PASSIVE ACOUSTIC IMAGING AND MONITORING USING AMBIENT NOISE

A Thesis
Presented to
The Academic Faculty

by

Shane W. Lani

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2012

PASSIVE ACOUSTIC IMAGING AND MONITORING USING AMBIENT NOISE

Approved by:

Dr. Karim G. Sabra, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. F. Levent Degertekin
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Justin K. Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: October 29, 2012

ACKNOWLEDGEMENTS

The research contained in this document is a culmination of more than two years of work. Throughout that period of time, I have learned a great deal from a number of individuals. These persons have given me advice, help, and support even in their busiest of times.

I would first like to thank my advisor, Dr. Karim Sabra. He has been instrumental in my academic growth as well as endless insights to my research. I know that he is providing me with a toolset that can allow me to achieve my goals in life. I would also like to thank my committee members, Dr. F. Levent Degertekin and Dr. Justin Romberg for presiding on my committee. Dr. Degertekin has also been very helpful with his expertise in micromachined electronics.

I also would like to thank my friends and family. Both Shaun Anderson and Akibi Archer helped show me the ropes of graduate school, along with answers to my endless questions about MATLAB and acoustics. I would like to thank my family for their help and support especially my brother, Ryan, and my dad. Last, but not least, I would like to thank my wife, Jenette. She has not only allowed me to achieve my goals of attending graduate school, but she has gone above and beyond supporting me in my endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

An approximate of the Green's function can be obtained by taking the cross-correlation of ambient noise that has been simultaneously recorded on separate sensors. This method is applied for two experiments, which illustrate the advantages and challenges of this technique. The first experiment is in the ultrasonic regime [5-30] MHz and uses capacitive micromachined ultrasonic transducer arrays to image the near field and compares the passive imaging to the conventional pulse-echo imaging. Both the array and target are immersed in a fluid with the sensors recording the fluid's random thermal-mechanical motion as the ambient noise. The second experiment is a passive ocean monitoring experiment, which uses spatiotemporal filtering to rapidly extract coherent arrivals between two vertical line arrays. In this case the ambient noise in the frequency band [250 1500] Hz is dominated by non-stationary shipping noise. For imaging purposes, the cross-correlation needs to extract the Green's function so that the imaging can be done correctly. While for monitoring purposes, the important feature is the change in arrivals, which corresponds to the environment changing. Results of both experiments are presented along with the advantages of this passive method over the more accepted active methods.

CHAPTER 1

INTRODUCTION

This thesis focuses around the concept of obtaining a signal from random acoustic noise field. A signal is desired information, while noise is typically a distortion of this signal. As such, noise has a reputation of being very negative. It is the static on the radio, the whirling of the HVAC system in an office, or the ruckus that the neighborhood kids make late into the night. However, the concept of noise has been one of the greatest accomplishments in science. Originally conceived by Einstein in 1905, it is present in nearly every engineering field and has had positive impacts. One such example is proving the existence of atoms before there were optics that could see them [8]. Edward Ng summed up the prominent influence of noise with his quote; "One's man's noise is another man's signal" [3].

One excellent example of obtaining a signal from a diffuse noise field is human sight. Human eyes can capture an ambient diffuse field of light while the brain deciphers colors, dimensions, and ranges. While the light spectrum is the preferred information carrier in air, light cannot work very well underwater or in the human body due to absorption effects. Sound is the preferred information carrier in these mediums with technologies such as SONAR and medical ultrasound. The experiments in this thesis will extract signals from acoustic noise fields for imaging and monitoring the propagating medium.

An estimate of the Green's function between two passive sensors can be extracted from a diffuse noise field by extracting the coherent noise that passes through both sensors. By averaging, the coherent portions of noise emerge from the mostly incoherent

noise. In essence, the coherent portions of the noise field become the signals, while the incoherent portions remain noise. This method has been researched heavily in the last decade, and this thesis expands on the current work by applying this method for two experiments [37,39].

When obtaining a coherent signal from ambient noise it is very important to understand the properties of that particular noise field. The first experiment uses ultrasonic arrays immersed in water. At the MHz frequencies, the predominant noise is due to thermal-mechanical random motion of the fluid particles, also known as Brownian motion. The second experiment uses hydrophones in shallow water near a busy harbor. The ocean is filled with many noise mechanisms, such as shipping noise and surface noise. A closer look at each of the noise environments is considered below.

1.2 Brownian Noise

Brownian motion is the random movement of fluid particles. The same motion was originally observed in 1828 with pollen in water and gave rise to the thought that pollen might be living. Brown, a botanist, systematically experimented with variants of pollen that had been dead for years or soaked in spirits to find that there was no special force, and the motion was a resultant of the fluid in which the pollen was suspended [5,6]. In his paper in 1905, Einstein laid the mathematical foundation behind Brownian motion and proved the existence of atoms in a year that had scientists heavily divided on the question of the existence of atoms [12].

One of the main consequences of Brownian motion is the equipartition of energy and that is represented by the following equation:

$$\frac{1}{2}mv^2 = \frac{3}{2}kT \tag{1.1}$$

where m is the mass of the particle, v the velocity, k Boltzmann's constant, and T temperature in Kelvin. This equation shows that the particles have mass and are moving which in turn relates to the concept that these particles can impart forces. A useful equation of the force of the Brownian motion is given by:

$$\overline{F_{B\_NOISE}^2} = 4kTR \tag{1.2}$$

where R is mechanical resistance [16]. Chapter 3 will use ultrasonic transducers capable of measuring this noise field at MHz frequencies.

1.3 Ambient Ocean Noise

The ambient ocean noise can be divided into two categories. The first category is the noise that can be classified as manmade such as shipping or noise from manmade structure such as rigs or coastal factories. The second category encompasses all other natural noises of the ocean and includes surface noise, biological, and even seismic noise. Each noise source has a particular frequency band and it is important when recording ocean acoustic to know which noise source is most prevalent in your frequency band. Infrasound, sound below the hearing range of humans (f<20Hz), in the ocean is predominately due to seismic events or ocean turbulence. Between the frequencies of 20 to 300 Hz manmade noises are predominate. Other possible contributions to this band can be made by biological noise. Even far away from shipping lanes, shipping noise is dominant due to the low attenuation at these low frequencies as shown in Fig. 1.1. Attenuation is so low in this frequency band that sound can travel 1000+ km with the aid of the SOFAR waveguide [27]. However, if recording closer to a shipping lane or even harbor, the shipping noise could dominate the noise from 20-2500Hz [42].

Noise from 500 Hz to 50 kHz is primarily due to ocean surface noise although biological noises are also present. There are a wide combination of noises that include surf noise, bubble noise, and precipitation noise. This noise is dependent on the sea state of the surface. The sea state corresponds to the roughness of the seas on a scale of 1 to 6 with 6 denoting the roughest sea state.

Frequencies above 50 kHz need to consider the thermal-mechanical motion of the water as described in section 1.3. A diagram illustrating the ambient noises in the ocean for frequencies between 1 Hz to 100 kHz is shown as Fig. 1.2 [4]. This figure is a representation of multiple researchers work and was compiled by Wenz [42]. The ocean acoustic work in this thesis focuses on the frequency band of [250-1500] Hz where shipping noise is predominant.

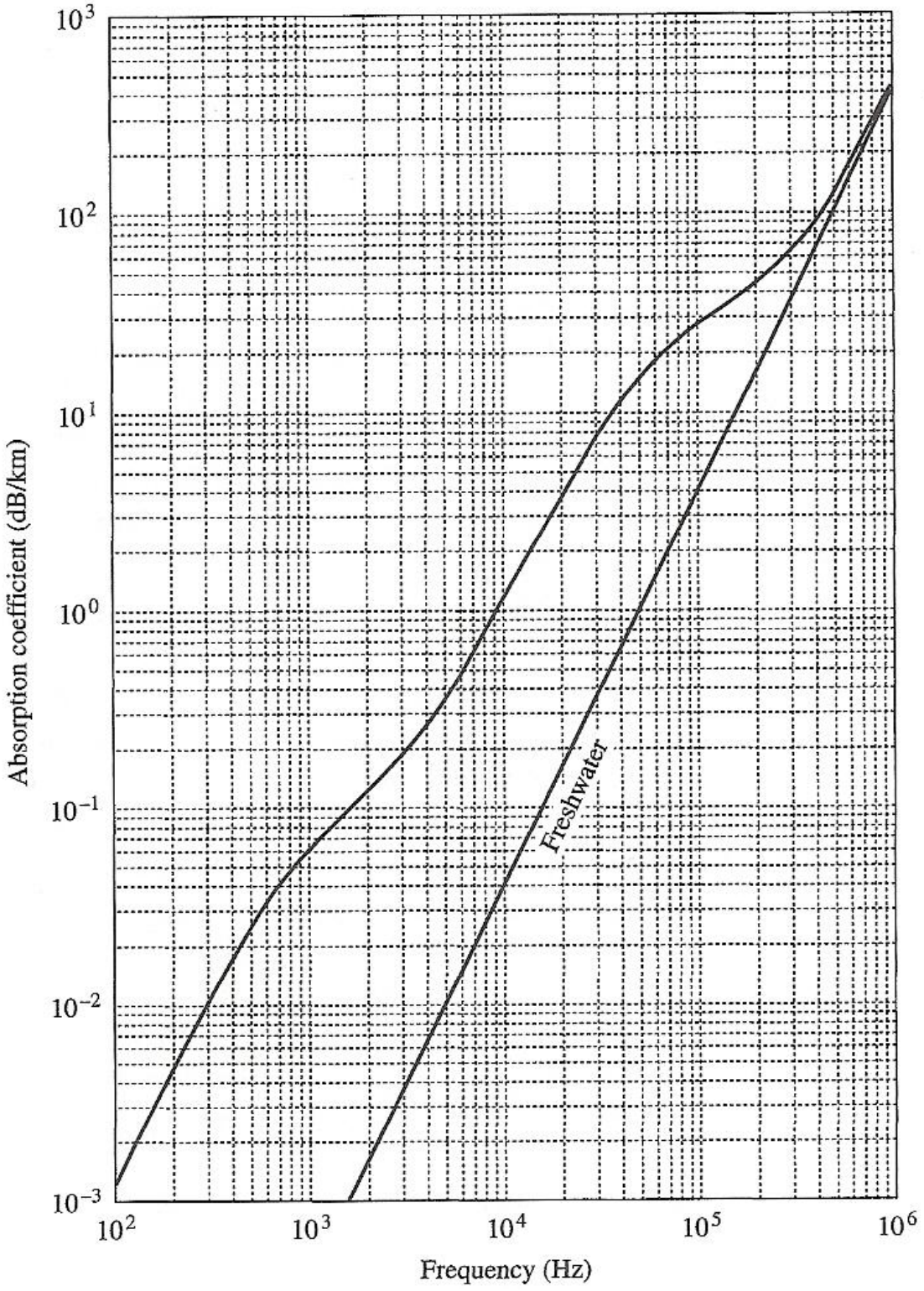

Fig. 1.1 Absorption coefficient for both seawater and freshwater from 100Hz to 1MHz [21]

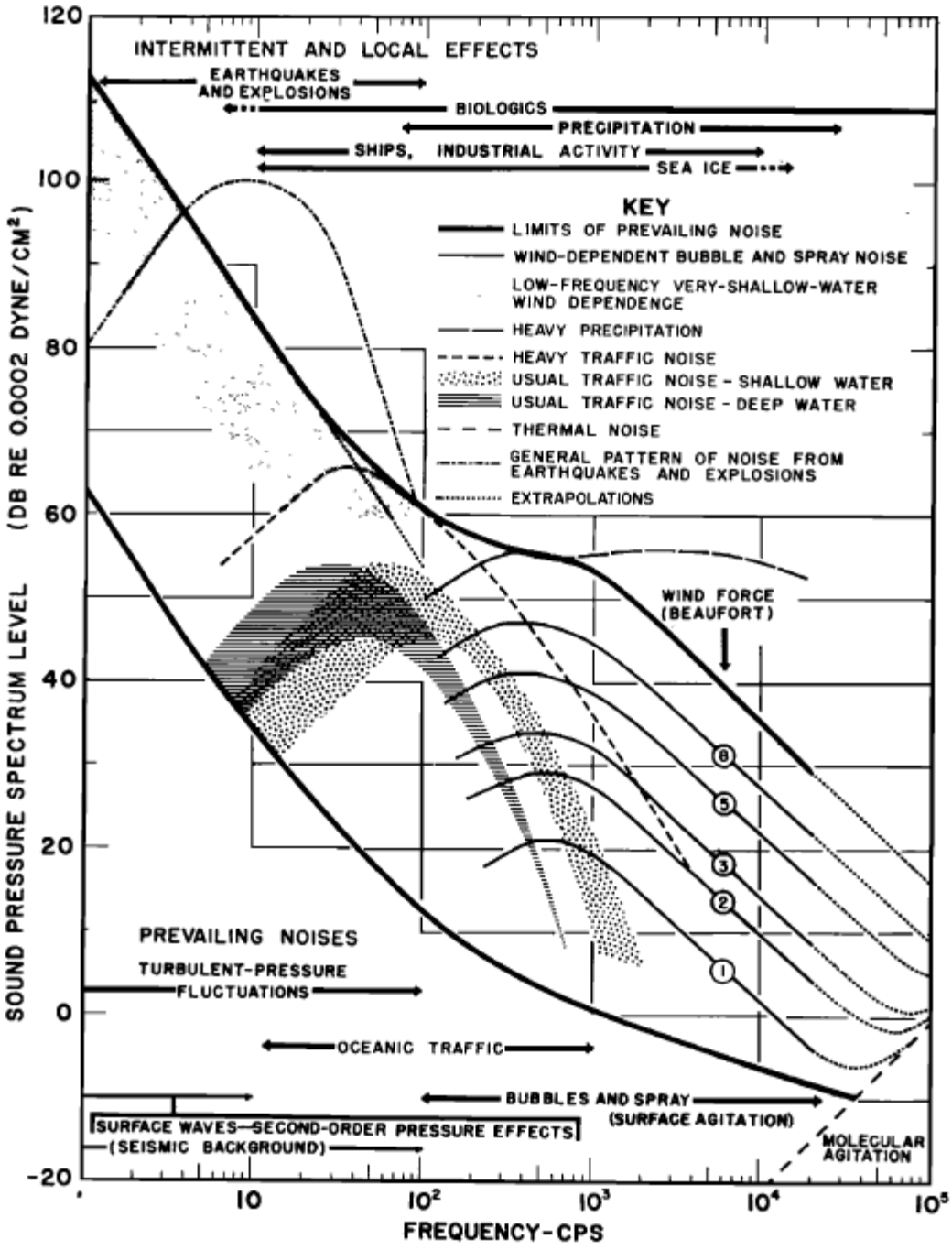

Fig. 1.2 Ambient noises in the ocean from different sources and levels [42]

1.4 Organization of Thesis

Following this Introduction, this thesis has four chapters. The next chapter will give a background of the signal processing methods that are used in chapters 3 and 4. Additional references are provided for the reader who requires a more in-depth understanding of the methodology. Chapter 3 contains an experiment using a capacitive micromachined ultrasonic transducer to perform passive imaging in the MHz regime and draw primarily from two papers published by this author [22,23]. Chapter 4 contains an experiment for passive ocean monitoring on vertical line arrays. Both of these experiments use the cross-correlation of ambient noise. The final chapter is the Conclusion, which summarizes results and discusses possible extensions of this work.

CHAPTER 2

SIGNAL PROCESSING TOOLS

2.1 Introduction

This chapter presents three main signal processing techniques that will be the basis for work in the later chapters. The first is the cross-correlation technique, which is the basis for obtaining a signal from noise by extracting a Green's function approximate from an ambient noise field. The second tool is beamforming. Beamforming is spatiotemporal filtering on data that were sampled in time and space. When the far field approximation is valid, plane wave beamforming is used; yet when the source (or target) is in the near field, an image of the source can be constructed with spherical beamforming. The last tool is time-frequency analysis. These tools when used properly, obtain signals from ambient noise, amplify signal to noise ratio (SNR), construct images, or allow the analysis of the frequency content of a non-stationary signal.

2.2 Obtaining a coherent Signal from ambient Noise

Research on obtaining a signal from random noise was advanced with a paper by Weaver and Lobkis in 2001 [26,40]. The authors demonstrated that the cross-correlation of two sensors recording a diffuse acoustic field will yield an estimate of the Green's function (transfer function) between the two sensors. While not immediately applicable in the ultrasonic regime (due to readily available and well studied sources in the ultrasonic band) in which Weaver and Lobkis undertook their experiments, they foresaw that this method could be readily applied to other fields such as seismology which was done first by Campillo and Paul in 2003 [7]. This concept was also used in underwater

acoustics by Roux in 2004 [28]. In ten years, the research topic had grown significantly, touching the fields of nondestuctive evaluation (NDE), biomedical monitoring, passive imaging, and resulting in the publication of two books [10,32,33,39]. The fundamentals of all of these applications lie with utilizing the cross-correlation.

