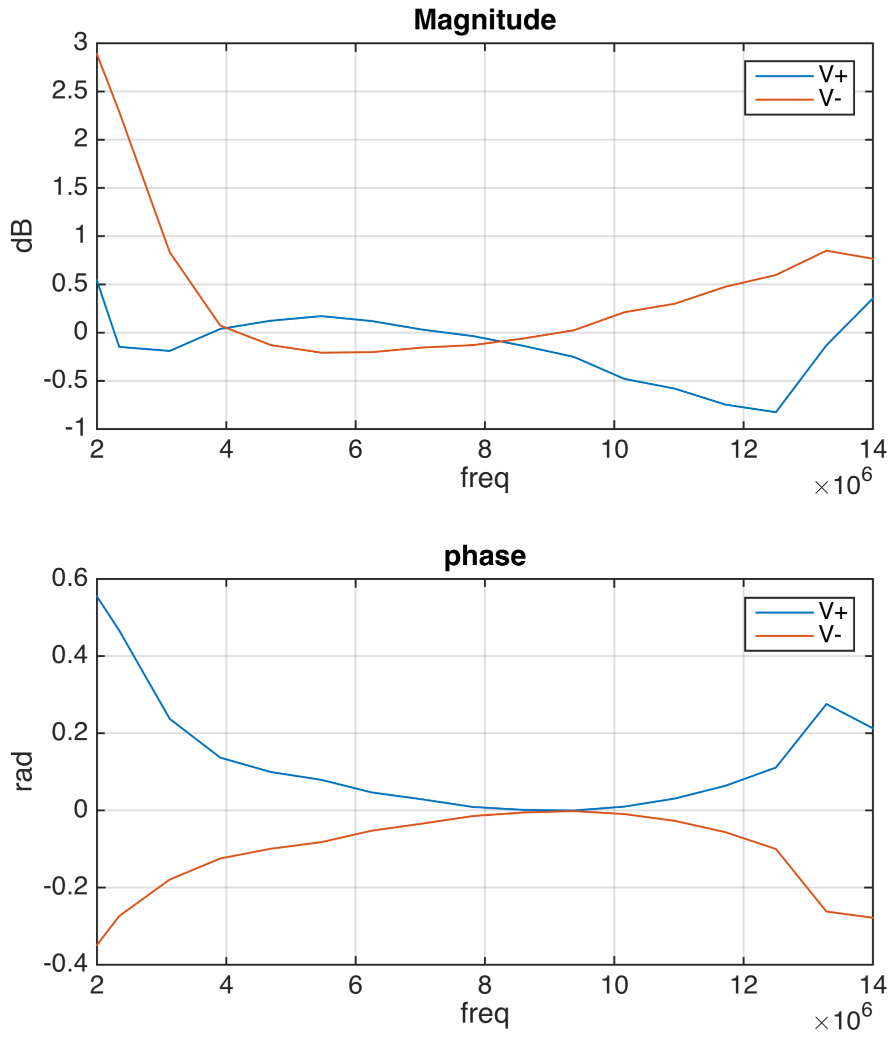
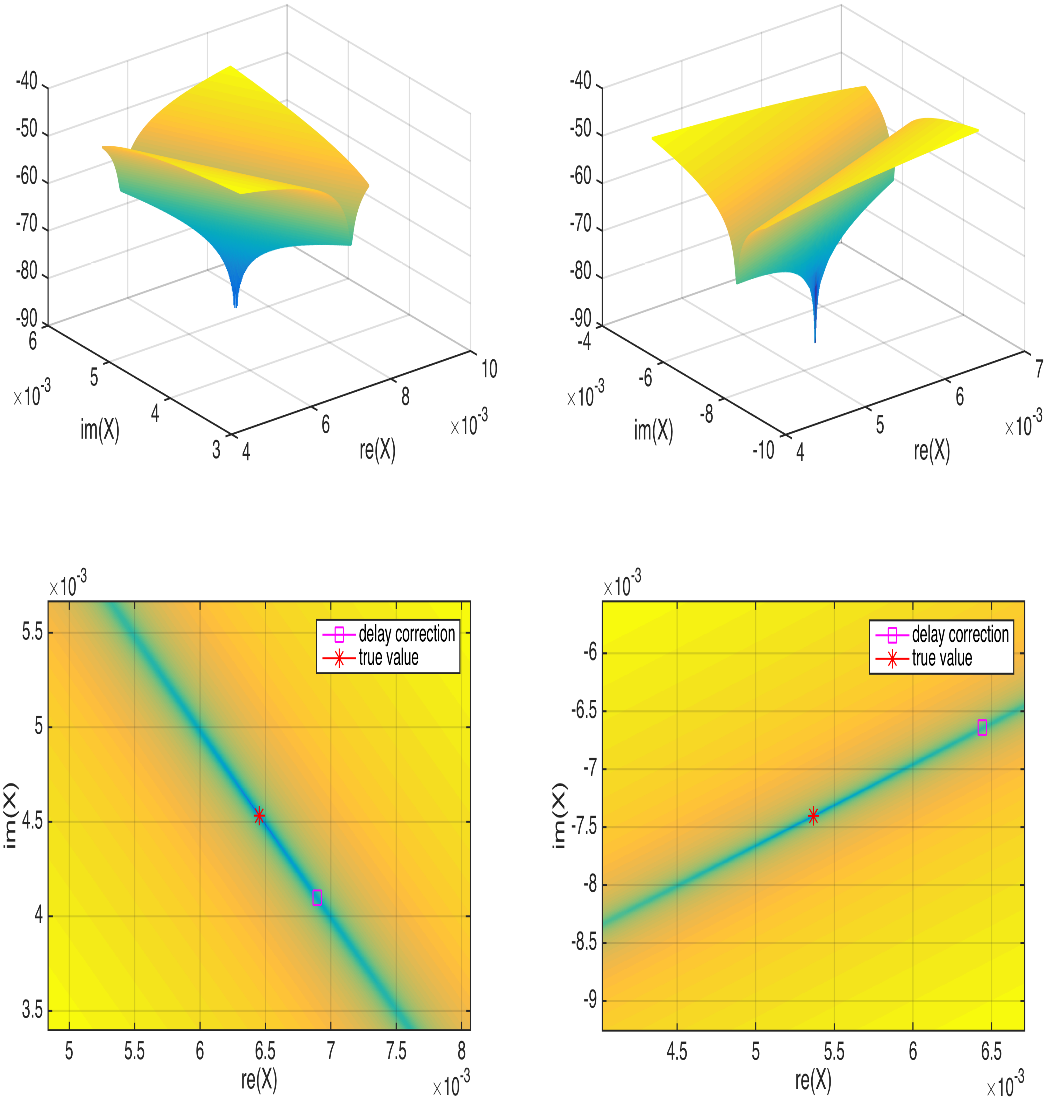