2.2.1 Cross-Correlation

The normalized cross-correlation, $C_{12}(t)$, between two signals, $S_1(t)$ and $S_2(t)$, is given by the following formula:

$$C_{12}(t) = \frac{\int S_1(\tau) S_2(\tau + t) d\tau}{\sqrt{\int (S_1(\tau))^2 \, d\tau} \sqrt{\int (S_2(\tau))^2 \, d\tau}} \tag{2.1}$$

where the radicals in the denominator are present to normalize with respect to the energy of the signals. The cross-correlation is very similar to the convolution formula as it integrates the first signal multiplied by the second signal with a time shift of *t*. However unlike the convolution, the cross-correlation does not time reverse the second signal. In essence, the cross-correlation examines how similar the signals are when applying a moving time shift, *t*. If the signals are similar at a given time shift, the normalized cross-correlation will return a number very close to 1 for that particular time shift. If the signals are very different for a given time shift, the normalized cross-correlation will return a value close to zero.

Roux showed in 2005 that the derivative of the cross-correlation function would result in the Green's function (in free space) from sensor one to sensor two and vice versa [29]. Roux's result is shown in Eq. (2.2) where the derivative of the cross-correlation is proportional to the time reversed Green's function of the second sensor to the first,

$G(\mathbf{r}_2,0;\mathbf{r}_1,-t)$, and the negative Green's function of sensor one to sensor two, -$G(\mathbf{r}_1,t;\mathbf{r}_2,0)$:

$$\frac{d}{dt}\langle C_{12}(t)\rangle \propto G(\mathbf{r}_2,0;\mathbf{r}_1,-t) - G(\mathbf{r}_1,t;\mathbf{r}_2,0) \tag{2.2}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the positions of sensors one and two respectively with $G(\mathbf{r}_1,t;\mathbf{r}_2,0)$ denoting the proper Green's function between the two sensors (example uses the free space Green's function with no attenuation). These properties of the cross-correlation allow it to obtain a signal from noise. To illustrate the process, a simplified example of the cross-correlation will be considered as shown in Fig. 2.1. The example entails two receivers separated by *x=150 m* in a lossless medium which has a sound speed of *c=1500 m/s*. Plane waves can be impinging on the sensors at any particular angle, θ, which is defined in the Fig. 2.1. The signals recorded by each of the receivers would be identical except for a time delay of $t_d=x*cos(\theta)/c$. So when these signals are cross-correlated, the signal will align at t_d and subsequently the cross-correlation will have a value of 1. Depending on the incoming angle of the source, the time delays can be *+/-(x/c)* if the source is on axis with the sensors (θ=0 or 180 *deg)*, or the time delay can be 0 if the source is broadside to the sensors (θ=90 or 270 *deg)*. Now if plane waves were simultaneously impinging the two receivers from all angles 0-360 (Fig. 2.1), the cross-correlation between the two recorded signals would result in a square pulse. Such a pulse starts from -x/c (from sound at 0 degrees going through sensor one then sensor two) and ends at x/c (from sound at 180 degrees traveling through sensor two then sensor one) as depicted in Fig. 2.2. All arrivals between $-x/c<t_d<x/c$ are resultants from the plane wave at the angles not on axis of the sensors (0 or 180 degrees). The derivative of this cross-correlation functions results in two Dirac delta functions at t=-x/c and t=x/c (the step

discontinuities) as illustrated in Fig. 2.2. These arrivals denote the Green's function (in free space) from sensor one to sensor two and vice versa.

These results so far have assumed an infinite bandwidth. In an experimental setting, this is not very feasible. Hence, a more likely situation is to have a finite bandwidth, which does not contain the direct current (DC) offset that is present in the square pulse result. It is interesting that the finite bandwidth actually gives a good estimate of the Green's function without taking the derivative. Using the same setup as in Fig. 2.1, the incoming plane waves were now band limited, [1000-1200] Hz. The resulting cross-correlation waveform is shown in Fig. 2.3 along with the time derivative of the cross-correlation. Notice both of the waveforms are very similar. Roux showed that the principle difference between the waveforms is a $\pi/2$ phase shift, which means that the cross-correlation of a finite bandwidth is a decent approximation of the Green's function. The cross-correlation may also be a preferred method to estimating the Green's function as taking a numerical time derivative could potentially add unwanted noise. A more detailed theoretical analysis of this problem, both the infinite case and the finite bandwidth case, can be found in Roux's paper [29].

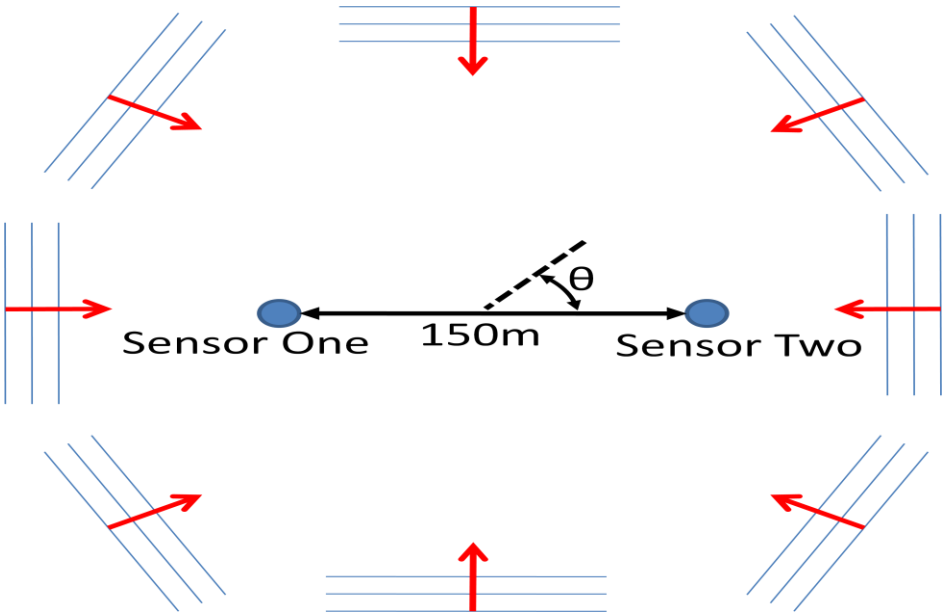

Fig. 2.1 Setup of the cross-correlation example with two sensors spaced as 150 m apart with plane waves impinging from all angles (θ) in a lossless medium.

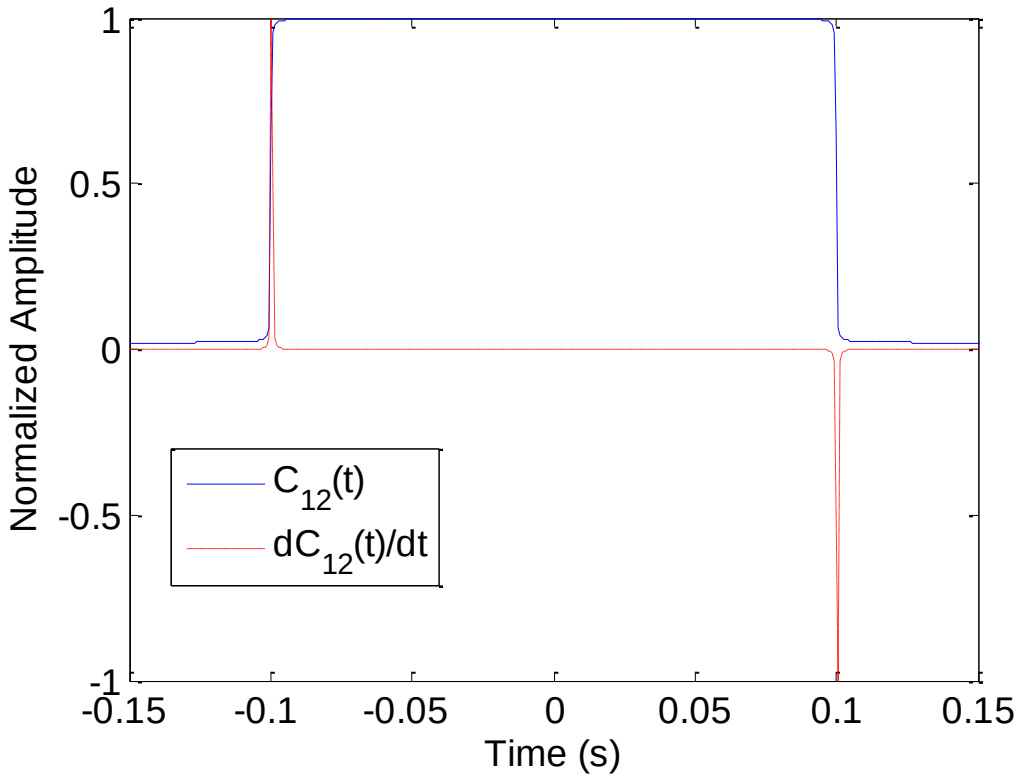

Fig. 2.2 Result of the cross-correlation between the two sensors with infinite bandwidth (blue solid line) and the time derivative of the cross-correlation which is directly proportional to the Green's function between the two sensors (red dashed line)

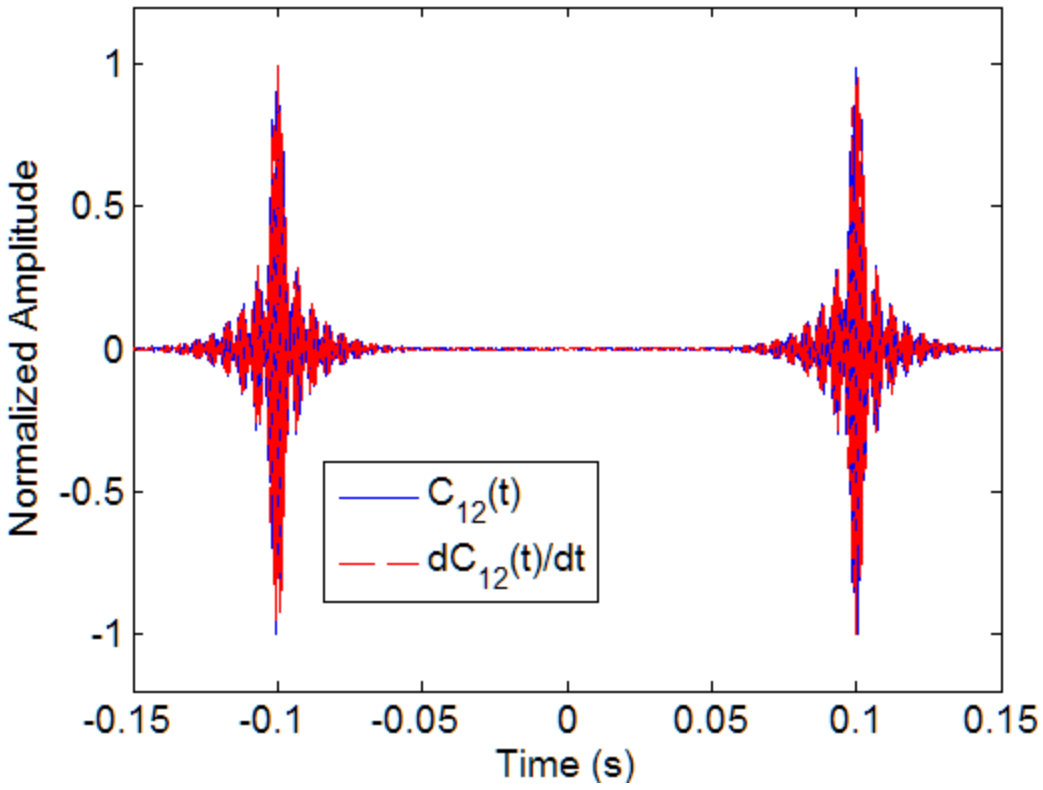

Fig. 2.3 Result of the cross-correlation between the two sensors with a finite bandwidth [1000-1200] Hz (blue solid line) and the time derivative of the cross-correlation (red dashed line). Both waveforms can be used as approximates of the Green's function

2.2.2 Limitations of the Cross-Correlation of Ambient Noise Technique

It must be noted that taking the cross-correlation between two sensors recording simultaneously will not always result in the Green's function estimate between those sensors. This can be perceived when there is only one source present which has the possibility of $t_d=0$ ($\theta=90$). Consequently, the sensors would record the identical signals, which is obviously not the Green's function between the sensors. Hence, there are limitations on obtaining the Green's function from noise using cross-correlation. These next sections will examine the requirements of using the cross-correlation of ambient noise technique and possible methods of mitigating these effects.

2.2.2.1 Large Dominate Source

One problem with extracting the Green's function by using the cross-correlation of ambient noise is when there is one (or multiple) source that is much louder than the ambient diffuse noise. Looking back at the example with sources arriving from all angles, each of the amplitudes of the plane wave was set to one. Fig. 2.4 illustrates how one loud interferer (one particular amplitude is set to 10 or 100 instead of 1) can drastically influence the arrival structure. It is clear that one dominant source will detrimentally affect the emergence of the Green's function estimate. It is possible that the dominate source can help the emergence of the Green's function if the source is in the endfire direction ($\theta=0$ or 180). A further analysis of strong sources and their effect on the emergence of the Green's function in more complex system (ocean waveguide) can be found in another paper by Roux [28].

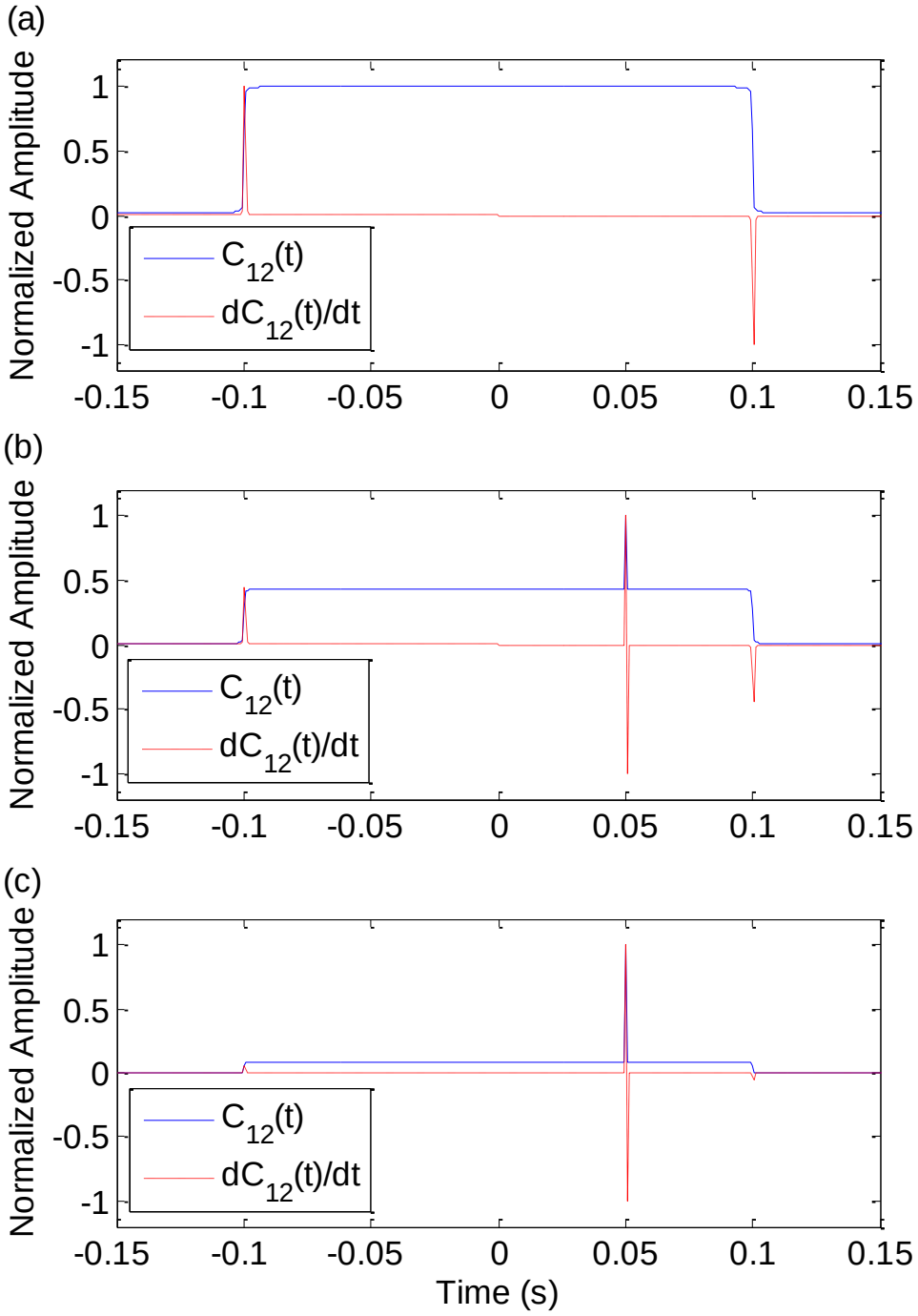

Fig. 2.4 Cross-correlation and time derivative of sources from all angles using an infinite frequency band with one source's amplitude arriving at t_d=.05 (a) equal to all other sources, (b) 10 times as strong as all other sources, or (c) 100 times as strong as all other sources.

2.2.2.2 Low SNR

The SNR is always a serious concern when using the cross-correlation of ambient noise technique. In essence, only a tiny fraction of the noise passing through the sensors adds up coherently. The longer the averaging time the more coherent noise can add up to achieve a better estimate of the Green's function. However, if the sensors are dominated by self noise (electrical noise) and not ambient noise, then the process requires extremely long averaging times. It would also require the environment to remain stable throughout that time duration. The extreme situation is when self noise dominates the sensors, then no Green's function can be extracted due to no coherence between the sensors, as the self noise is independent to each sensor.

2.2.2.3 Limited bandwidth

Another issue is a limited bandwidth, which was already mentioned. Ideally, as shown in the Fig. 2.2, the noise has an infinite frequency band equally weighted. Roux shows in his paper the effects of a limited bandwidth, and an example is shown by limiting the bandwidth of the Fig. 2.2 for three different bands ([1000-1200] Hz, [900-1300] Hz, and [800-1400] Hz) in Fig. 2.5. One effect is that a limited bandwidth can resolve an approximate of the Green's function without taking the derivative of the cross-correlation. The other effect is that the narrower the frequency band (Fig. 2.5(a)) the wider the arrival in the time domain compared to a wider frequency band (Fig. 2.5(c). For imaging and monitoring purposes, it is best to have as wide a band as possible to ensure a clear arrival. It is possible during an experiment that the frequency content of the recorded signals is not evenly weighted across the limited band, which will result in limiting the bandwidth even further. The effects of loud interferers, low SNR, finite frequency bands will need to be as limited as possible to ensure proper a proper Green's

function approximate from the cross-correlation.

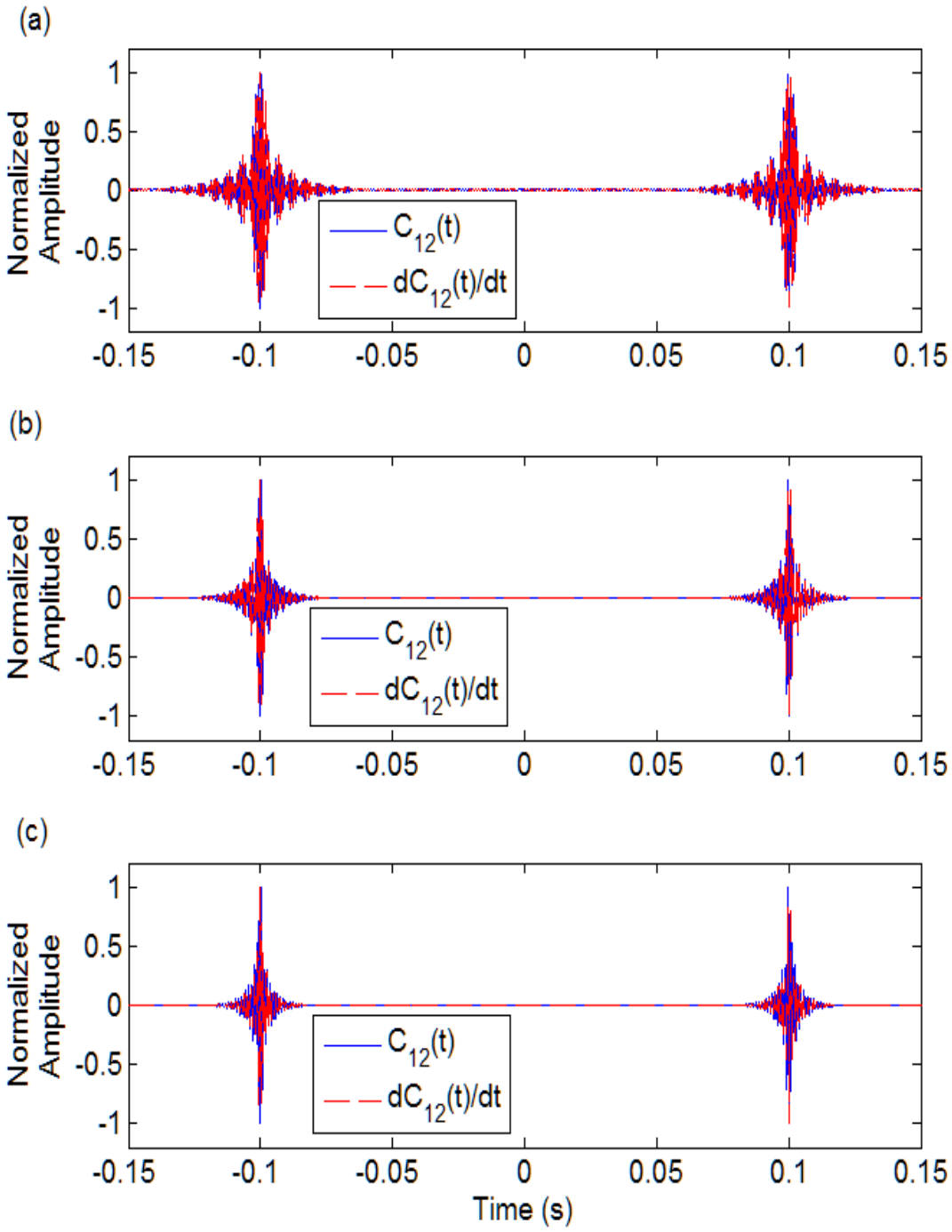

Fig. 2.5 Cross-correlation and time derivative of sources from all angles using a finite frequency band (a) [1000-1200] Hz, (b) [900-1300] Hz, or (c) [800-1400] Hz.

2.2.3 Signal Processing Methods to Mitigate Undesirable Effects

While there are several factors that can severely affect the cross-correlation of ambient noise, there have also been some methods developed that can help to mitigate these effects. These effects are nonlinear and do not have well set parameters. Each data set requires tailoring to extract the best Green's function approximation.

2.2.3.1 Frequency Whitening

Frequency whitening is a method to equalize the amplitude of all frequencies in a desired band. To accomplish this end, a Fourier transform is used, decomposing the signal into an amplitude and phase for each frequency. Frequency whitening sets the amplitudes of the signal to unity while keeping the phase information intact. It is very difficult when extracting signal from noise to select the proper frequency band. The loudest band is not always the best band to utilize especially when frequency whitening enhances the weaker bands so that all frequencies are equal. On the other hand, frequency whitening can also amplify the self noise of the sensors and making the signals less coherent. The best method to select a frequency band is to try numerous frequency bands and select the band that has the best SNR.

2.2.3.2 Amplitude Clipping

Amplitude clipping is used to curb the influence of large outliers in the time domain, just as the frequency whitening curbs the large influence of different strength frequency bands. Amplitude clipping is done by removing any amplitudes of the time signal that exceed 3 standard deviations of the noise. This value can be changed depending on the nature of the noise and environment. Both frequency whitening and amplitude clipping are performed before the cross-correlation is calculated. Another

method to mitigate the effect of loud event is to exclude those time windows in the time window of the cross-correlation.

2.3 Beamforming

Another useful signal processing tool that will be used extensively throughout this thesis is beamforming. Beamforming, is combining signals that were sampled at different spatial locations in a purposeful manner. Since all propagating waves will have some time delay between successive spatially separated sensors, beamforming uses those time delay to shift the recorded signals and add the waveform coherently.

There will be two different types of beamforming used in the thesis. The simplest is plane wave beamforming. This type of beamforming is valid when a spherical source is in the far-field and it can be assumed that plane waves are propagating across the array. Spherical beamforming is beamforming that is done in the near field and needs to take the spherical spreading of the source into consideration. Plane wave beamforming is only sensitive to the angle of an incoming wave, while spherical beamforming contains information about angle and the range of the source and can be used for imaging in the near field. The following subsections will be a brief overview of both types of beamforming, but a more thorough explanation can be found in the book by Johnson and Dudgeon [20]

2.3.1 Plane wave Beamforming

The delay sum beamforming algorithm, z(t), is one of the simplest ways to comprehend beamforming and is given by the following equation:

$$z(t) = \sum_{m=0}^{M-1} w_m y_m \left(t - \Delta_m\right) \tag{2.3}$$

where M is the total number of receivers, w is the weighting for a particular receiver, and $y_m(t-\Delta_m)$ is the time shifted signal that was recorded at each m^{th} sensor. The time shift Δ_m depends on position of the sensors, x_m, speed of traveling wave in the medium, c, and either the search angle or the angle of propagation, ζ, as shown:

$$\Delta_m = \frac{-\vec{\zeta} \cdot \vec{x}_m}{c} \tag{2.4}$$

There are a few different ways that the beamforming algorithm can be used. The first is to "steer" the array to one particular angle. An example of an incoming plane wave at 45 degrees onto a line array of 5 elements can be seen with array setup in Fig. 2.6 and waveform results in Fig. 2.7. The incoming signal to the array is *s(t)* and shown in Fig. 2.7(a). When the beamforming algorithm is not steered directly at 45 degrees, the signals do not add up constructively, Fig. 2.7(b) (broadside search). However, when the beamforming algorithm is steered toward the incoming plane wave, the beamforming output is *M*s(t)* Fig. 2.7(c). This means that the beamforming is amplifying the signal, which is very helpful when working with low SNR signals. Ideally, an array of M sensor will increase the SNR by M times [20].

The second way to use the beamforming is to search all possible angles and record the maximum value of z(t). Using the same incoming wave the search method is now applied and the results can be seen in Fig. 2.8. It is clear that the incoming wave is at approximately 45 degrees from this plot because the value is *M*max[Z(t)]*=25 is at 45 degrees.

Beamforming can also be done in the frequency domain and it is presented as:

$$B(\mathbf{e}, f) = \mathbf{e}(f)'\mathbf{W}\mathbf{Y}(f)\mathbf{Y}(f)'\mathbf{W}'\mathbf{e}(f) \tag{2.5}$$

where **e**(f) is the steering vector given by:

$$\mathbf{e}(f) = \begin{bmatrix} \exp\{-j\vec{k}_0 \cdot \vec{x}_0\} \\ \vdots \\ \exp\{-j\vec{k}_{M-1} \cdot \vec{x}_{M-1}\} \end{bmatrix} \tag{2.6}$$

W are the weights in a square MxM matrix along the diagonal, **Y**(f) is the Fourier transform of the signals, and k_i is the wavenumber equal to $2\pi f/c$. If zero delay beamforming is desired (assuming an incoming wave broadside to a line array) then the equation can be simplified to by having the steering vectors **e**(f) equal to unity and **Y**(f)**Y**(f)' is equivalent to the spatial cross-correlation matrix, **R**(f). This simplified version will be used in Chapter 4.

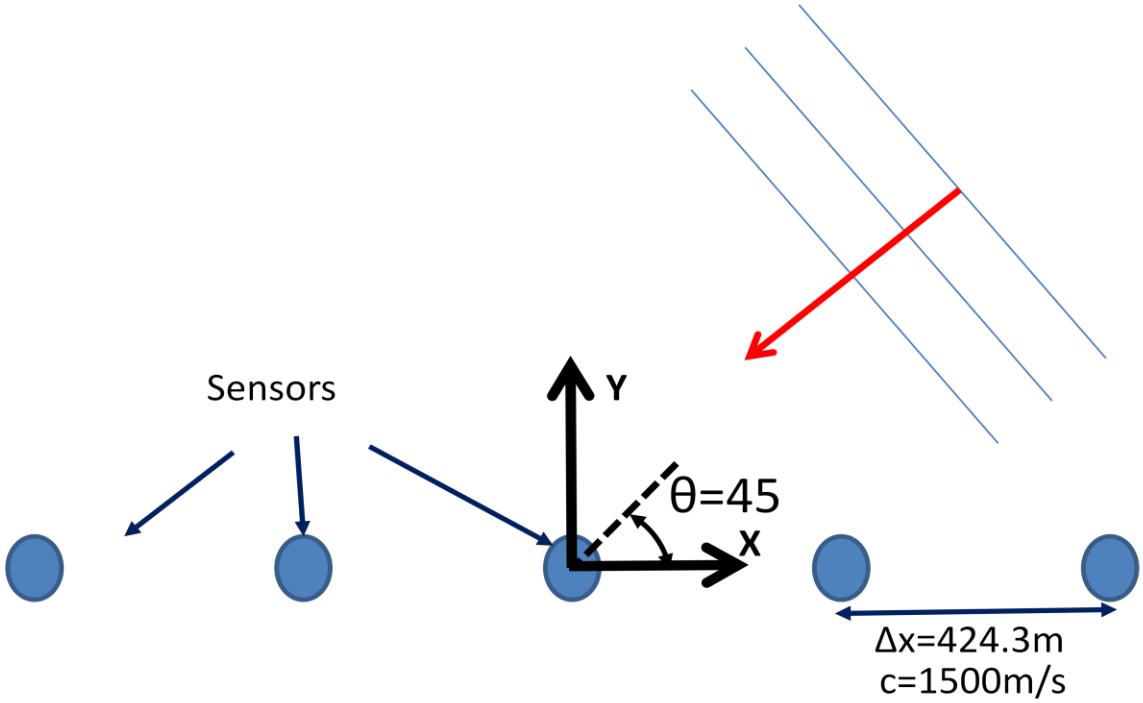

Fig. 2.6 Beamforming setup with M=5 sensors and an incoming plane wave at 45 degrees.

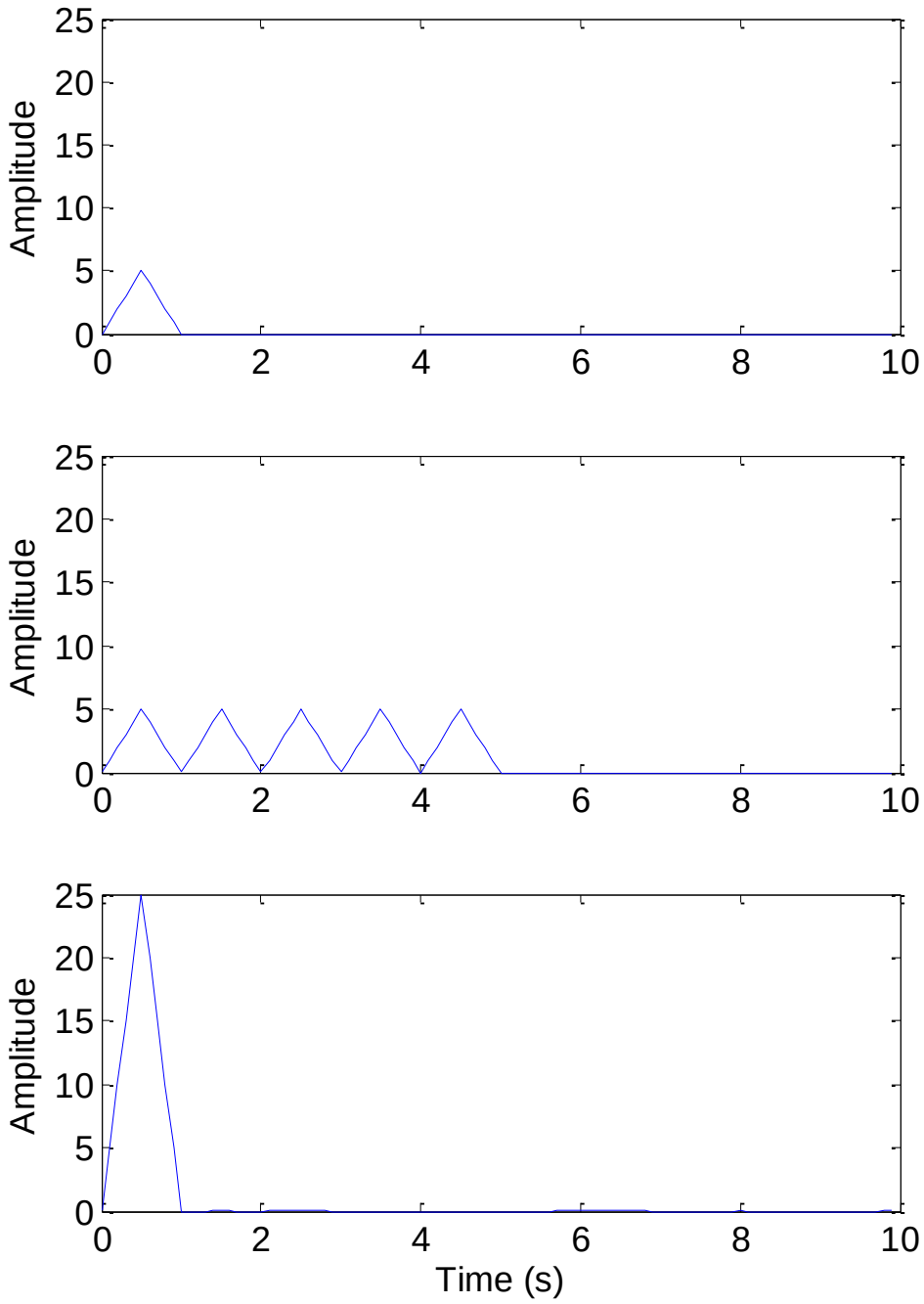

Fig. 2.7 (a) The incoming signal, *s(t)* (b) beamformed signal when using time delay corresponding to θ=0 or a broadside wave (no time delays) (c) beamformed signal in the correct direction resulting in the original signal amplified by the number of sensors (in this case five times)

2.3.2 Spherical Beamforming.

The only difference between plane wave beamforming and spherical beamforming is the assumption of how the wave is propagating. If the wave is in the near field then spherical beamforming has the added advantage of finding not only the direction of the source but also the range due to the curvature of the wavefronts and the amplitude loss associated with a spherical spreading propagation model. Spherical beamforming uses the same delay sum as Eq.(2) but the time delays instead of being determine solely on the direction are now determined by the range:

$$\Delta_m = \frac{-\vec{r} \cdot \vec{r}_m}{c} \tag{2.7}$$

where r_m is the location of the m^{th} sensor and r is the point of interest. Therefore, instead of searching all angles, the spherical beamformer searches all positions and records the maximum of the added signals. As seen by Fig. 2.8, an image is created which highlights where the source is located. This method will be utilized in Chapter 3 for near field imaging.