Figure 6.2: Comparison of V_+ and V_-

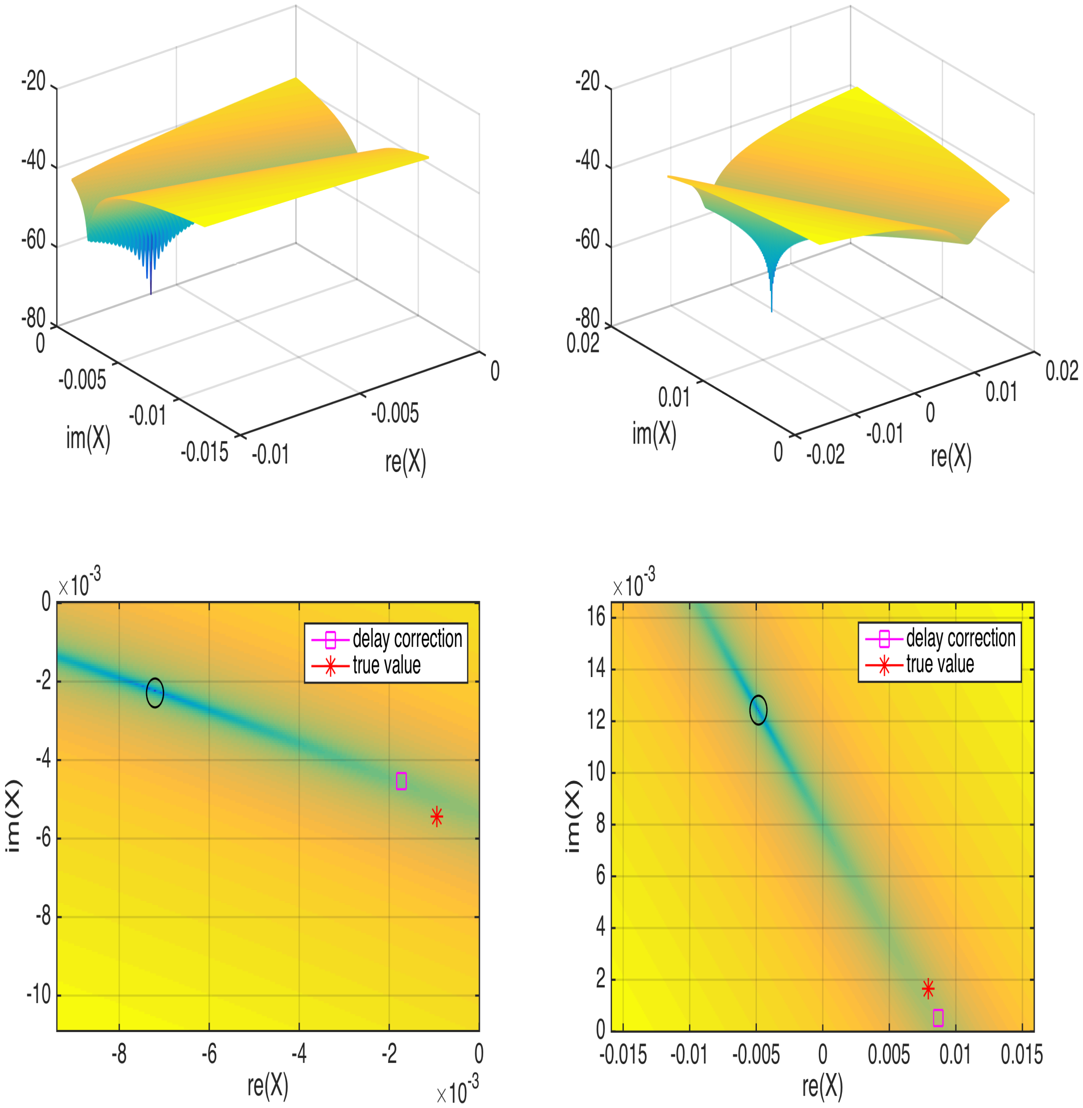
After 1st order delay correction, at a depth of 22mm, interval length 128 samples, weighted with a hanning window, averaged over 30 signals



(a) 6MHz

(b) 9MHz

Figure 6.3: J(X) with ideal signals, 22mm



(a) 6MHz

(b) 9MHz

Figure 6.4: J(X) with realistic signals, 22mm

6.2 Average non-linear phase

In section 5.2 it was observed an almost linear development of the chirping parameter, which gives the non-linear phase of the PFD, and that the parameter is almost proportional to the delay. This observation inspired an idea that just as one corrects with an average noise delay in DCS processing given by $\tau(z)/2$, one can do a partial speckle correction with an average non-linear phase, given by $\tilde{V}_{\pm}(\omega, z)/2$.

Since this method is a pure phase correction, the phase of the speckle filter needs to be an important factor in order to get good results. To check whether it is the magnitude or the phase of the speckle filter that is of most importance, the ideal speckle filters have been calculated and used for DSCS processing. This has been done with the following steps:

- First the signals have been delay corrected with a continuous delay.
- After delay correction, the signals have been divided into intervals. The interval length that has been used is 128 samples. This corresponds to 1 mm. The intervals have then been multiplied with a hanning window.
- An ideal speckle filter has then been calculated directly from each noise interval, as:

  $$\tilde{L}_{\pm,i} = \frac{\tilde{N}_{\pm,i}}{\tilde{N}_{0,i}} \tag{6.15}$$

  where $\tilde{N}_i$ is interval number i of the delay corrected noise signal, in the frequency domain.
- Finally, the magnitude and phase of these ideal filters have been used for DSCS processing.

The resulting SNR gain, averaged over each interval, can be seen in figure 6.5. It can be seen that the phase is definitely the most important factor of the speckle filter. A correction with the phase of the speckle filter gives 5-10dB improvement over the DCS processing. In comparison, a correction with the magnitude of the speckle filter gives only about 2dB improvement. A pure phase correction thus has the potential to give very good results.

A comparison of the phase of the ideal speckle filter, and the PFD filter $\angle\tilde{V}(\omega, z)/2$ from the pulse simulations, is shown in figure 6.6, exemplified with an interval around 20mm depth. Note that the ideal speckle filter has been averaged over all 30 signals. It can be seen that the shape of the curves is quite similar, but that there is a small, approximately linear phase difference between them. This small difference actually causes the speckle correction with $\angle\tilde{V}/2$ to give worse results compared to DCS processing. This is shown in figure 6.7. Between 15mm and 25mm, there is a decrease in SNR gain of almost 2dB compared to DCS processing. This is an area where the difference between the ideal filter and $\angle\tilde{V}/2$ is large. After 25mm however, there is an increase in SNR gain of about 4-5dB. This is an area where the difference between the ideal filter and $\angle\tilde{V}/2$ is smaller.

To compensate for the small linear phase difference, and get improved SNR gain for all depths, $\angle\tilde{V}/2$ can be combined with a fine-tuning of the delay for each interval. The results in figure 6.8 shows the SNR gain achieved after doing a brute force search for the optimal delay to combine with $\angle\tilde{V}/2$. The SNR gain is now 3-5dB for all depths. This implies that this method does have some potential, if the optimal fine-tuning delays can be found in a realistic way. Note that only fine-tuning the delays, and not combining them with $\angle\tilde{V}/2$, does not gives a significant improvement over the reference DCS processing. By only fine-tuning the delays, the improvement is typically less than 1dB. This is also shown in figure 6.8. It is hence a combination of $\angle\tilde{V}/2$ and a fine-tuned delay that gives a significant improvement.