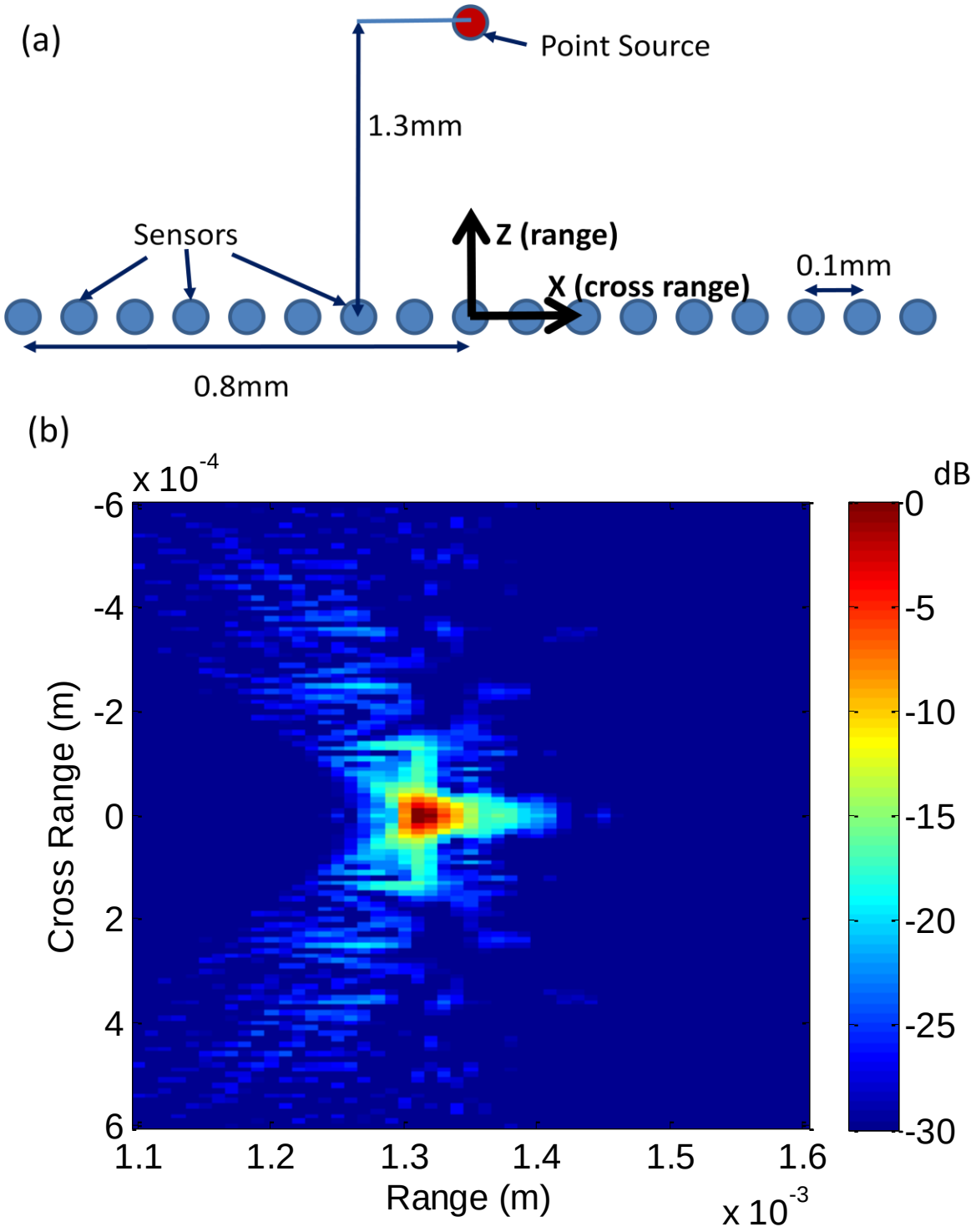

Fig. 2.8 (a) setup of line array and point source for near field beamforming (b) resulting image from spherical beamforming that located the source in both direction and range.

2.4 Time-Frequency

A time-frequency representation of signals can be very informative. In this thesis, two different time-frequency methods were used. This simpler of the two methods is the spectrogram. This method sections the time data into smaller sections and the subsequently takes the Fourier transform. While this method is great for very long data sets, it is not as effective when capturing transient behavior. This is because the smaller the time windows, the coarser the frequency resolution becomes and is known as the time-frequency uncertainty principle [1,38]. Hence, a higher order time-frequency method could be used instead. The higher order time-frequency method chosen was the Smooth Pseudo Wigner-Ville Transform. This method can better localize time-frequency features while eliminating (smoothing out) false features that appear with just the Wigner-Ville Transform by itself. The false features are signal processing artifacts and one can be seen in Fig. 2.9(b), where the middle line is the artifact. Fig. 2.9 illustrates the difference between the spectrogram, Wigner-Ville, and the Pseudo Smoothed Wigner-Ville on two linear chirps. In the thesis, the Smoothed Psedo Wigner-Ville Transform will be used in Chapter 3while the spectrogram will be used in Chapter 4.

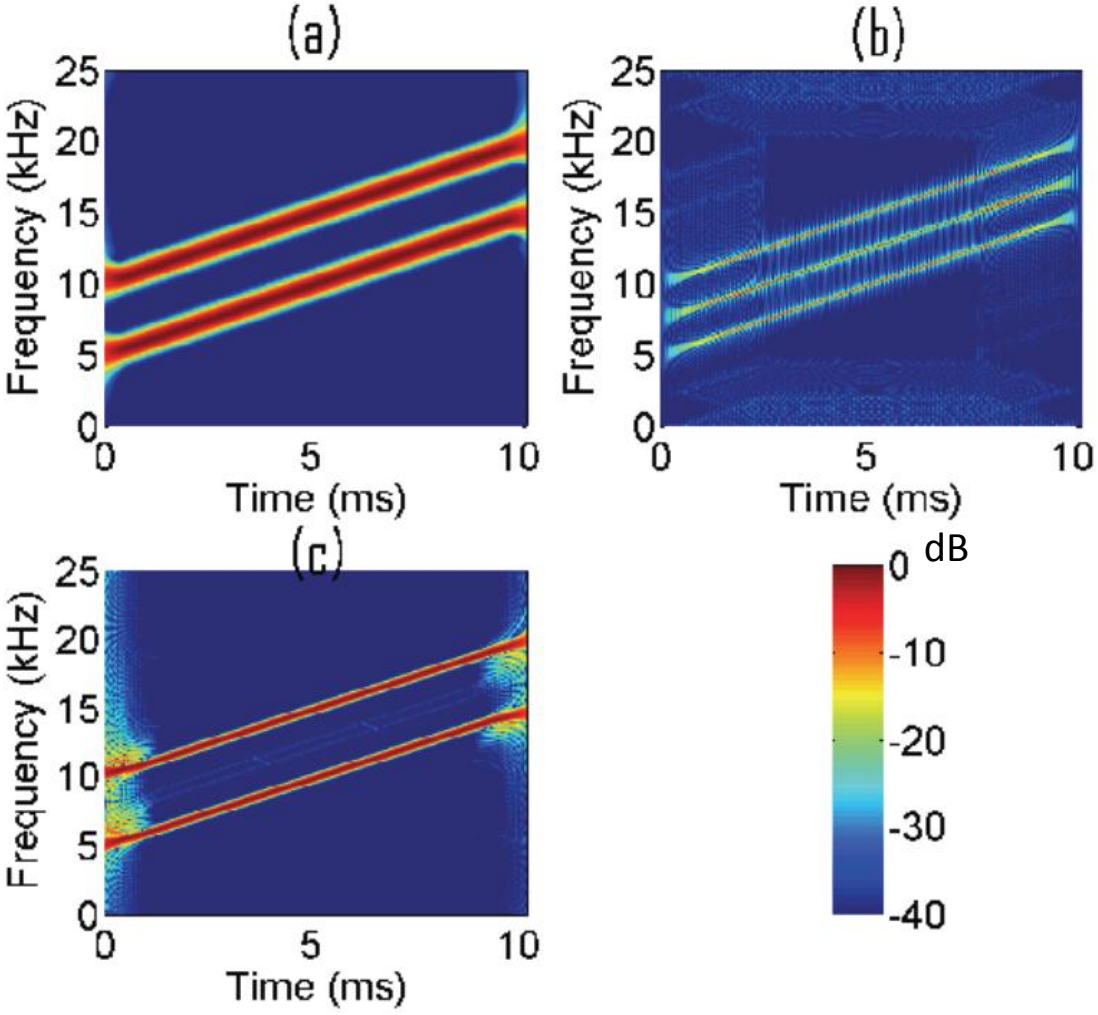

Fig. 2.9 (a) spectrogram of two linear chirps (b) Wigner-Ville transform of two linear chirps with processing artifact present (c) Pseudo Smoothed Wigner-Ville transform of two linear chirp. Each plot is in dB and normalized to its maximum value [1].

2.5 Conclusions

This chapter presents the basic knowledge of the signal processing that will be used in the later chapter. Further information on any of these topics is readily available in textbooks and/or compilation journals that have been referenced in the above sections.

CHAPTER 3

ULTRASONIC IMAGING ON CMUT ARRAYS

3.1 Introduction

Ultrasonic imaging is a readily available and well studied medical tool. The imaging method proposed here has two differences from the current methods: (1) this method will use capacitive micromachined ultrasonic transducers (CMUTs) instead of piezoelectric transducers and (2) this method will image passively instead of active imaging. While passive imaging is not a new concept, the recent development of CMUTs has allowed for the application of passive imaging using the cross-correlation of ambient noise at MHz frequencies.

To apply the passive imaging technique at MHz frequencies, it is very important to have a sensor with a noise floor below that of the thermal-mechanical noise. Above 5MHz it is difficult for piezoelectrics to have a noise floor below the thermal-mechanical motion of a fluid in part due to dicing the piezoelectrics small enough and having the amplifiers integrated close enough to the sensor. On the other hand, CMUTs arrays monolithically integrated with complementary metal oxide semiconductor (CMOS) electronics have demonstrated to have self noise below the Brownian motion of a fluid in a frequency band of [1-30] MHz [15,16,17,18].

There is an advantage of using passive imaging rather that active imaging. Active pulse echo ultrasound (CMUT or piezoelectric) sends a high energy pulse into the medium and then records the echoes (pulse-echo measurements). The active pulse has

the effect of "blinding the array" in the near field as a result from the large pulse that was transmitted which over saturates the receivers. A passive method based on the cross-correlation of ambient noise can remedy this issue with the tradeoff of imaging time and/or SNR. This chapter of the thesis will illustrate the feasibility of using a passive CMUT array to image the near field by taking advantage of the thermal-mechanical noise present in water and oil at MHz frequencies.

3.2 Baselines Noise Tests

3.2.1 Properties of CMUT Array

The CMUT array used in these experiments were fabricated and developed by the Degertekin group at Georgia Tech. The arrays were developed as a dual ring array for the purposes of Forward-Looking Intravascular Ultrasound. Each array consists of an outer receiver ring of 32 elements (725 um diameter) and an inner transmit ring with 24 elements (572 um diameter) as shown in Fig. 3.1. Each element was comprised of four membranes arranged in a 2x2 shape as seen in the inset of Fig. 3.1. Notice that the membranes farther from the center of the ring array are slightly larger than the other two membranes. This size discrepancy means that each element is comprised of two natural frequencies.

The membranes were designed to operate in a frequency range of 10-20MHz with each membrane having an area of 70x70 um. To record data from the 32 element receive ring array there are four 8-to-1 multiplexers. This limits the array to being able to only simultaneously record four receivers at any time (one sensor for each quadrant of 8

sensors). Further details about the fabrication and specifics of the CMUT arrays used can be found in the following references [9,18,19]

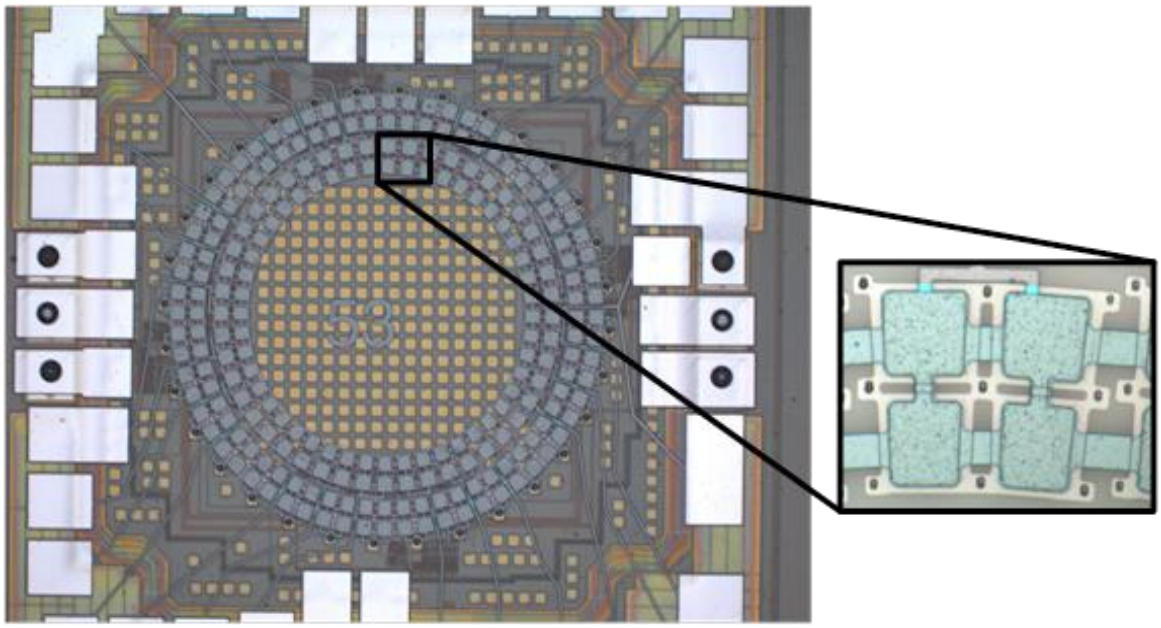

Fig. 3.1 Overhead view of the CMUT array used in the experiments. The outer receive ring of 32 elements is highlighted in green and the inner transmit array of 24 elements is highlighted in red. The exploded view illustrates one transmit element which is composed of 4 membranes of two different sizes [16].

3.2.2 Recording Thermal Mechanical Noise

The first step to use the CMUT arrays for noise based imaging required that the CMUT membranes be sufficiently sensitive to record the Brownian (Thermal-Mechanical) motion of the fluid. This research was carried out by Gurun and was presented in his paper [16,17]. To accomplish this goal, the output of a single membrane was observed at different bias voltages as shown in Fig. 3.2. The collapse voltage of the membrane was about 100 volts. The bias voltage was set to 0 volts to obtain a baseline noise level, which is only due to the front end electronics, and then subsequently the bias

voltage was set to 60 and 95 volts. The latter voltage is very close to collapse resulting in a very high sensitivity. From Fig. 3.2, it is clear that as the DC bias is increased the output voltage is reading information beyond the electronic noise. The additional noise, is the thermal-mechanical noise, which is present in the fluid.

There are certain regions in this plot that are of interest. From the 75 DC bias noise curve to the 90 DC bias noise curve, the peaks between 5-10 MHz increase and shift towards lower frequencies with the increasing bias voltage. This is due to the increase in sensitivity of the membrane and a result of the spring softening effect, which results from the larger bias voltages. The presence of two resonance peaks is because each CMUT element consists of four membranes with two different sizes and subsequently two separate resonance frequencies (see Fig. 3.1). Also of note is the region between 10-20 MHz, which then tapers to the baseline electronic noise above 20 MHz with several dips.

This proof of concept shows that the CMUTs can record the random fluid particle motion of the fluid. Also of note is that higher the bias voltages (i.e. very close to collapse) result in greater sensitivity. The subsequent tests use these CMUT arrays and have the bias voltage set near collapse to ensure the best possible sensitivity and subsequently the best possible SNR from the CMUT array.

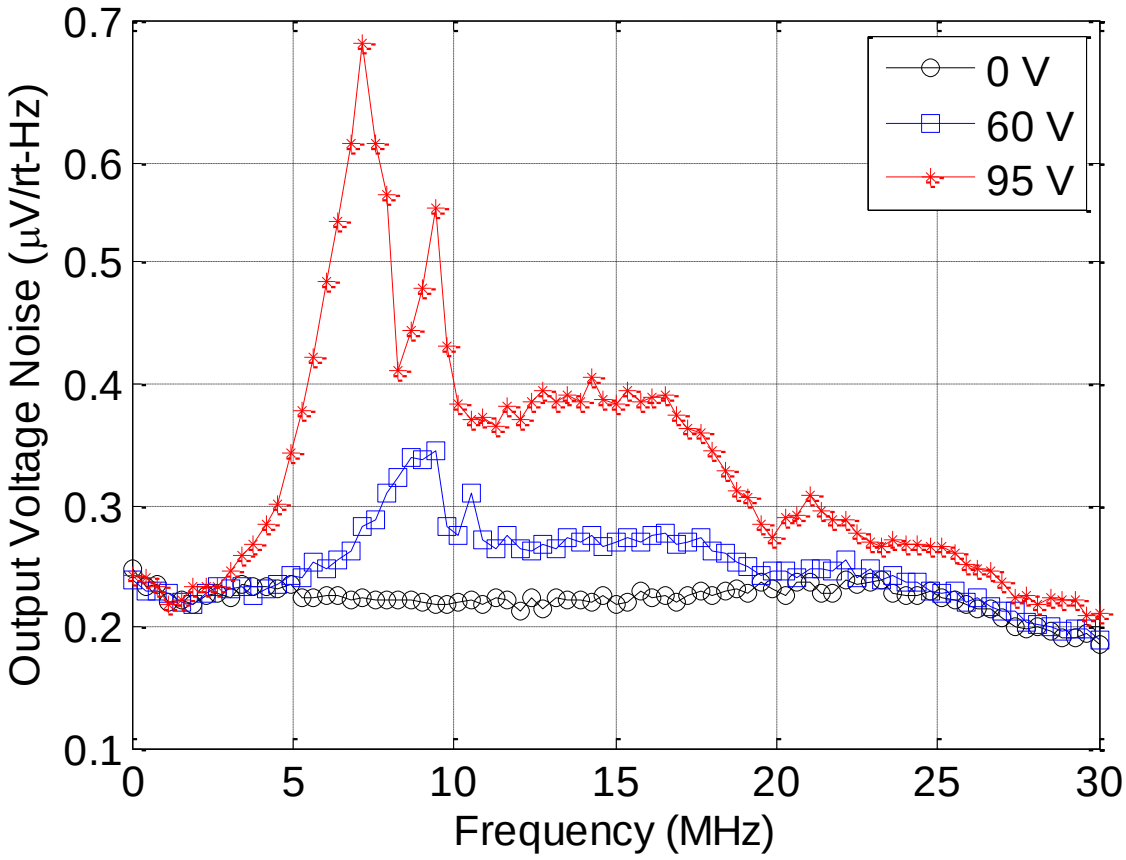

Fig. 3.2 Output voltage noise of one CMUT membrane under different bias voltages of no bias (0 V) which is the black line and shows the electronic noise, 60 V which is the red line which shows noise being recorded above the electronic noise floor and 95 V which is very near collapse.

3.3 Experiments

Two experiments were performed to test the feasibility of passive acoustic imaging with the CMUT array. The first experiment's objective was to record the reflection of the air-water interface and observe the SNR emergence of the reflection when the CMUT array was immersed in water. This test used the cross-correlation between two sensors on the array and hence the results are only a single waveform. The Green's function approximation contains all paths of noise between the two sensors

including a direct path along the surface of the array and the fluid propagating wave, which reflects off the air-water interface and its subsequent reflections. For imaging purposes, the acoustic propagating wave is the most important as it contains information about the height of the water over the array. The second test used the full receive array to image a point scatterer that was placed in oil above the immersed array. A passive image of the point scatter was produced and compared to active pulse echo imaging of the point scatterer.

3.3.1 Air-Water Interface Test

3.3.1.1 Setup

The first test to passively image on the CMUT array was to observe the expected reflections of the air-water interface with different levels of water above the CMUT array. For this test, the array was immersed in a layer of water either ~1.5 mm or ~2.0 mm thick as shown in Fig. 3.3. Data was collected from sensor one and sensor two as depicted in Fig. 3.4. The distance between the two sensors is 513 μm and the expected arrivals of the reflected sound between them is ~2.0 μs for a water height of 1.5 mm and ~2.7 μs for a water height of 2.0 mm using a sound speed of 1500 m/s for the water. The array was biased at 95 V (very close to collapse). The data was recorded on both sensors for 1 s. It was frequency whitened in the applicable band of the receivers [5-25 MHz] and subsequently cross-correlated (no amplitude clipping was necessary). The frequency whitening was utilized to further enhance the bandwidth and mitigate the restricted band of the receivers.

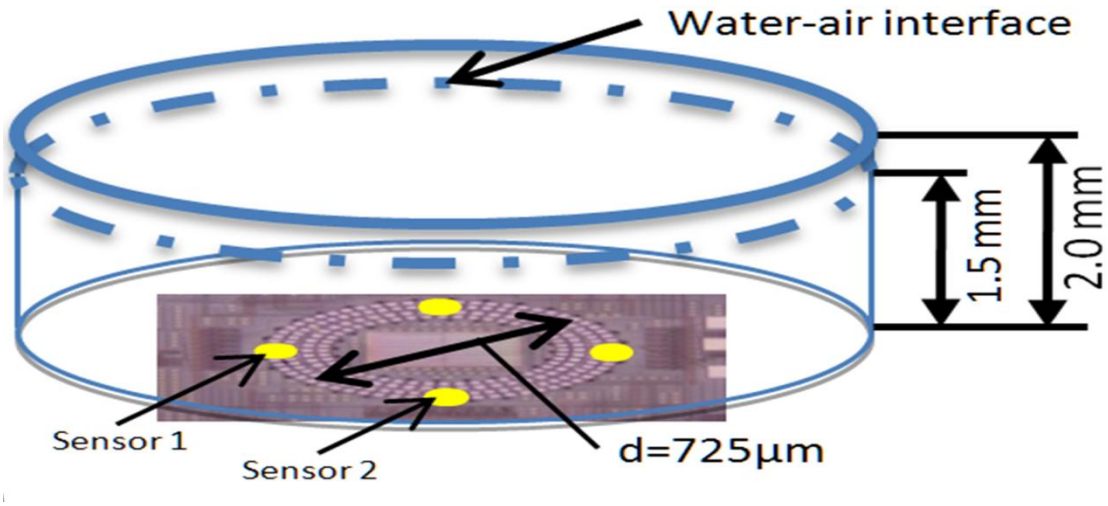

Fig. 3.3 CMUT array is placed in a dish with two different water heights above the array (1.5 and 2.0mm). Sensor one and two are denoted on the figure illustrating that they are separated by an arc length of 90 degrees.

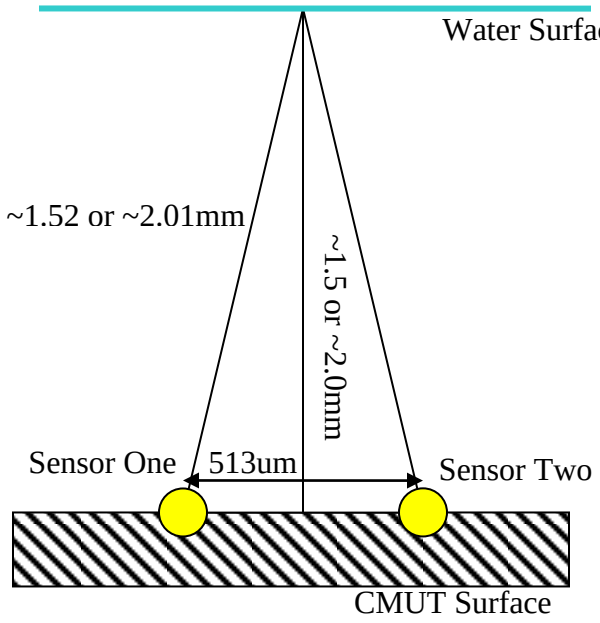

Fig. 3.4 Sensor one and two are separated by 513um and the acoustic path to travel from sensor one to the air-water interface and then back to sensor two is either 3.04 mm (2.0 µs for c=1500 m/s) or 4.03 mm (2.7 µs for c=1500 m/s) for the 1.5 and 2.0 mm water height respectively.

3.3.1.2 Results and Observations

The two cross-correlation waveforms, one for each water height, are presented in Fig. 3.5(a). There are three areas of interest; the large central peak, the low frequency wave between +/- 0.5 to 1.5 μs, and the arrivals that occur at 2.0 μs or 2.7 μs for the 1.5 mm and 2.0 mm water height respectively. The most pertinent area of interest is the arrivals. The arrivals show that indeed that the 1.5 mm water height has an arrival that is appropriately before than the arrival from a water height of 2.0 mm. Another interesting point from the waveforms is the large central peak. The large central peak corresponds to components of the waveforms that have no time delay. In other words, the part of the recorded waves, which occur at the same time, results with the arrival at or near zero. The most likely reason for this central peak is that it results from the non-propagating electronic noise. Since both of the sensors used the same electronics to capture their signal, any noise from those electronics would result in this zero time delay peak. The last area of interest is the oscillations that arrive immediately after the central peak and before the 1.5 mm arrival. These waves are crosstalk waves that travel on the interface between the transducer and the water. There has been extensive work to researching crosstalk on CMUTs, and if the crosstalk is removed, these arrays could image as close to 0.5 μs or just 275 μm above the array [11].

Another way to visualize the waveforms is to have a time-frequency representation of the cross-correlation waveform. Fig. 3.5(b) is the time frequency image of the 1.5 mm cross-correlation waveform. The time-frequency image was obtained by using the Smoothed Pseudo Wigner-Ville time-frequency method. From this perspective, additional information is available for each of our time windows of interest. First off, it

is clear that the central peak is very strong and broadband having frequency content in the entire frequency band that was measured [5-25] MHz. The crosstalk wave is also apparent in the time-frequency representation and shows that the crosstalk is relatively narrowband, [5-9] MHz. The acoustic propagating wave, which has the arrival near 2.0 μs, contains frequency content from about [12-22] MHz. This suggests that sufficient frequency filtering can eliminate the cross talk and focus on the propagating wave. However, a more narrow frequency band will result with a broader arrival in the time domain. Therefore, frequency filtering needs to be carefully considered, a frequency band that is too wide introduces noise and a frequency band that is too small distorts the arrival.

The last examination of the water-air interface experiment was to determine how the SNR of the arrival grew with increasing averaging time. Only the acoustically propagating waves were used to examine the SNR emergence. Hence, the signal was filtered in the band [12-22] MHz. The definition used for SNR is:

$$SNR(T) = \frac{\max\{abs(CC(\Delta t_{arrival};T))\}}{stdev\{CC(\Delta t_{noise};T)\}} \tag{3.1}$$

where $\Delta t_{arrival}$ is the time window around the expected arrival (1.9 μs<|t|<2.3 μs for 1.5 mm and 2.4 μs<|t|<2.9 μs for 2.0 mm), Δt_{noise} is the time window following the arrival window (2.5 μs<|t|<4.0 μs for 1.5 mm and 3.0 μs<|t|<4.5 μs for 2.0 mm). From this definition the SNR of the correlation waveforms shown on Fig. 3.5(a) are respectively 16.95 (1.5 mm water depth) and 11.34 (2.0 mm). The difference in the SNR is due to two main effects: 1) the closer the target, the more coherent noised sources between the two receivers illuminate the target and 2) the incident and scattered energy of the thermal-mechanical noise decay due to spherical spreading over the propagation distance.

Fig. 3.6 illustrates that the SNR does indeed grow proportionally to the square root of total averaging time, √T, as expected [31,41]. Best fit curves (solid lines) were applied to the experimental data (cross and dot symbols) with the trend in the form *a*(T)1/2+b*. Both trend lines were an excellent fit with r^2=.9959 for the water height of 1.5mm and r^2=.9928 for the water height of 2.0mm. Now that the cross-correlation waveforms have been examined, this information could be applied to imaging.

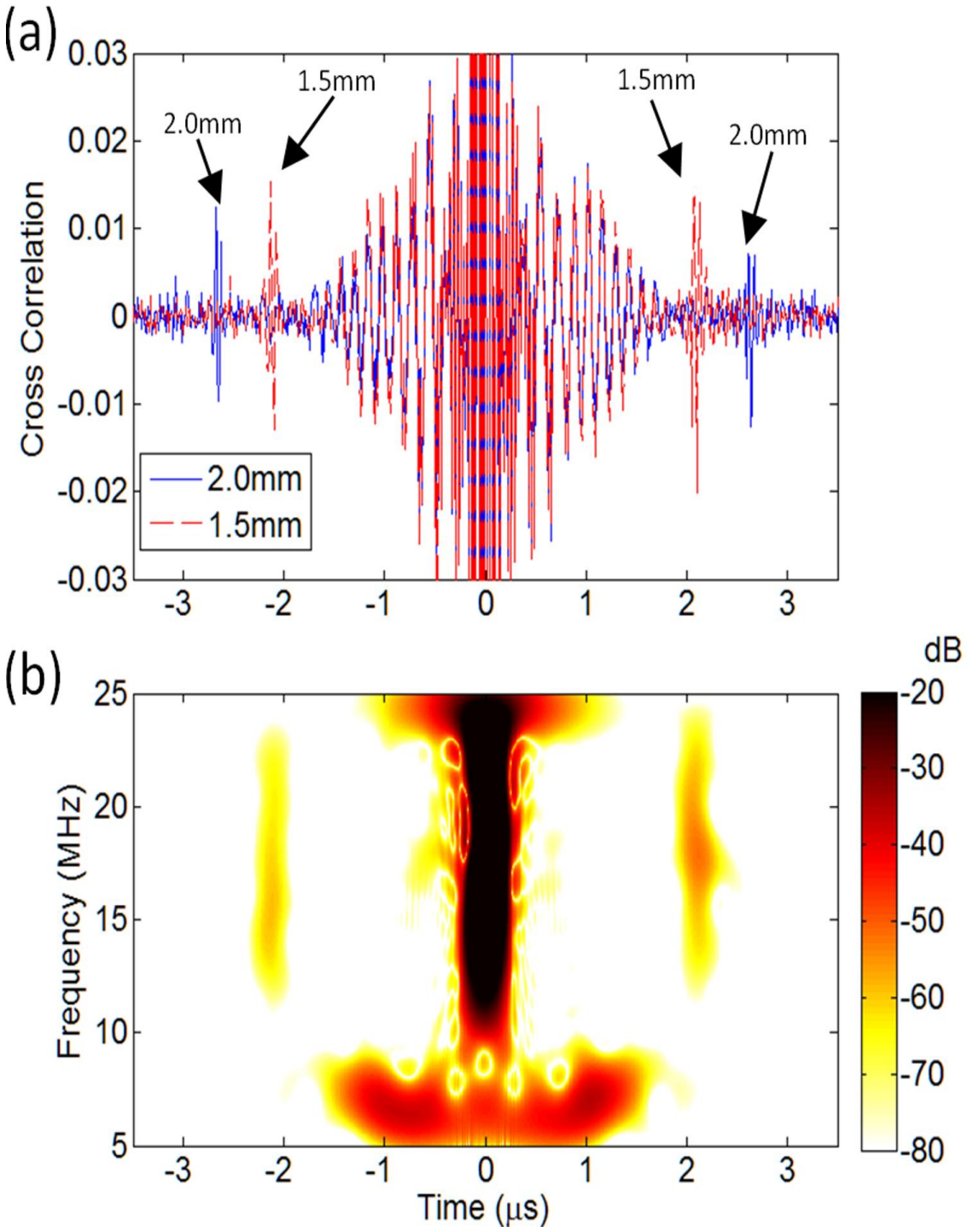

Fig. 3.5 (a) One second cross-correlation waveforms between sensors one and two for two different water heights of 1.5mm (red dashed) and 2.0mm (solid blue). (b) Time-frequency representation of the cross-correlation waveform for the 1.5mm water height. The color scale is logarithmic and normalized to the maximum value.

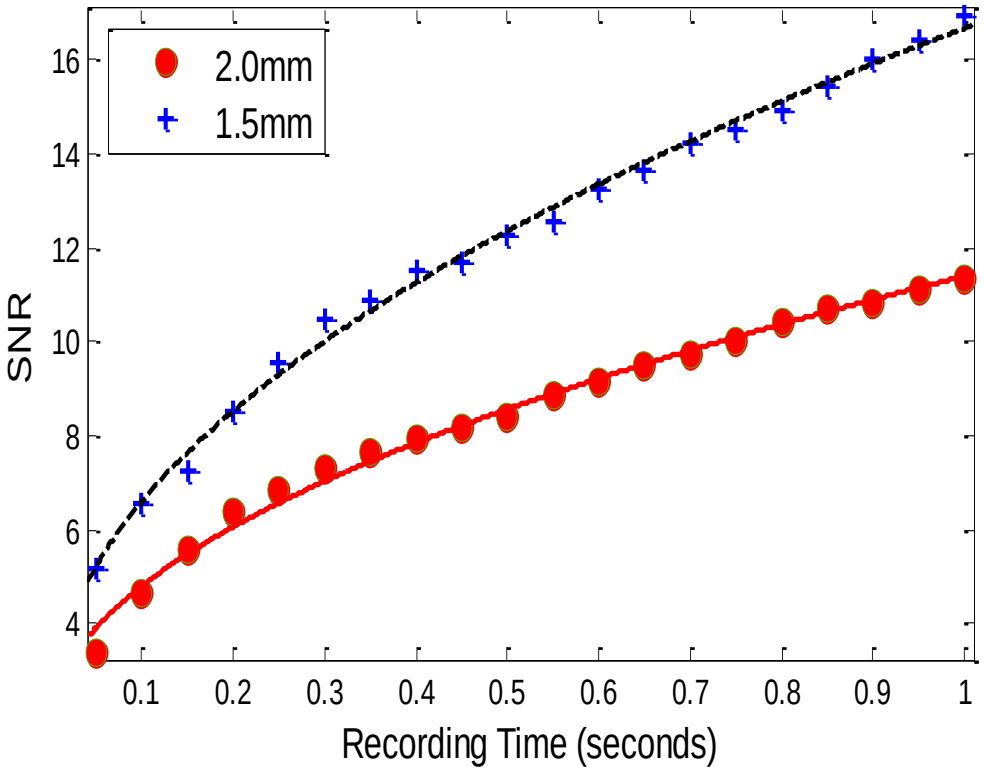

Fig. 3.6 Plots illustrating the relation between SNR and recording time (T) for both the 2.0mm and 1.5mm waveforms. The solid lines are best fit lines in the form of $a*(T)^{1/2}+b$ with a=9.67, b=1.72, r^2=.9928 for water height of 2.0mm and a=14.8, b=1.90, r^2=.9959 for water height of 1.5mm.