To summarize, this method is dependent on two estimates, the phase of the PFD filter and a fine-tuned delay for each interval. Future work should explore if it is possible to use the relationship between the delay and the chirping parameter in the PFD model (section 5.2.3) to provide a decent estimate of the non-linear phase.

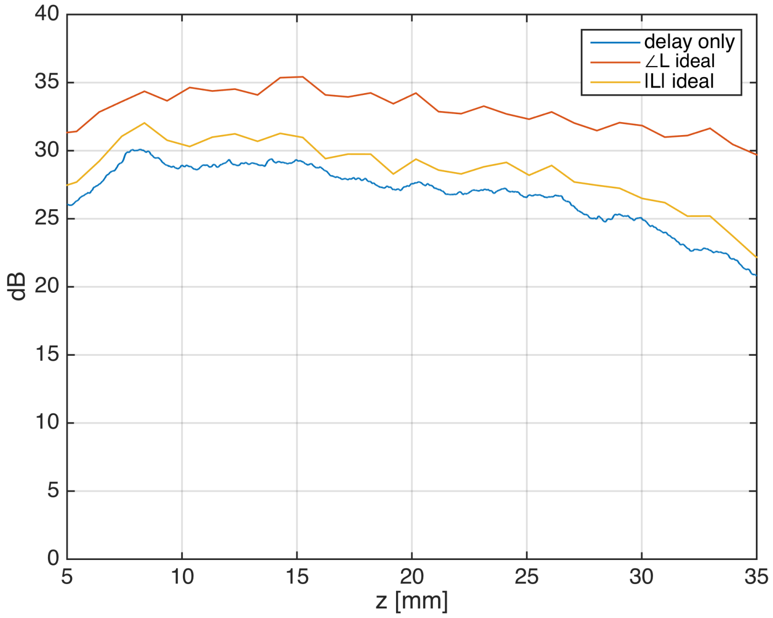


Figure 6.5: SNR gain when correcting for the magnitude and the phase of the ideal speckle filter.

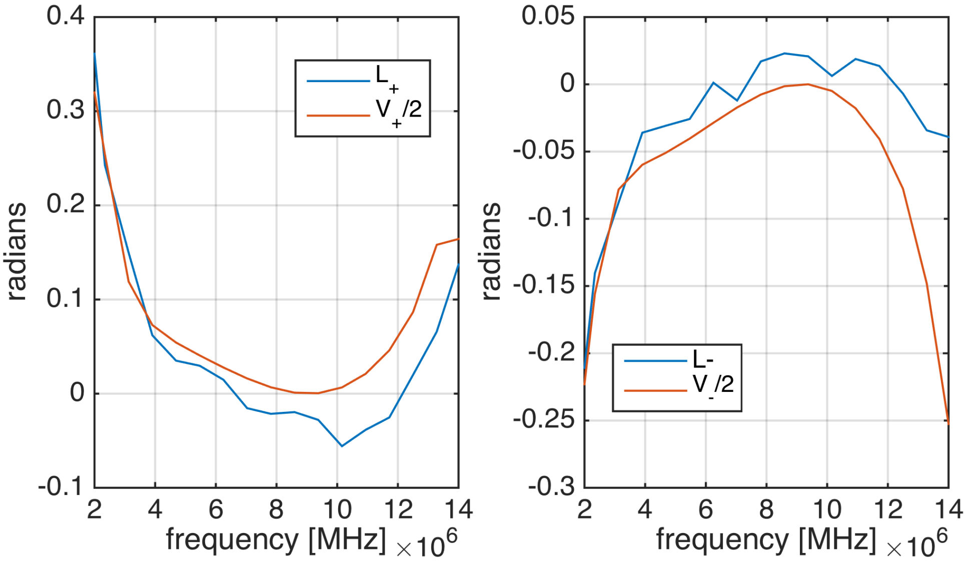


Figure 6.6: Comparison of the phase of the average ideal speckle filter and $\angle\tilde{V}(\omega, z)/2$.
Interval length: 128 sampes, centered at 20mm, weighted with a hanning window

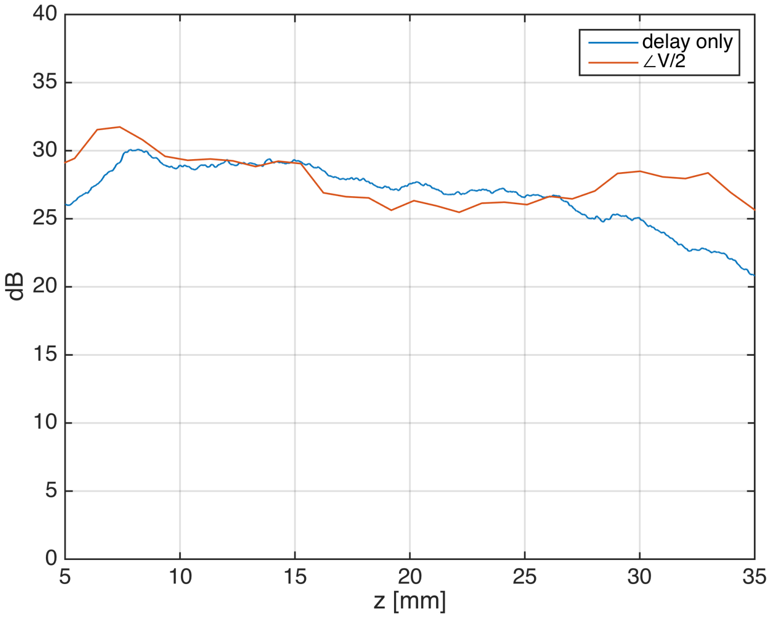


Figure 6.7: SNR gain when correcting for the speckle with $\angle\tilde{V}(\omega, z)/2$.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

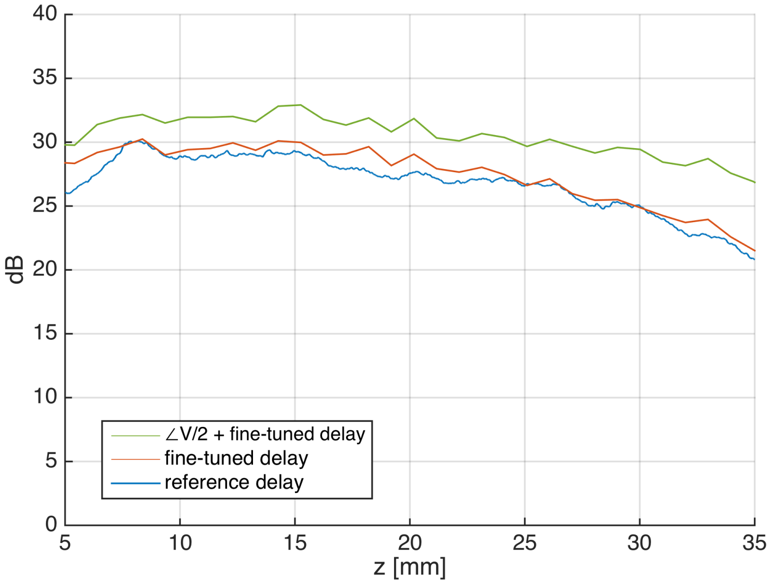


Figure 6.8: SNR gain when combining $\angle\tilde{V}(\omega, z)/2$ with a fine-tuned delay.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