3.3.2 Point Scatterer Test

3.3.2.1 Setup

The second test entailed the CMUT array imaging a point scatterer both passively and actively (for validation). The CMUT array was immersed in oil with a point scatterer (500 μm diameter) suspended 1.3 mm above the array as shown in Fig. 3.7(a). The oil height was about 10 mm, which ensured that reflections from the air-oil interface arrived much later than the reflections from the point scatterer. Since only four receivers could be recorded simultaneously, the total of passive imaging waveforms was severely limited. Ideally, if all 32 receivers recorded at the same time then there would be 496 possible waveforms between different sensors. Due to the restrictions, only 48 waveforms were captured (eight sets of recording on four sensors resulting in six waveforms from each cross-correlation of the four sensors for one set).

Again, the array was biased at 95 V and the data was recorded 1 s long time durations. Since the receivers were in oil and not water, the optimal band for filtering was [16.5-30] MHz. The signals were cross-correlated for each group of four sensors. The four sensors were each separated by 90 degree. For example, if the receivers were numbered in order about the circle from 1 to 32 the first group of 4 sensors were 1, 9, 17, and 25 while the second group of four was 2, 10, 18, and 26. Once all the cross-correlations were complete, each waveform was time windowed between 1.6 and 2.2 μs, which was the expected time window for arrivals. The 48 cross-correlation waveforms were then compiled to form an image using spherical beamforming as described in Chapter 2.

An active pulse echo image was also used to verify the passive image technique. Each of the 24 transmitters was pulsed one at a time, and for each transmitter data was recorded on all 32 receivers. This was done by pulsing the first transmitter 8 separate times and recording the results on the 8 different groups of four receivers. For example, the first transmitter was pulse and data was recorded on receivers 1,9,17, and 25. Then the first transmitter pulsed again but this time data was recorded from 2, 10, 18, and 26. This process continued until all receivers were recorded for the first transmitter. Then this process was repeated for each transmitter resulting in 768 waveforms. In the same fashion as the passive image, the active pulse echo waveforms were frequency filtered and then compiled with spherical beamforming to make the image.

3.3.2.2 Results and Observations

As mentioned, spherical beamforming was used to obtain the images from the passive and active methods. This method allows imaging in 3 dimensions, however for this proof of concept test the image was kept as 2 dimensional. The cross-range is the dimension that is in the plane of the CMUT array while the range is the distance above the CMUT array as shown in Fig. 3.7(a).

The passive image is shown in Fig. 3.7(b) with the cross-range across the x-axis and range for the y-axis. The color scale is logarithmic and normalized to the maximum value in the image. For this image, only 32 out of the 48 waveforms were used due to the quality of the signals. The qualities of the signals were determined by comparing the using the definition of SNR from Eq. (3.1) (both Δt_{noise} and $\Delta t_{arrival}$ were chosen appropriately for this particular dataset). If the SNR was above 5.5, then the signal was

used for the image. From this image, a bright spot is clearly visibly at 1.3 mm range and just off the center of the array.

The verification of this image is the active pulse echo image, which is shown as Fig. 3.7(c). Out of the 768 waveforms only 171 were kept using the same SNR standards as the cross-correlations. Some pulse echo waveforms were thrown out due to the ringing of the pulse contaminating the arrival of the reflection, which is a real concern when imaging so close to the array. The active pulse echo image shows that the point scatterer is found in the exactly same location as the passive image. Also, note that the color scale in both figures is the same meaning that the point scatterer is being resolved with similar SNR between the two methods by visual inspection. The difference in the side lobe structures, the other bright points at different cross-range positions but still around 1.3mm range, is due to the difference of total number of waveforms used.

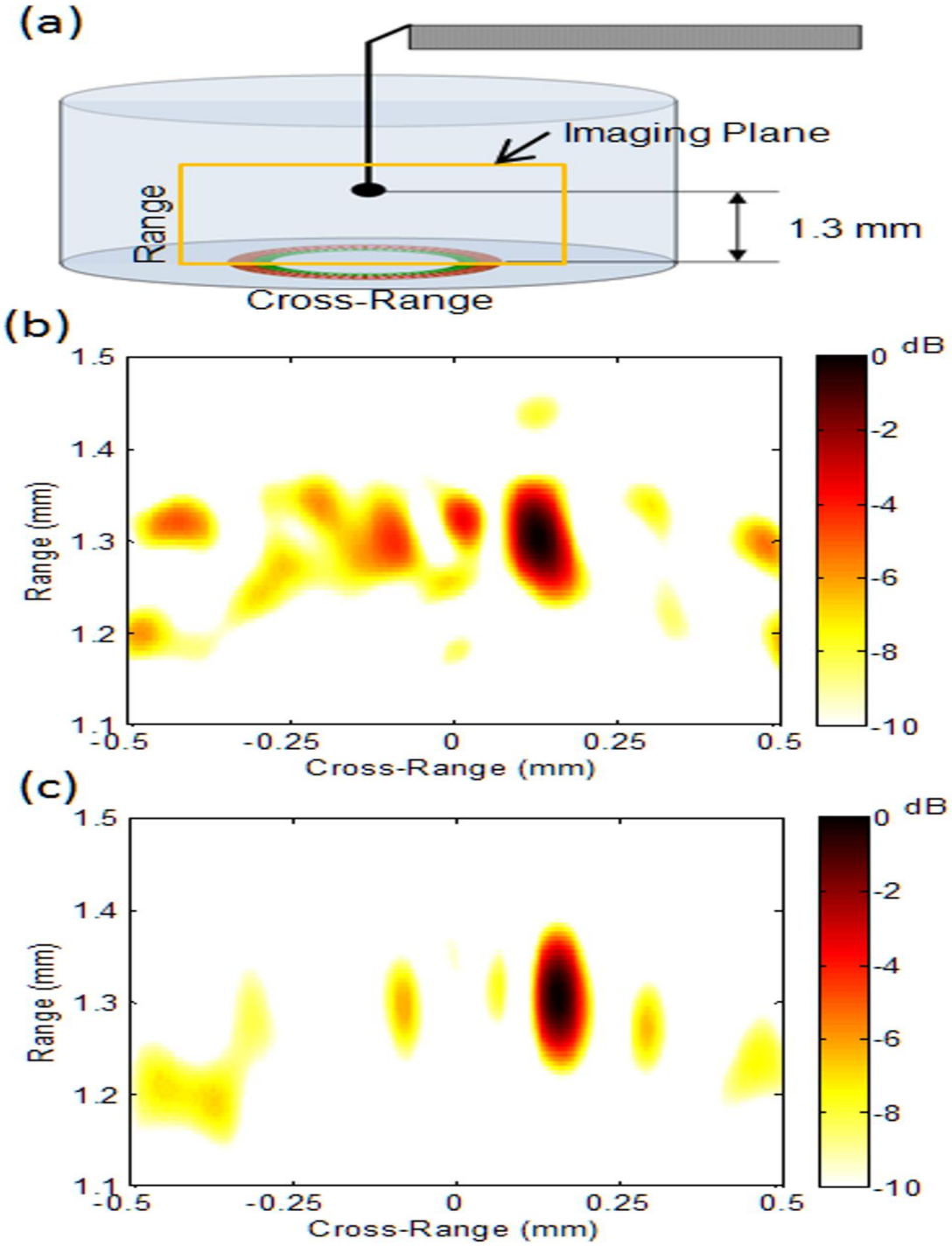

Fig. 3.7 (a) Experimental setup for both the active and passive image test with a point scatterer positioned 1.3mm. Everything was immersed in an oil layer of 1cm. (b) Passive image of the point scatterer (c) Active image of the point scatterer

3.4 Conclusions

This chapter presented the feasibility of ultrasonic passive imaging. First, it was determined by prior research that CMUTs were capable of measuring the thermal-mechanical motion of a fluid. Two experiments were then conducted. The first experiment focused on finding the reflections of the air-water interface. This experiment showed that there were three interesting portions of each cross-correlation waveform; the non-propagating noise at the central peak, the crosstalk that exists in a narrow frequency band, and the arrivals that were at the correct locations given the different water heights. It also showed the SNR increases proportional to the amount of recording time. The second experiment focused on passively imaging a point scatterer immersed in oil and comparing it to active pulse echo image. Both imaging techniques positioned the point scatterer at the same location with similar SNR. These two experiments illustrate that passive ultrasonic imaging is possible.

CHAPTER 4

OCEAN MONITORING

4.1 Introduction

The first person to suggest the cross-correlation of ambient noise as a viable method for ocean monitoring was Roux in 2004. Roux's paper used the cross-correlation of ambient noise in water ~1800m deep on four coplanar vertical line arrays (VLAs). Coherent wave fronts were extracted using a frequency band of [70-130] Hz and a separation of up to 3.5km. This result led Roux to believe that passive ocean monitoring was feasible using the cross-correlation of ambient noise [28]. However, in order for the cross-correlation of ambient noise to be feasible for ocean monitoring the Green's function must be extracted quickly and robustly. This is because the ocean environmental noise can only assumed to be constant over very short time scales (ie less than a half a day; eg. tides, currents, rain, surf conditions). Hence, the Green's function arrivals must have sufficient SNR with only a few hours of averaging.

Three other notable papers have utilized the cross-correlation of ambient noise as a method to monitor the ocean. In 2008, Fried et al. was able to use the cross-correlation method with a bottom mounted horizontal array to find the critical angle between the water and sediment. In Fried's experiment, the noise was in the band [250-750] Hz and dominated by biological noise in shallow water (~21m) [13]. Another experiment used horizontal arrays and beamforming to enhance the SNR of the Green's function. Siderius et. al used horizontal arrays as a fathometer and showed the enhanced effects of beamforming in an environment dominated by surf noise in the frequency range of [200-4000] Hz [35]. Both of these experiments had relatively diffuse noise fields allowing for

easy extraction of the Green's function estimate. However, recording near harbors introduces the possibility of loud shipping events that can dominate the ambient noise. Chapter 2 illustrated that the cross-correlation of ambient noise method works when the ambient noise is diffuse. To successively use the cross-correlation of ambient noise technique near harbors or shipping lanes the loud events will need to be mitigated.

A recent paper by Leroy et al. presented a possible solution to passive ocean monitoring near harbors. The solution consisted of using noise recorded on VLAs and then uses spatio-temporal filters to aid in the enhancement of the SNR. The method was demonstrated in the frequency band of [100-1000] Hz in a shallow water coastal region where the noise was dominated by shipping traffic in one direction (positive arrivals). While the method was proven to work with the negative arrivals which were more diffuse over a duration of one day and a short array separation (143m) it was not extended to further time durations, array separations, or deeper array depths [24,25].

This chapter presents a long term passive ocean monitoring experiment using the cross-correlation of ambient noise technique near a harbor. The experiment spans six days of passive recording on two VLAs near San Diego, CA, which was dominated by loud shipping events. The recorded data was preprocessed and utilized the method developed by Leroy et al. to enhance the SNR emergence by using spatio-temporal filters. This method allowed acceptable SNR emergence with 8 hours of averaging. The arrivals and their SNR were tracked over the six days revealing a trend in changing arrival times, in other words effectively monitoring the environmental changes. Insights about selecting an effective spatio-temporal filter and how to implement this method for longer durations are also discussed.

4.2 Experimental Setup

Two bottom moored VLAs were positioned 450m apart on the Coronado Bank (depth ~150m) offshore of San Diego, California as shown in Fig. 4.1. Ambient noise was continuous recorded for six days spanning from January 31, 2009 to February 5, 2009 (denoted hence forth as Day 1-Day 6) using a sampling frequency of 25 kHz. Each of the VLAs was time synchronized with GPS timing at time of deployment and has 16 elements with 1 m spacing. The lowest element was positioned 7m above the seafloor as denoted in Fig. 4.2. Further details of the VLA can be found in previous publications, which have used the same VLAs [24,25,36].

Due to the close proximity of the test site to San Diego harbor the ambient noise was largely comprised of loud discrete shipping noise. A spectrogram of the ambient noise is presented in Fig. 4.3. The spectrogram averaged all 16 elements from the first VLA for the third day of recording. As seen in the spectrogram, there are independent loud events, evidence of the shipping traffic. Since the cross-correlation of ambient noise requires a diffuse noise field for best results, special care and preprocessing was undertaken to attempt to force a diffuse field and mitigate the effects of loud events. The first consideration was given to the frequency band to be used for the passive ocean monitoring method. There were two aspects to consider for the frequency band; (1) loud enough noise to have cross-correlation of ambient noise have acceptable SNR and (2) a band that is appears to be diffuse (ie no discrete events). It was determined that the best band by observation was [250 1500] Hz. This band had a sufficient level of noise, yet it did have discrete events present, which will need to be mitigated in subsequent

processing. The band above 5 kHz was also considered, due to it appearing to be much more diffuse. However, the lack of energy in that band would not have been able to resolve arrivals efficiently.

Once the frequency band was chosen, preprocessing steps were used to make the noise field more diffuse. The preprocessing focused on two factors of a diffuse field, the first being that the noise is broadband in the frequency band. Hence, frequency whitening was used as described in Chapter 2. The second property of a diffuse field is that there are no strong outliers with energy. Therefore, in this experiment each discrete ship has much more energy than the normal ambient noise. To mitigate this effect, any signals that were stronger than three standard deviations of the noise were clipped again as described in Chapter 2. These steps have been used in previous experiments and have shown to enhance the extraction of the Green's function [29,25]. After the preprocessing was performed, the data was cross-correlated in 1 minute intervals for all elements on VLA1 to all elements on VLA2 for a total of 256 cross-correlations for each 1 minute of recording on the 32 elements.

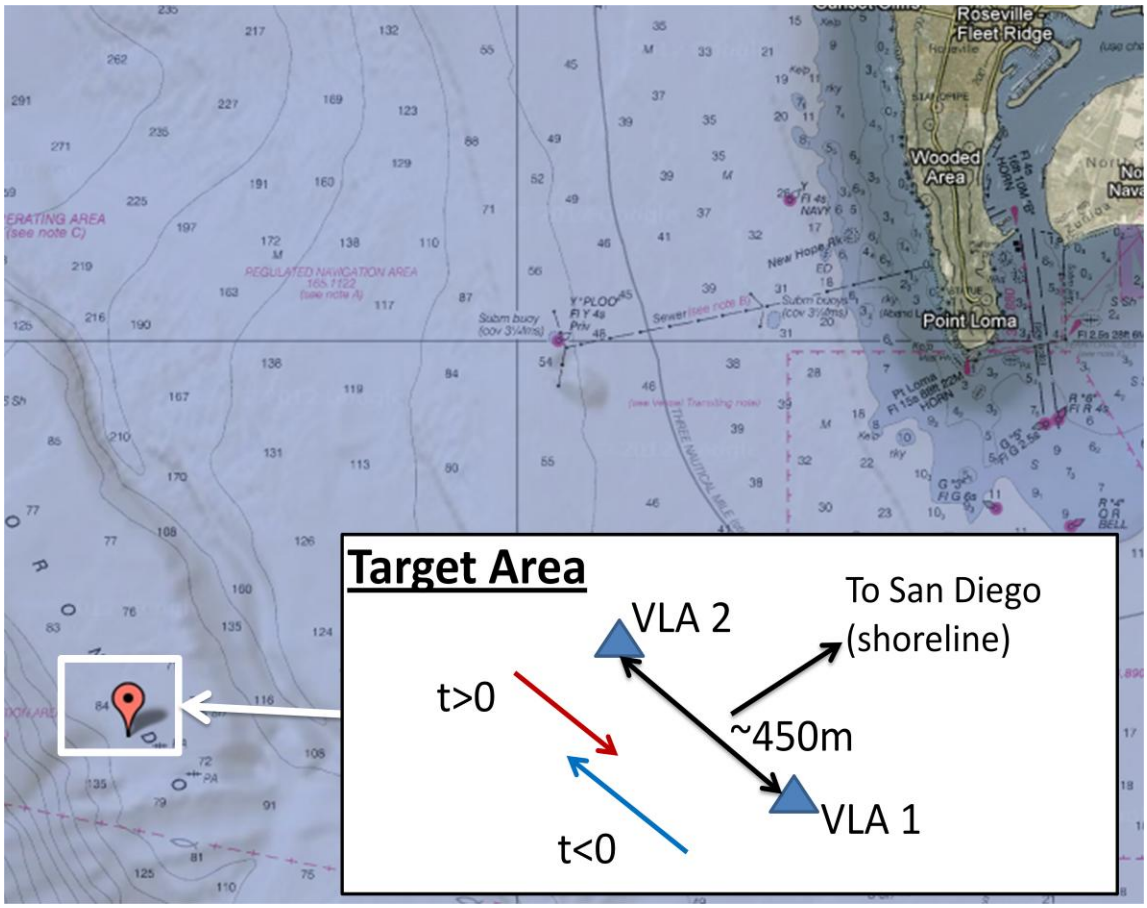

Fig. 4.1 The experimental setup located off the coast of San Diego, CA. Two vertical line arrays (VLAs) were positioned on the Coronado Bank with 450m separation positioned as shown in the insert. Positive time delays in the cross-correlations correspond to sound traveling successively from VLA2 to VLA1 and vice versa for negative time delays.