6.3 Physical model

The last method for estimation of the speckle filter that has been tested in this thesis, is an implementation of the physical speckle model given in section 2.3.1, repeated here for convenience:

$$L_{+,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}\left(\alpha(\omega,z_1)e^{-i\omega\tau_+(z_1)}\tilde{V}_+(\omega,z_1)+\alpha(\omega,z_3)e^{-i\omega\tau_+(z_3)}\tilde{V}_+(\omega,z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}\left(\alpha(\omega,z_1)+\alpha(\omega,z_3)\right)} \tag{6.16}$$

$$L_{-,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}\left(\alpha(\omega,z_1)e^{-i\omega\tau_-(z_1)}\tilde{V}_-(\omega,z_1)+\alpha(\omega,z_3)e^{-i\omega\tau_-(z_3)}\tilde{V}_-(\omega,z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)H(\omega,z)e^{-i\omega\frac{2z}{c}}\left(\alpha(\omega,z_1)+\alpha(\omega,z_3)\right)} \tag{6.17}$$

The combined transmit-receive linear beam transfer function, $H(\omega, z)$ has been estimated from a linear 1st order pulse simulation as

$$H(\omega,z) = \frac{X_{l,z}(\omega)}{P_t(\omega)} \tag{6.18}$$

where $X_{l,z}$ is the beamformed 1st order pulse from depth z, and $P_t(\omega)$ is the initial pulse at the transducer.

The non-linear attenuation, $\alpha(\omega, z)$, has been estimated from a non-linear 1st order pulse simulation (with zero LF pressure), as

$$\alpha(\omega,z) = \frac{X_{nl,z}(\omega)}{X_{l,z}(\omega)} \frac{1}{1+\mu\left|\frac{X_{nl,z}(\omega)}{X_{l,z}(\omega)}\right|^2} \tag{6.19}$$

where $X_{nl,z}$ is the beamformed 1st order pulse from the non-linear simulation, and $X_{l,z}$ is the beamformed 1st order pulse from the linear simulation. μ, which is set to 0.01, is a wiener factor that is included because the non-linear simulation introduces new frequency components not present in the linear simulation. The wiener factor thus avoids the extreme amplification at frequencies where $|X_{l,z}(\omega)|$ is small.

The PFD filters, $\tilde{V}_{\pm}(\omega, z)$ has been estimated similar to equation 5.3 and 5.5, and the delay similar to equation 4.1.

To begin with, the exact reflection coefficients, $R(z)$, have been used. This is to test the ideal performance of the model. Estimation of the reflection coefficients is explored in section 6.3.2.

The matlab implementation of this speckle model can be found in appendix A.2 and A.3. In short, the implementation consists of the following steps:

- First, noise model signals are created for the whole imaging depth
- The model signals are then delay correct the same way the simulated signals are
- After delay correction, the model signals and the simulated signals are divided into intervals of equal size, and the model speckle filters are calculated from each interval of the model signals. The interval length that has been used is 128 samples, and each interval is weighted with a hanning window.
- Finally, the model speckle filters are used for speckle correction of the simulated signals

The average SNR gain achieved when using this speckle model with ideal parameters, is shown in figure 6.9. It can be seen that the SNR gain is almost constant for all depths, in to contrast to the DCS processing where the gain decreases with depth. The improvement compared to DCS processing is between 7-14dB depending on depth, resulting in a total gain of 35-37dB. The ideal performance of the model is hence very good.

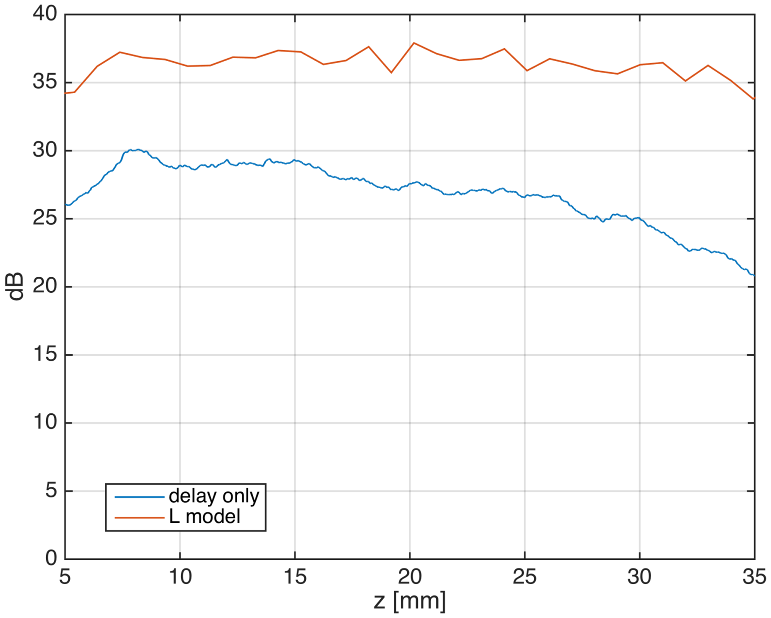


Figure 6.9: SNR gain with the physical speckle model, with ideal parameters.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

6.3.1 Importance of the parameters

There are a lot of parameters in the physical speckle model that needs to be estimated, making it very complicated to use. The model can be greatly simplified if some of the parameters are unimportant, and can be removed from the model without degrading the performance too much. The beam transfer function, $H(\omega, z)$, should be almost constant within an interval as long as it is not too long, since it is only dependent on the total propagation distance. It may therefore be possible to remove it from the model. The non-linear attenuation should be quite small, since the amplitude of the HF pulse used in the simulation is small, and can also maybe be removed. Generally, the non-linear attenuation could be made as small as desirable by using a small enough amplitude.

To test the importance of these two parameters, three speckle models have been constructed, one where $H = 1$, one where $\alpha = 1$, and one where both is set to one. The results of the speckle correction with these models are shown in figure 6.10.

As expected, setting $H = 1$ does not have much impact on the model. The decrease in SNR gain is between 1-2dB in most intervals. The exception is the largest depths, where the decrease in gain is about 4-5dB.

Setting $\alpha = 1$ has a bit more impact. The decrease in SNR gain is depth dependent, as is expected due to an accumulation of non-linear attenuation. There is a steady decrease in gain from about 1dB at shallow depths to about 5dB at the deepest depths. In the future, it should be tested if this performance loss can be avoided by using a lower amplitude on the HF pulse.

Setting both $H = 1$ and $\alpha = 1$ does not change the results much compared to setting only $\alpha = 1$.

Even in the worst case, where both $H = 1$ and $\alpha = 1$, there is still an increase in SNR gain of about 4-7dB compared to DCS processing. A decent speckle model can thus be constructed as:

$$L_{+,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} \left(e^{-i\omega\tau_+(z_1)}\tilde{V}_+(\omega, z_1) + e^{-i\omega\tau_+(z_3)}\tilde{V}_+(\omega, z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}}} \tag{6.20}$$

$$L_{-,i} = \frac{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}} \left(e^{-i\omega\tau_-(z_1)}\tilde{V}_-(\omega, z_1) + e^{-i\omega\tau_-(z_3)}\tilde{V}_-(\omega, z_3)\right)}{\int\limits_{z_{l,i}}^{z_{u,i}} dz \int\limits_{0}^{z/2} dz_1 R(z_1)R(z_3)e^{-i\omega\frac{2z}{c}}} \tag{6.21}$$

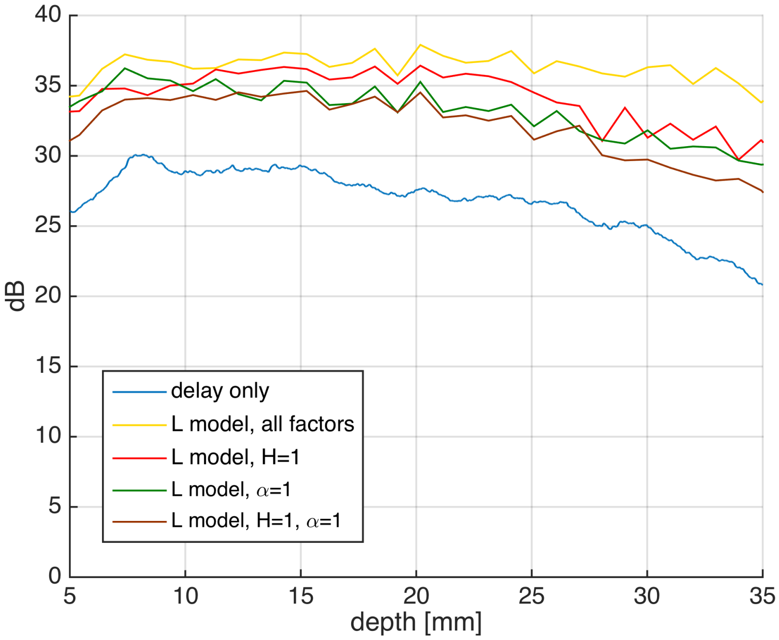


Figure 6.10: SNR gain with the physical speckle model, with parameters removed.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

This means that the problem of estimating the speckle filter is reduced to an estimation of the PFD and an estimation of the reflection coefficients.

6.3.2 Estimation of reflection coefficients

A simple estimation of the reflection coefficients can be done by finding the peaks of the envelope of the received signal. The envelope is found by taking the absolute value of the hilbert transform of the received signal:

$$\text{env}(y) = |\mathcal{H}(y)| \tag{6.22}$$

Both the hilbert transform and a function for finding the peaks of a signal is built-in in Matlab, so this basic estimation method is quite easy to implement, see code listing 6.1. This method has a couple of drawbacks:

- It gives no information about the sign of the reflection coefficient, only the magnitude.
- The attenuation of the received signal will be included in the estimate. This should however not be too much of a problem. Since all the overlapping noise pulses in an interval have propagated approximately the same length, the extra attenuation

factor in the estimate should be more or less a constant. This constant will then be removed in the fraction in equation 6.16.

- Reflection coefficients that are close together cannot be resolved individually, due to the interference pattern of the overlapping pulses.

In addition to these drawbacks, there is a more general problem. Estimation of the reflection coefficients in the received signal will also include false echoes from the multiple scattering noise. An idea to solve this problem is that a DCS processing can be applied to the signal first, which may give sufficient suppression of the noise such that a decent estimate of the reflection coefficients can be made. If this estimate can be used to create a decent speckle filter that gives better suppression of the noise, it may be possible to get an even better estimate of the reflection coefficients. This can in turn be used to create a better speckle filter, and so on. An iterative method that for each iteration gives a better speckle filter can then be used.

To begin with, the reflection coefficients have been estimated directly from the 1st order signal. This gives an impression of the best performance that is possible to achieve. The results from creating a speckle filter with these estimates, where all the other parameters in the model have been ideal, are shown in figure 6.11. The interval size and weighting window is the same as for the ideal case. The results are, unfortunately, not good. There is no improvement in the SNR gain compared to the DSC processing. To see if this is caused by lack of information about the sign of the reflection coefficient, a new set of signals have been created with only positive reflection coefficients. Note that the magnitude of the coefficients are the same, only the sign has been changed. The results from these signals, are shown in figure 6.12. In this case, the speckle filter with estimated reflection coefficients give 2-4dB improvement, compared to DCS processing. These results implies that it is important to account for the sign of the reflection coefficients, and that the simple estimation method presented in this section is not sufficient in a general case.

Listing 6.1: estimation of reflection coefficients in Matlab

```
function [R] = estimate_R(y)

        L = length(y);

        %find envelope of the signal
        y_env = abs(hilbert(y));

        %use built-in function to find values and locations of peaks
        [peaks, locs] = findpeaks(double(y_env));

        R = zeros(L,1);
        R(locs) = peaks;
end
```

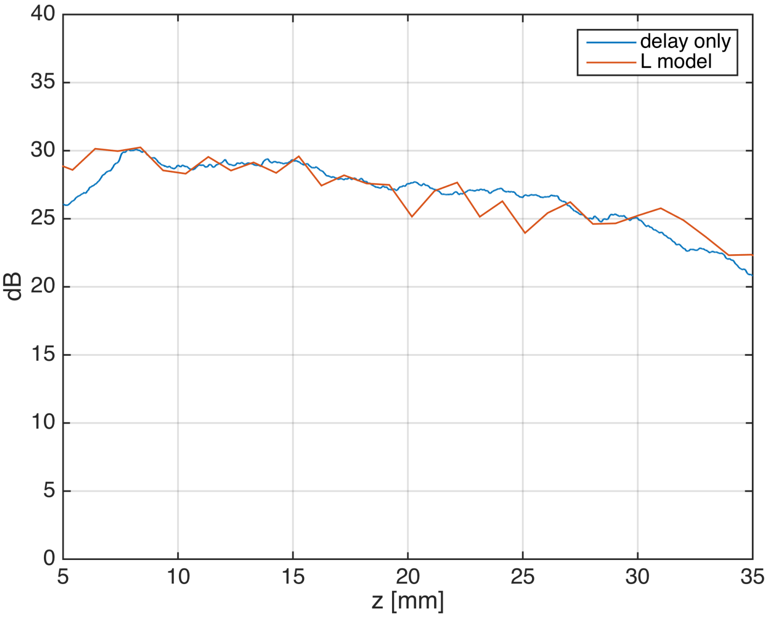


Figure 6.11: SNR gain with physical speckle model, with estimated reflection coefficients.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