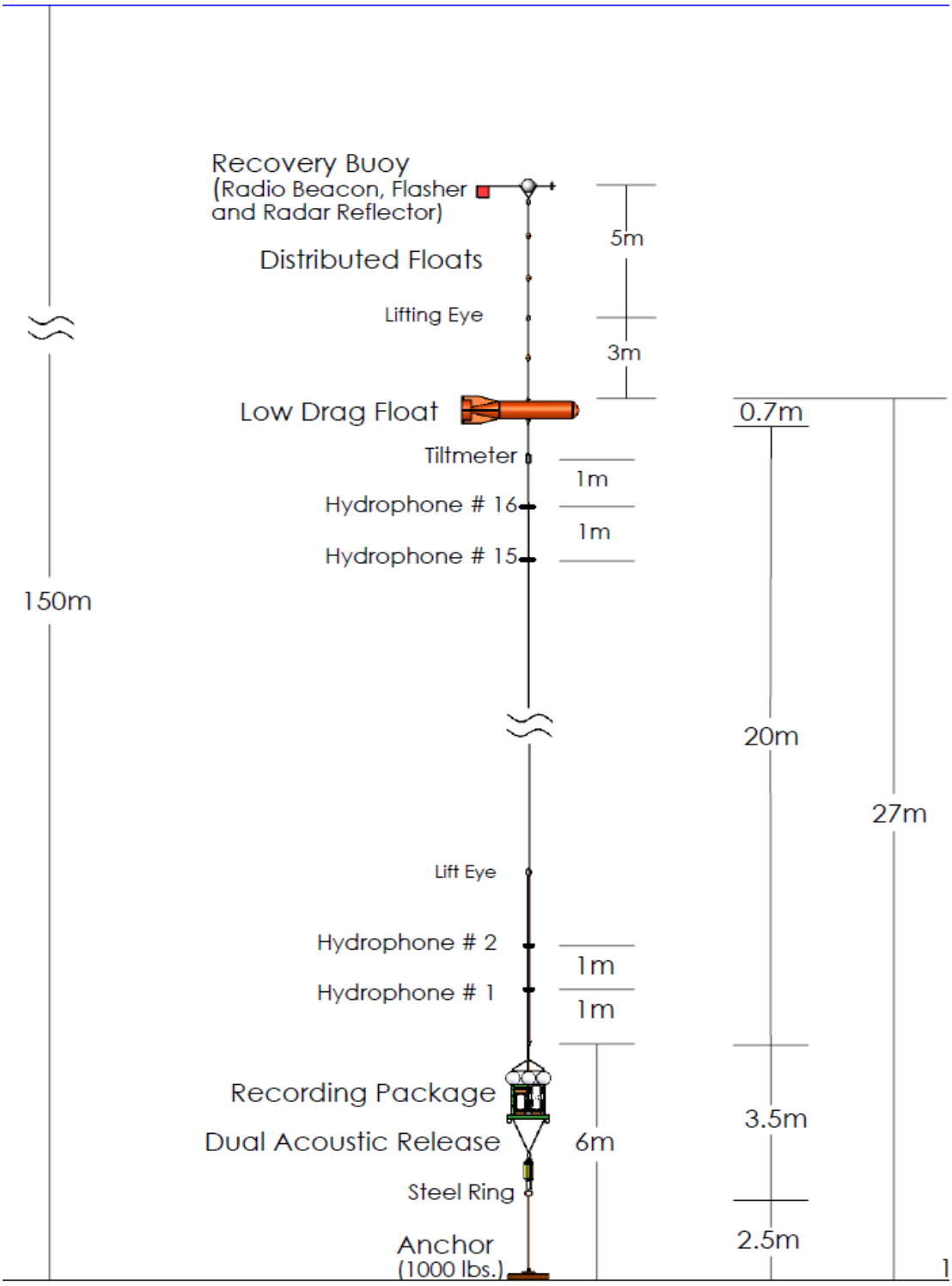

Fig. 4.2 Schematic of the bottom moored arrays that were used

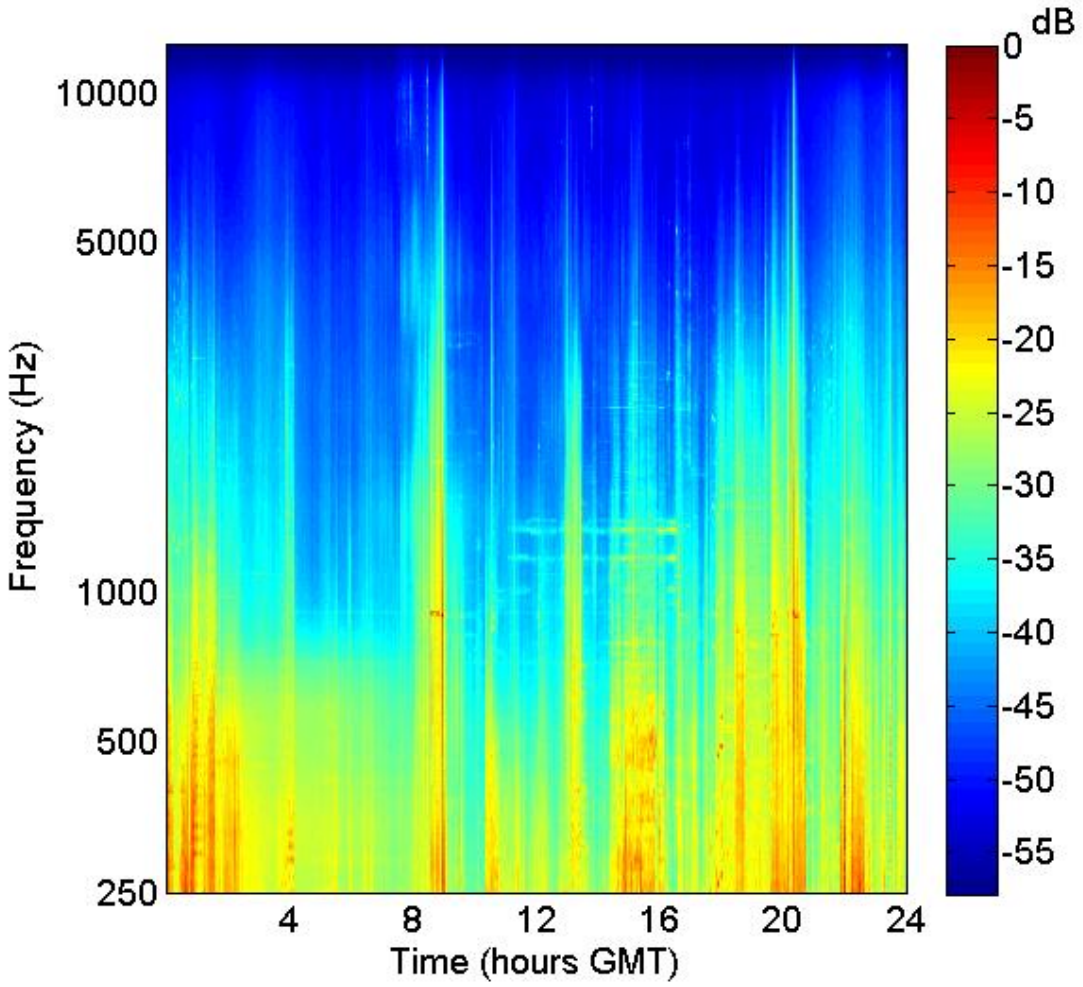

Fig. 4.3 Spectrogram average of all 16 sensors on VLA1 over 24 hours of the third day in recording. Plot is in dB and normalized to the maximum value.

4.3 Spatiotemporal filtering and arrival tracking

This section reiterates the advantages of the spatio-temporal filter used in the paper by Leroy et al. and applies them to speed the SNR growth or arrivals. This process hinges on obtaining an effective spatio-temporal filter. This filter contains information about each expected arrival time between all elements in each VLA and can be derived either from a model or experimentally. Once the reference matrix is selected, it can be applied to a moving summation window to obtain an acceptable SNR for the arrival. This rest of this chapter will examine several different reference matrices and how each performs in the arrival tracking over the entirety of the experiment.

4.3.1 Reference Selection

As stated earlier the reference matrix or spatio-temporal filter can be derived experimentally or analytically. Leroy et al. showed that a long-time average was an effective method to establish an experimental reference matrix. This reference matrix is made from extensive averaging. The time averaged cross-correlation, $R_{i,j}(t;L,N)$, can be defined by the following equation:

$$R_{i,j}(t;L,N) = \sum_{k=L}^{L+N} C_{i,j}(t;k) \tag{4.1}$$

where $C_{i,j}(t;k)$ is the kth minute of the cross-correlation waveform between the ith element of VLA1 and the jth element of VLA2. The variable L denote the starting minute while N is the number of minutes in the averaging.

Fig. 4.4 shows three coherent wavefronts, which were obtained by averaging the experimental data. The wavefronts are made by using an average cross-correlation, $R_{4,j}(t,L,N)$, from sensor 4 on VLA1 to all other sensors on VLA2 (denoted as depth on the y-axis). The first two wavefronts (Fig. 4.4(a,b)) used 1920 minutes of averaging

starting from the first minute of the third day or the fifth day (N=1920 and L=2881 or L=5761). Both of these wavefronts used the negative arrivals as denoted on the time axis. Fig. 4.4(c) used the positive delays (riddled by more discrete shipping) and used only the minutes of one loud ship passing (L=3180 N=160).

There are some important differences between the displayed wavefronts. The first and most obvious is the difference in SNR. The wavefront that was only an average of one passing ship (Fig. 4.4(c)) has a greater SNR than the other two wavefronts (Fig. 4.4(a,b)). Also of note are the arrival times of the wavefronts. Again Fig. 4.4 (a,b) used the negative arrivals while Fig. 4.4(c) used the positive arrivals. There is a clear difference in the absolute arrival time between the positive and negative wavefronts. Due to the difference in the arrival time between positive and negative, the arrivals were not added coherently. Instead, they were processed independently. Based on the ordering of the signals $S_i^{(1)}(t;k)$ and $S_j^{(2)}(t;k)$ in the definition of the cross-correlation function in Eq. (2.1), the coherent arrivals from the negative arrivals (Fig. 4.4(a,b)) correspond to coherent ambient noise propagating successively between VLA1 and VLA2 (i.e. along the northward direction, see inset on Fig. 4.1), while the positive arrivals (Fig. 4.4(c)) corresponds to sound traveling from VLA2 then to VLA1. The two wavefronts for the negative arrivals appear to be similar in shape and arrival time albeit with a difference in SNR. This consistency in the negative arrival wavefronts suggests a relatively stable ocean environment over a period of days. This chapter will focus more on the negative arrivals, but just have well could have focused on the positive arrivals.

It is important to note that in an effort to reject loud interferers, if $C_{1,1}(t;k) > .01$ for any *t* then that particular minute, *k*, was removed from the summation so as not to bias

the summation by strong events. The elements, i=j=1, were chosen as the standard due to these sensors having a greater amplitude than the other combinations of i and j. The threshold of 0.01 was chosen based on visual observation of levels between different minutes and is equivalent to about 4 standard deviations of the cross-correlation waveforms (standard deviation 0.0022 for t<0,and 0.0026 for t>0;). Each of these experimentally derived reference matrices will be tested for the arrival tracking along with a plane wave beamforming model based reference matrix.

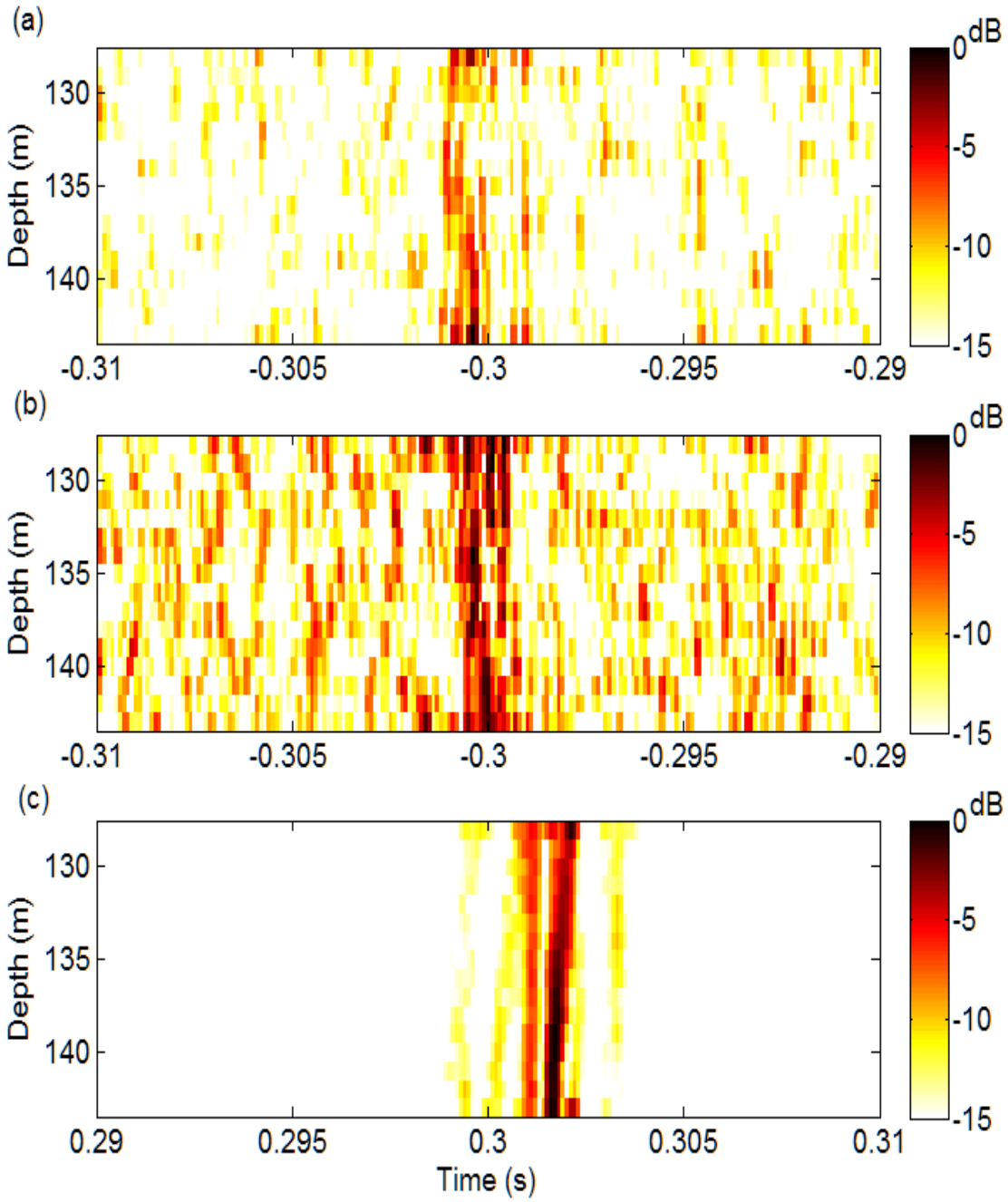

Fig. 4.4 Wave fronts are displayed by using the cross-correlation between sensor 4 on VLA1 to all sensors on VLA2 with averages of (a) $R_{4,j}(t, 4321,1920)$ with negative arrivals (b) $R_{4,j}(t, 7201,1920)$ with negative arrivals and (c) $R_{4,j}(t,3180,160)$ for the positive arrival corresponding to one loud ship passing.

To illustrate the difference of SNR emergence between using the spatio-temporal filter (reference matrix) and not using the filter, a direct comparison will be done. This comparison will track the SNR of the arrivals with SNR given by:

$$SNR(t,L,N) = 10\log_{10}\left(\frac{R_{i,j}(t_{low} < t < t_{high}; L, N)}{3\cdot\sqrt{\left(\frac{1}{N}\sum_{k=L}^{L+N}(C_{i,j}(t_{low} < t < t_{high}; k))^2 - \left(\frac{1}{N}\sum_{k=L}^{L+N}C_{i,j}(t_{low} < t < t_{high}; k)\right)^2\right)}}\right) \quad (4.2)$$

The radical in the denominator is the standard deviation of the cross-correlation waveforms with regard to the time variable, *t*, which is windowed around the expected arrival by t_{low} and t_{high}. (t_{low}=-.33 and t_{high}=-0.28 for the negative arrivals and t_{low}=0.28 t_{high}=0.33 for the positive arrivals). Assuming Gaussian statistics the peak values of $R_{i,j}(t,L,N)$ is divided by three times the standard deviation to obtain the SNR [2]. It has been shown that if the environment remains stationary the SNR will grow as the square root of recording time (√N) [31,41]. Hence using more one minute averages (increasing N) will result in a higher SNR if the environment is stationary with no loud interferers.