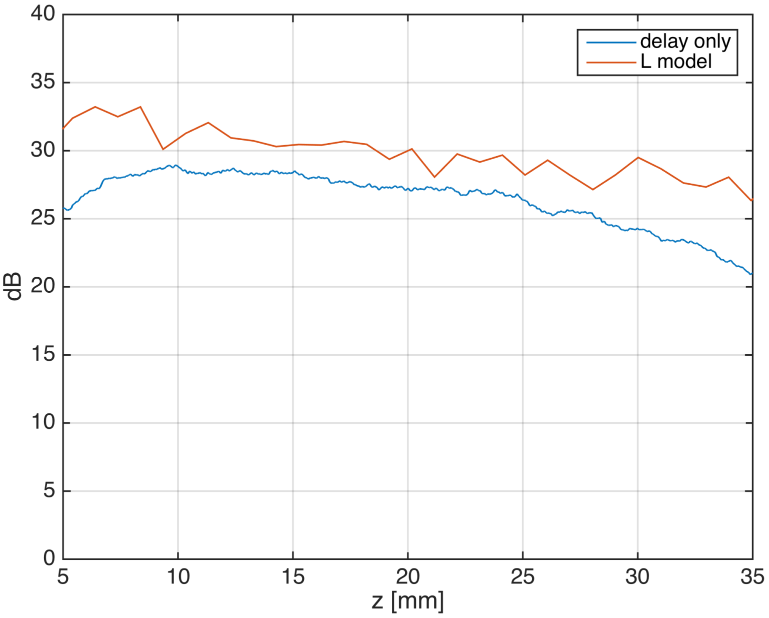


Figure 6.12: SNR gain with physical speckle model, with only positive reflection coefficients.
Interval length: 128 samples, weighted with a hanning window. Results are averaged over 30 signals.

6.4 Discussion

Of the three methods that have been tested in this chapter, two of them are showing promise, the average non-linear phase, and the physical speckle model. Of these two, the physical speckle model gives the best results in the ideal case where the perfect parameters are used, and also when some of the parameters are ignored. The performance decreases dramatically however, when the estimated reflection coefficients are used instead of the ideal ones. Not having information about the signs of the reflection coefficients seems to be especially damaging. Therefore, in order to be useful on a general basis, a method for estimating the signs of the reflection coefficients has to be devised. The physical speckle model is very complex however, and it should be considered whether the achieved SNR gain is worth the extra complexity.

The average non-linear phase method is much simpler, although it yields a bit less SNR gain. This method is especially interesting if the observed relation between the delay and the chirping parameter in the PFD model (section 5.2.3) can be used to get a good estimate of the average non-linear phase. In the subjective opinion of the author, this is the method that seems to be the most promising. It is however, dependent on an approximate linear development of the non-linear phase with depth. The material that has been used in the simulations in this thesis, is homogeneous muscle. In a more realistic scenario, the material will have layers of fat and other tissue types mixed in with the muscle. This may destroy the linear development of the non-linear phase. Future experiments should test how well the average non-linear phase method works in such a case. It may be that in such a case, the physical model will be the better choice.

The results in this thesis do not yield any definitive solutions to the problem of estimating a speckle filter, but do point to some ideas that are worth studying further. It may also be that some other ingenious method not considered in this thesis can prove to be better than the methods presented here. In any case, more work is needed, and concrete suggestions for future work will be given in section 7.1.

Chapter 7

Conclusion

Suppression of multiple scattering noise with SURF imaging can be improved by performing a speckle correction on the received signals with different LF manipulation pressures. Three methods for performing such a speckle correction have been examined in this thesis, and evaluated by measuring the increase in SNR gain compared to traditional methods.

The first method consisted of finding a relation between the speckle change with a positive LF pressure and the speckle change with a negative LF pressure. This led to a reduction in the number of unknowns in the equation set describing the received SURF signals, which allowed the equation set to be solved for the 1st order signal with a search algorithm. Unfortunately, the approximated relations that were found turned out to be too coarse to be of practical use.

The second method was inspired by two observations:

- A phase correction is the most important factor of the speckle correction
- The non-linear phase of the PFD develops almost linearly with depth

These observations led to the average non-linear phase method. Combined with an ideal delay correction, this method was shown to give an increase in SNR gain of about 3-5dB.

The third method utilized a theoretically derived, physical speckle model. This method proved to perform very well when the ideal parameters of the model were used, with an increase in SNR between 7-14dB. It did however suffer a major decrease in performance when the reflection coefficients in the model were estimated. It was shown that a lack of information about the signs of the reflection coefficients led to zero increase in SNR gain.

Considering the complexity of the physical speckle model, and its dependence on ideal parameters, it is concluded that the average non-linear phase method is the most promising method. It should be noted however, that it needs to be tested in more realistic scenarios than what has been done in this thesis.

A model for describing PFD effects has also been examined in this thesis. The model proved to perform well, with the difference between the model and the simulated pulses shown to be up to 15dB smaller compared to a simple model that only accounts for the propagation delay. This model can possibly be used in the speckle correction methods, and it can also be used as a tool for quantifying PFD effects.

7.1 Future work

Future work consists of two parts. The first part is to further study the topic of speckle correction. This includes both the promising speckle estimation methods that have been presented in this thesis, and to develop other methods. Specific tasks are:

- Devising a method for estimating the sign of the reflection coefficients, for use in the physical speckle model.
- Further exploring the relation between the delay and the chirping parameter in the PFD model, and the use of this relation to create a decent estimate of the phase of the PFD filter. This estimate can then be used in the average non-linear phase method.
- Finding a realistic method for fine-tuning the delay to combine with the average non-linear phase method.
- Exploring the performance of both the average non-linear phase method and the physical speckle model in a more complex material.
- Devising new methods for speckle estimation and correction.

The second part, which is especially important if the estimation of a decent speckle filter turns out to be too difficult, is to find the optimal pulse transmit setup that minimizes the need for doing a speckle correction. This involves a setup that as closely as possible creates the ideal conditions for the DCS processing method, as listed in section 2.3.1.

Appendix A

Matlab scripts

A.1 Parameter estimation for PFD model

```
%load data
load full_vorat_offset0_apodhf.mat;

hfp = beamformed.surf - beamformed.lf;
hfn = beamformed.surf_n - beamformed.lf_n;
hf0 = beamformed.hf;

%standard delay estimation
[tp, tn] = estimate_delays(hf0, hfp, hfn, PropCtrl_hf, 2,0);

%compensate for absorbtion
[hfp, hfn, hf0] = absorbtion_compensate(hfp, hfn, hf0, PropCtrl_hf, 0.52, 1.1);

Fs = PropCtrl_hf.Fs;
fc_lf = PropCtrl_lf.fc;
nt = PropCtrl_hf.nw;

%in case an offset is used, define the time axis so that t = 0 at the
%center of the HF pulse
offset = (0*pi/180)/(2*pi*fc_lf);
t = (-nt/2:(nt/2 - 1)).*1/Fs + offset;

%find amplitude and phase shift of lf-pulse
tau_l = zeros(40, 1);
p_l = zeros(40, 1);
for z = 1:40
  [p_l(z), i] = max(onaxis.lf(:,z));
  tau_l(z) = t(i);
end

p_l = [1.2; p_l];
tau_l = [offset; tau_l];

% initial parameter estimation
bp = 0.00153162;
c0 = 1549.9;
wl = 2*pi*fc_lf;
```

```

tau = -cumtrapz(1/c0*bp*p_l.*cos(wl*tau_l)).*1e-3;
a1 = wl*cumtrapz(1/c0*bp*p_l.*sin(wl*tau_l)).*1e-3;
a2 = -0.5*wl^2*tau;

tau = tau(2:end);
a1 = a1(2:end);
a2 = a2(2:end);
alpha = ones(40, 1);

%%

% a2 parameter creates replicas if timeaxis is too long
% therefore use only the central part

N = 128;
I = (512 - N/2):(512 + N/2 - 1);

%pre-allocation
xp = zeros(N, 40);
xp_d = zeros(N, 40);
E = zeros(40,1);

%parameter step size
da1 = 1e-4;
da2 = 1e3;
dtau = 1e-11;
dalpha = 1e-3;

num_it = zeros(40,1);
for z = 1:40

  %with all parameters:
  %gradient descent for optimizing parameters
  %max 100 iterations
  E_old = 1;
  while (num_it(z) < 100 && abs(10*log10(E(z)/E_old)) > 0.01)

    num_it(z) = num_it(z) + 1;
    E_old = E(z);

    xp(:,z) = nonlin_model(hf0(I,z), a1(z), -a2(z), tau(z), alpha(z), t(I));
    E(z) = sum((hfp(I,z) - xp(:,z)).^2)./sum(hfp(I,z).^2);

    %small change in parameters
    xp_a1 = nonlin_model(hf0(I,z), a1(z) + da1, -a2(z), tau(z), ...
      alpha(z), t(I));
    xp_a2 = nonlin_model(hf0(I,z), a1(z), -(a2(z) + da2), tau(z), ...
      alpha(z), t(I));
    xp_tau = nonlin_model(hf0(I,z), a1(z), -a2(z), tau(z) + dtau, ...
      alpha(z), t(I));
    xp_alpha = nonlin_model(hf0(I,z), a1(z), -a2(z), tau(z), ...
      alpha(z) + dalpha, t(I));

    %calculate gradient
    dE_a1 = (sum((hfp(I,z) - xp_a1).^2)./sum(hfp(I,z).^2) - E(z));
    dE_a2 = (sum((hfp(I,z) - xp_a2).^2)./sum(hfp(I,z).^2) - E(z));
    dE_tau = (sum((hfp(I,z) - xp_tau).^2)./sum(hfp(I,z).^2) - E(z));
    dE_alpha = (sum((hfp(I,z) - xp_alpha).^2)./sum(hfp(I,z).^2) - E(z));

    %step size (experimental, seems to work)
    h = 5e5;

    %update parameters
    a1(z) = a1(z) - dE_a1*da1*h;
    a2(z) = a2(z) - dE_a2*da2*h;
    tau(z) = tau(z) - dE_tau*dtau*h;
    alpha(z) = alpha(z) - dE_alpha*dalpha*h;
  end

  %use parameters for current depth as start point for next depth?
  if (z < 40)
    % a1(z+1) = a1(z);
    % a2(z+1) = a2(z);
    % tau(z+1) = tau(z);
    alpha(z+1) = alpha(z);
  end
end
```

A.2 Generation of noise model signals

```
%% find linear transfer function

%load linear simulation data
load linear_hfonly.mat;

N = 256;
I = (-N/2:(N/2 - 1)) + 512;

hf0_lin = beamformed.hf(I,:);
HF0_lin = fft(fftshift(hf0_lin, 1), [], 1);

%initial pulse at the trasnducer
P0 = fft(fftshift(p0(I,65,65)), [], 1);
Rp = 0.1;   %reflection coefficient in the simulations

H = (HF0_lin./ Rp)./repmat(P0, [1,40]);
H(isnan(H)) = 1;
H(isinf(H)) = 1;

%% find non-linear effects

%load non-linear simulation data
load full_vorat_offset0_apodhf;
hf0_nonlin = beamformed.hf(I,:);  %0LF
hfp_nonlin = beamformed.surf(I,:) - beamformed.lf(I,:); %+LF
hfn_nonlin = beamformed.surf_n(I,:) - beamformed.lf_n(I,:); %-LF
HF0_nonlin = fft(fftshift(hf0_nonlin, 1), [], 1);
HFP_nonlin = fft(fftshift(hfp_nonlin, 1), [], 1);
HFN_nonlin = fft(fftshift(hfn_nonlin, 1), [], 1);

%non-linear attenuation
H_nl = HF0_nonlin./HF0_lin.* 1./(1 + 0.01*abs(HF0_nonlin./HF0_lin).^2);
H_nl(isnan(H_nl)) = 1;
H_nl(isinf(H_nl)) = 1;

%PFD and delay
Vp = HFP_nonlin./HF0_nonlin;  %delay will be included in V
Vn = HFN_nonlin./HF0_nonlin;
Vp(isnan(Vp)) = 1;
Vn(isnan(Vn)) = 1;
Vp(isinf(Vp)) = 1;
Vn(isinf(Vn)) = 1;

%% interpolate H, H_nl and V for all possible scatter positions

Fs = PropCtrl_hf.Fs;
dz = PropCtrl_hf.dz;
c0 = PropCtrl_hf.c0;
DZ = ceil(2*dz/c0 * Fs);   %samples per mm
L = 40*DZ;  %samples in total signal (40mm)
%z-axis
z = 0:0.001:0.04;
z2 = 0:c0/(2*Fs):((L-1)*c0/(2*Fs));

H = [complex(ones(N, 1)), H];
H_nl = [complex(ones(N, 1)), H_nl];
Vp = [complex(ones(N, 1)), Vp];
Vn = [complex(ones(N, 1)), Vn];

```

```
w = 1:N;

[Z, W] = meshgrid(z,w);
[Z2, W2] = meshgrid(z2,w);

Z2 = double(Z2);

H2 = interp2(Z,W,H,Z2, W2, 'linear', 0);
%H2 = ones(size(H2)); %ignore H?
H_nl_2 = interp2(Z, W, H_nl, Z2, W2, 'linear', 0);
%H_nl_2 = ones(size(H_nl_2)); %ignore non-linear attenuation?
Vp2 = interp2(Z,W,Vp,Z2, W2, 'linear', 0);
Vn2 = interp2(Z,W,Vn,Z2, W2, 'linear', 0);

%% load scatter set

load scatterset_lambda20;
runs = size(R_mat, 3);

%load signal set for estimating R
sigset = load('signalset_lambda20.mat');
runs = size(sigset.y0, 2);

%%

%allocation
n0_mod = zeros(L,runs);
np_mod = zeros(L,runs);
nn_mod = zeros(L,runs);

% loop over number of runs
for k = 1:runs

  %use exact reflection coefficients
  R = squeeze(R_mat(:,2,k));

  %...or estimate R from signal
  %R = estimate_R(sigset.y0(:,k), z2);

  %indices of scatterer positions
  is = find(R~=0);

  %depths of scatterers
  zs = z2(is);

  %number of scatterers
  ns = length(zs);

  %% create noise model

  %loop over all combinations of scatterers
  for i = 1:ns
    for j = 1:ns

      z1 = zs(i);
      z3 = zs(j);

      if (z1 + z3) > 0.04
        continue
      end

      %create the pulses
      N0_mod = R(is(i))*R(is(j))*P0.*H2(:,is(i)+is(j)).*H_nl_2(:,is(i));
      NP_mod = R(is(i))*R(is(j))*P0.*H2(:,is(i)+is(j)).*H_nl_2(:,is(i)).*Vp2(:,is(i));
      NN_mod = R(is(i))*R(is(j))*P0.*H2(:,is(i)+is(j)).*H_nl_2(:,is(i)).*Vn2(:,is(i));

      %find position to place the pulse in the signal
      center = is(i) + is(j) - 1;
      start = max(1, center - N/2);
      stop = min(L, center + N/2 - 1);

      pstart = N/2 - (center - start) + 1;
      pstop = N/2 + (stop - center) + 1;

      %add the pulses to the total signal
      n0_mod(start:stop,k) = n0_mod(start:stop,k) ...
        + fftshift(real(ifft(N0_mod(pstart:pstop))));
      np_mod(start:stop,k) = np_mod(start:stop,k) ...
        + fftshift(real(ifft(NP_mod(pstart:pstop))));
      nn_mod(start:stop,k) = nn_mod(start:stop,k) ...
        + fftshift(real(ifft(NN_mod(pstart:pstop))));
    end
  end
end
```

A.3 Speckle correction with model filters

```
%% load data

% load simulated signals
load signalset_lambda20.mat;

% load noise model signals
load modelsignalset_lambda20.mat;


%% process data

N = 128;  %interval size
win = hanning(N); %weighting window
NSEG = floor(L/N);  %number of segments

SNRG = zeros(NSEG, runs);
Lp_mod = zeros(N,NSEG, runs);
Ln_mod = zeros(N,NSEG, runs);

% use noise delay = tau(z)/2
tp_corr = tp2/2;
tn_corr = tn2/2;

%loop over runs
for i = 1:runs

  %continuous delay correction first
  %1.order
  yp_shifted = interp1(t, yp(:,i), t+tp_corr, 'spline')';
  yn_shifted = interp1(t, yn(:,i), t+tn_corr, 'spline')';

  %noise
  np_shifted = interp1(t, yprev(:,i), t+tp_corr, 'spline')';
  nn_shifted = interp1(t, ynrev(:,i), t+tn_corr, 'spline')';

  %noise model
  np_mod_shifted = interp1(t, np_mod(:,i), t+tp_corr, 'spline')';
  nn_mod_shifted = interp1(t, nn_mod(:,i), t+tn_corr, 'spline')';


  % divide into intervals
  %1.order
  [seg_p, seg_0, seg_n] = ...
    split_segments(yp_shifted, y0(:,i), yn_shifted, N, win);
  %noise
  [seg_p_rev, seg_0_rev, seg_n_rev] = ...
    split_segments(np_shifted, y0rev(:,i), nn_shifted, N, win);
  %noise model
  [seg_p_mod, seg_0_rev_mod, seg_n_mod] = ...
    split_segments(np_mod_shifted, n0_mod(:,i), nn_mod_shifted, N, win);

  %% find model speckle filters
  for seg = 1:NSEG
    Lp_mod(:,seg, i) = fft(seg_p_mod(:,seg))./fft(seg_0_rev_mod(:,seg));
    Ln_mod(:,seg, i) = fft(seg_n_mod(:,seg))./fft(seg_0_rev_mod(:,seg));
  end


  %% speckle correction

  %loop over segments
  for seg = 1:NSEG

    %speckle correction with model filter
    %1.order
    seg_p_corr = real(ifft(fft(seg_p(:,seg)).*(1./Lp_mod(:,seg,i))));
    seg_n_corr = real(ifft(fft(seg_n(:,seg)).*(1./Ln_mod(:,seg,i))));
    %noise
    seg_p_rev_corr = real(ifft(fft(seg_p_rev(:,seg)).*(1./Lp_mod(:,seg,i))));
    seg_n_rev_corr = real(ifft(fft(seg_n_rev(:,seg)).*(1./Ln_mod(:,seg,i))));
    %subtract signals after correction
    corr_sum = seg_p_corr - seg_n_corr;
    corr_rev_sum = seg_p_rev_corr - seg_n_rev_corr;

    %calculate SNR gain, averaged across each interval
    SNR_before = 10*log10(sum(seg_0(:,seg).^2)./sum(seg_0_rev(:,seg).^2));
    SNR_after = 10*log10(sum(corr_sum.^2)./sum(corr_rev_sum.^2));

    SNRG(seg, i) = SNR_after - SNR_before;
  end
end

%average SNR gain over all signals
SNRG_mean= mean(SNRG, 2);
```

Bibliography

[1] Fourier transform properties. http://www.thefouriertransform.com/transform/properties.php. Accessed: 03.06.2015.

[2] *Ultrasound and Carotid Bifurcation Atherosclerosis.* Springer, 2012.

[3] Bjørn A.J. Angelsen. Estimation of 1st order linear and nonlinear scattering. 2015.

[4] Bjørn A.J. Angelsen and Thor Andreas Tangen. Nonlinear imaging with dual band complexes. 2012.

[5] Ole Martin Brende and Bjørn Angelsen. Adaptive noise delay estimation in dual band ultrasound imaging for reverberation suppression. *Journal of the Acoustical Society of America.* Submitted march 2015.

[6] Halvard Høilund-Kaupang. *Models and Methods for Investigation of Reverberations in Nonlinear Ultrasound Imaging.* PhD thesis, NTNU, 2011.

[7] Johannes Kvam. Fast simulation of surf ultrasound transmit pulse complexes using heterogeneous computing platforms. Master's thesis, NTNU, 2012.

[8] Jochen Matthias Rau. *Dual Frequency Band Ultrasound for Suppression of Multiple Scattering.* PhD thesis, NTNU, 2013.

[9] Stian Solberg. Signal processing in surf ultrasound imaging. Specialization project, NTNU, 2014.

[10] Thomas L. Szabo. *Diagnostic ultrasound imaging: Inside Out.* Academic Press, second edition, 2014.