Fig. 4.5(a) shows how the arrivals SNR grows with increasing averaging time up to 24 hours (1440 minutes) for the summed average between the 4th sensor on VLA1 and the 2nd sensor on VLA2, $R_{4,2}(t,L,N)$. There are two trends that track the SNR of the arrival for the negative times and one for the positive. For each SNR emergence line, the starting minute, L, remains constant with the number of averaging minutes, N, increasing. It is clear that the SNR is not growing proportional to √N; instead the SNR grows only in short time windows which are due to one dominate ship whose effects could not be mitigated by the processing (i.e. 310-450minutes for the positive arrivals). Both the slow

emergence of SNR and the discrete SNR jumps justify the need to enhance the SNR by a spatiotemporal filtering process.

The method by Leroy et al. was subsequently applied to enhance the emergence of the arrivals. The method takes advantage of the spatial arrays by the use of beamforming as define by :

$$\hat{B}(f, L, N) = W_1^H(f)\,\hat{R}(f, L, N)\,W_2(f) \tag{4.3}$$

where $W_1^H(f)$ and $W_2(f)$ are the steering vectors that can be derived from a reference matrix as the first singular vectors obtained from the singular value decomposition (SVD) or the steering vectors can be defined to fit a model. $\hat{R}(f, L, N)$ is the Fourier transform of the summed cross-correlation $R(t, L, N)$. Again, L denotes the starting minute and N is the total minutes of averaging while ^{H} denotes the Hermitian operation.

The beamforming coherently combines all 256 cross-correlation waveforms into one waveform to effectively enhance the SNR. To measure the SNR the beamformed results in the frequency domain, $\hat{B}(f, L, N)$, were transformed to the time domain with an inverse Fourier transform $B(t, L, N)$. The time domain beamformed data was then used in Eq (4.2) by replacing $R_{i,j.}(t,L,N)$ and the results are shown in Fig. 4.5(b). Four separate cases were used as the weights. Case 1 used unitary weights, which corresponds to plane wave beamforming. This case was used as a baseline test and due to the visual observation of the wavefronts, which appeared to have little curvature. The other cases used a long time average as a reference (Case 2 $R_{i,j.}(t>0,3180,160)$; Case 3 $R_{i,j.}(t<0,2881,1920)$; and Case 4 $R_{i,j.}(t<0,5761,1920)$). The beamforming weights, $W_1^H(f)$ and $W_2(f)$ were obtained by taking the first singular vectors of the reference matrix $R_{i,j.}(t,L,N)$. As can be seen in comparison to Fig. 4.5(a), the SNR grows more steadily

when more averaging minutes are used for Cases 2-4. Case 1 plane wave beamforming, on the other hand, shows little to no SNR growth compared to Day 5 Case 4, which uses the same day but the weights are derived from the reference matrix, R(t,7201,1920). From Fig. 4.5(b), 480 minutes (8 hrs) was chosen as the duration of a moving window that will be used in subsequent processing due to all waveforms (that use a reference matrix) reaching an arbitrary threshold of 6dB SNR. This moving window is used to track the arrivals over all six days of recording.

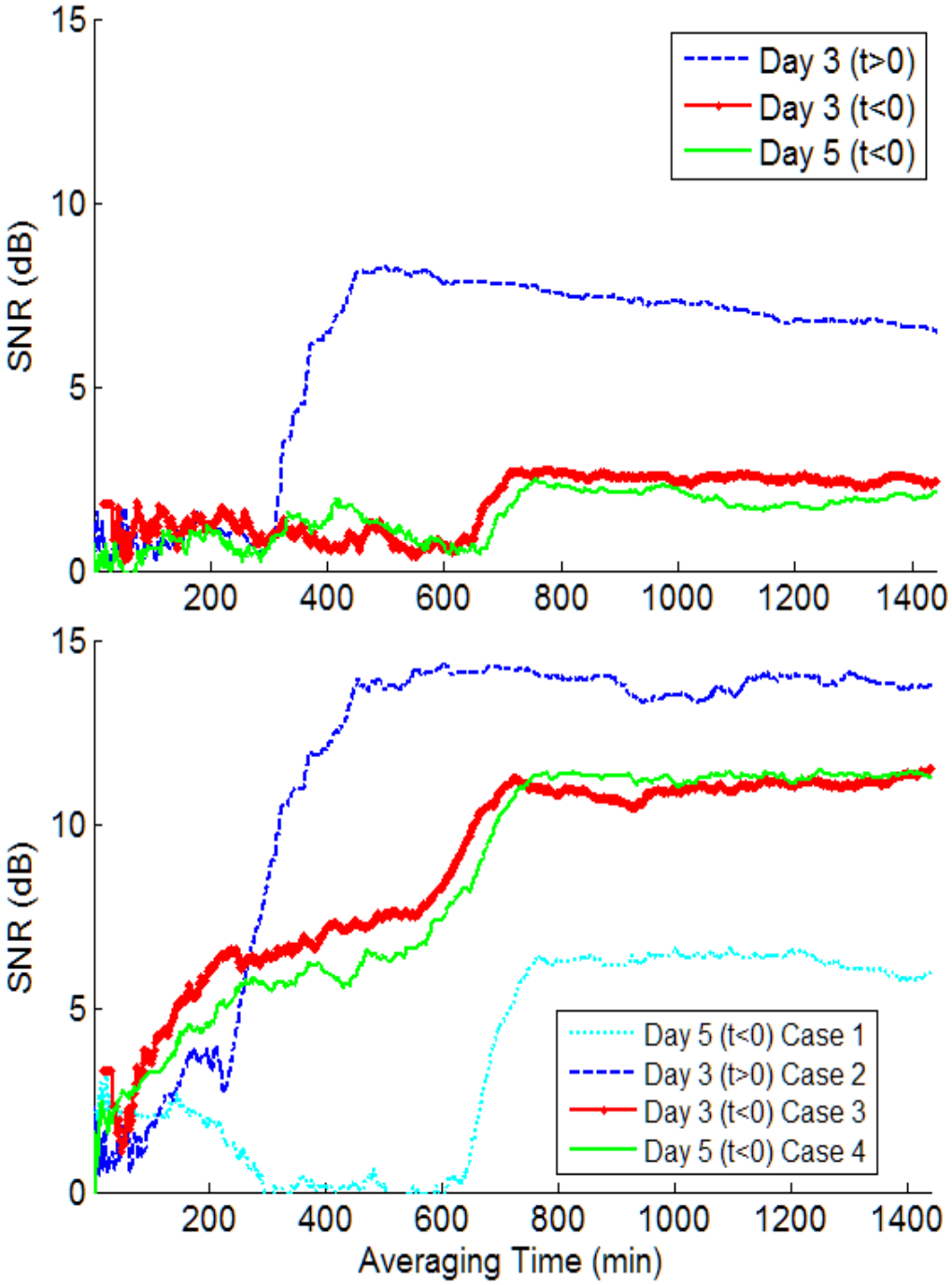

Fig. 4.5 (a) SNR growth of the arrival between the 4th element of VLA1 and the 2nd element on VLA2 over 24 hours for different starting times (b) SNR growth of arrival with spatio-temporal filtering and beamforming for day and sign of arrivals as noted with a reference matrix of Case 1: W_1=W_2=[1, 1, … 1] (plane wave beamforming); Case 2: R(t>0, 3180,160); Case 3: R(t<0, 2881,1920); Case 4: R(t>0, 5761,1920);

4.3.2 Arrival Tracking

To track arrival over six days, the beamformed output for all four cases was calculated with N=480 and L stepping through minute by minute through the six days. From this data, the arrival time and arrival SNR (as defined by Eq. (4.2)) were recorded and plotted in Fig. 4.6. Only arrivals above an SNR of 1.5 dB were used in the plots to show relevant data points for visual trends. Each of the four cases were applied to the six days of data with Fig. 4.6(a) plane wave beamforming with no SVD filtering (Case 1: $W_1=W_2=[1, 1, \ldots 1]$), Fig. 4.6(b) beamforming of positive arrivals using Case 2: R(t>0, 3180,160) as a reference, Fig. 4.6(c) beamforming of the negative arrivals with Case 3: R(t<0, 2881,1920) as a reference, and Fig. 4.6(d) beamforming of the negative arrivals with Case 4: R(t>0, 5761,1920) as a reference. It should be noted that the arrival tracking works best (highest SNR) around the day that is used as the reference as seen with Fig. 4.6(c-d). In addition, the negative tracking days tend to illustrate a smoother arrival tracking as compared the plane wave method alone (Fig. 4.6(a)) or the positive arrivals (Fig. 4.6(a)) which were plagued with discrete loud shipping. From this data it shows that our model (plane wave beamforming), was not an effective reference to use. The better reference is to use a long time average of the data if it is available, but if it is not available, a model can give limited results. Also sometimes, the reference that has the strongest SNR such as the passing of the ship (Fig. 4.4(c)) may not be the best for long time tracking.

Each set of arrivals show a linear trend that suggests a global change of the environment over the six days. The total changes in time over the six days for Fig. 4.6(a-d) are as follows 872us, 1047us, 1226us, and 1272us. These changes in time could be

resultant from one of three reasons. The first is that it is a result of clock drift between the two separate arrays. The second could be that the sound speed is changing (speeding up) by (3.7-5.5m/s over six days). The last possibility is that the arrays are not stationary and slowly drift (1.1-1.6m over six days). Due to both the negative arrivals (Fig. 4.6(a,c,d)) and positive arrivals (Fig. 4.6(c)) having a positive trend, the data suggests that the linear trend is due to clock drift. If the arrivals would have different trends for the different arrivals (one slope positive and the other negative) then that would point to the target area's environment changing whether it be the sound speed or position drifting. The clock drift occurs when the arrays are no longer synchronized by the GPS. Hence when the arrays are deployed at the start of the experiment (Day 1) the arrays no longer have a means to stay synchronized and start to drift. Previous studies on these arrays have estimated the clock drift at 200 μs over 5 days [43]. The change that was observed over the six days ranged from 872 μs to 1272 μs, about five times the expected result.

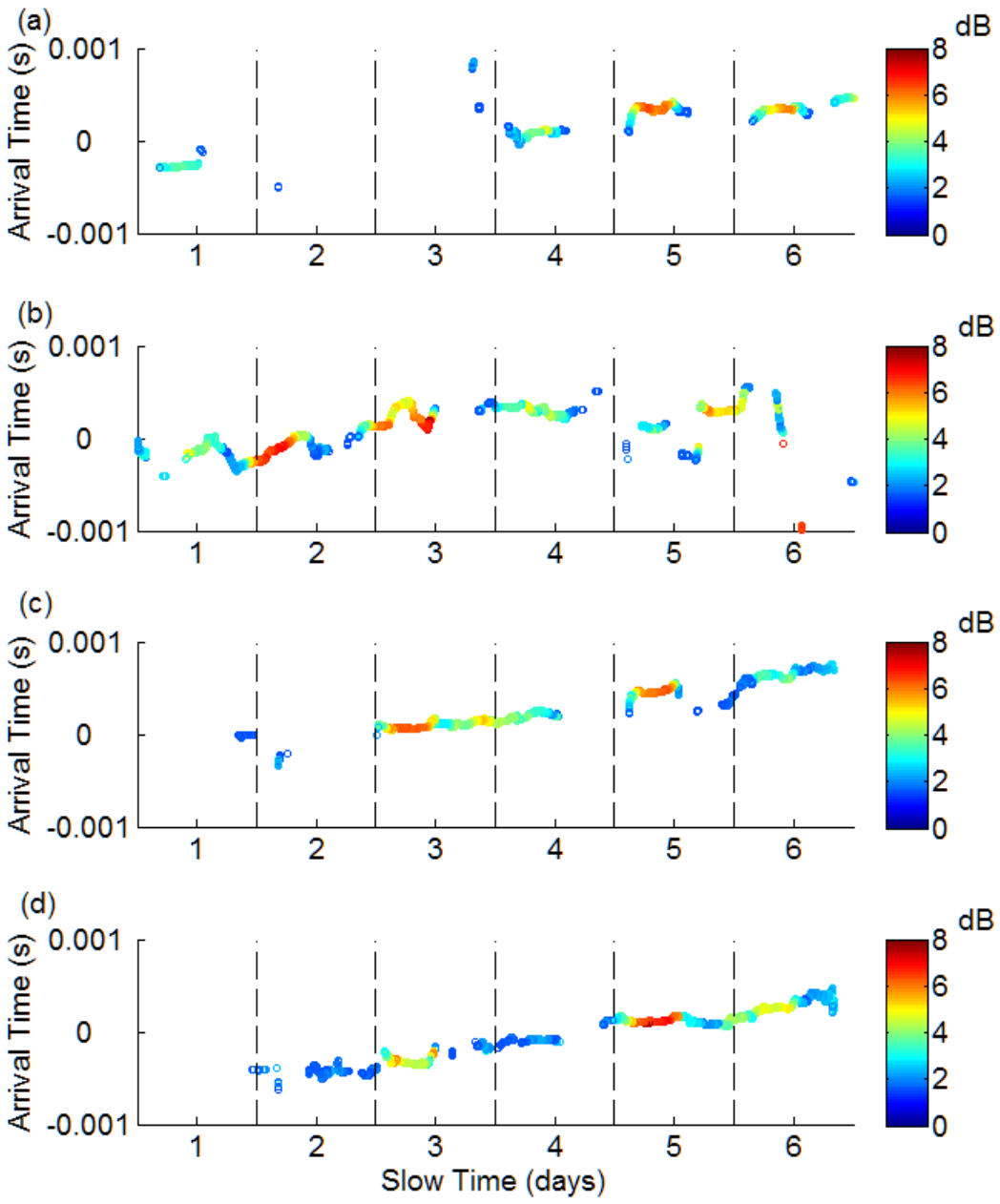

Fig. 4.6 Arrival tracking of beamformed waveform, *B(t,L,N=480)*, with L stepping through each minute over six days with each arrival having its associated SNR denoted by the colorbar in dB for (a) plane wave beamforming with no SVD filtering (Case 1: $W_1=W_2=[1, 1, \ldots 1]$) (b) beamforming of positive arrivals using Case 2: R(t>0, 3180,160) as a reference (c) beamforming of the negative arrivals with Case 3: R(t<0, 2881,1920) as a reference and (d) beamforming of the negative arrivals with Case 4: R(t<0, 5761,1920) as a reference. Only arrivals above 1.5dB are presented in this plot in order to show trustworthy data trends.

4.4 Conclusions

Passive ocean monitoring was done by recording ambient noise for six days off the coast of San Diego, CA on two vertical line arrays. A Green's function approximate was achieved by using the cross-correlation, and the emergence of the arrival SNR was enhanced by the use of a spatio-temporal filter. Clear arrivals (above 1.5dB) could be tracked by using a moving window and reference matrix in an environment that was plagued with discrete shipping events. Methods of mitigating the discrete shipping noise along with selection of the reference matrices are presented and shown to aid in the effective arrival tracking.

CHAPTER 5

CONCLUSION

Two applications of the cross-correlation of ambient noise have been presented. Each method has pros and cons over the more typical active methods. The differences will be presented along with alternative applications of the cross-correlation of ambient noise method.

The first experiment dealt with imaging in the ultrasonic regime. The classic alternative is to the passive recording on CMUTs is the pulse-echo ultrasound. In terms of near field imaging the passive method is ideal, as not only does SNR increase as the target moves closer to the array, the array is not saturated from the pulse that the active system sends out. On the other hand, the passive array can take one minute of averaging noise to obtain an image as opposed to pulse-echo measurements, which only take microseconds. Due to long averaging time of the passive method, medical imaging is not likely the best use of this technology give that the human heart beat about 60 times in a minute. With each heartbeat, the image can be distorted leaving a very unclear image. The controlled experiment allowed the luxury of keeping the environment stationary. A more suitable application area in the ultrasonic regime may be the long term monitoring of structures for non destructive evaluation. Even though the exposure time was much greater than the pulse-echo, both images had comparable SNR.

The second experiment demonstrated the long time ocean monitoring using the cross-correlation of ambient noise. Some current methods of monitoring the ocean include stationary floats, disposable sensors, and others. The main advantage of the

passive method is that the arrays can be deployed for other experimental purposes while the passive ocean monitoring is being tracked. This can allow relatively fast knowledge of how the ocean environment is changing around the target area.

Further work can be done with both of these experiments. As for the ultrasonic imaging, the next steps would be to image more complex target. In addition, the spatial limits of the cross-correlation should be well understood. For example, how close can a target be to the array and how far can a target be and still be detected. Since this method of imaging will likely not be applied in the medical field, other related applications should be examined. As for the ocean monitoring experiment, the next steps would be to extend the experiment. Now that it is known that ocean monitoring is possible over such distances and durations the next step would be to examine longer duration with strict control on array position, clock drift, and target area sound speed. Additionally, longer distances and different frequency ranges can be used.

APPENDIX A

ADDITIONAL ARRIAL TRACKING

The slopes of the arrivals determined that the most likely cause of the arrival change was due to clock drift. This is because both the positive and negative arrivals exhibited a positive time change. Since both arrivals were shifting in the same direction, clock drift is the most likely cause. If the arrivals for the negative time were becoming more negative while the positive arrivals were becoming more positive then this would point to a change in the target area's environment. To ensure that the positive arrivals were becoming more positive as shown in Fig. 4.4(b) another reference was obtained from positive arrivals and used across all six days. The reference used was Day 5 of the experiment for 1920 minutes, R(t>0, 5761,1920), with the results shown in Fig. A.1. As expected the arrivals still have a positive trend, which validates the theory of clock drift. Also of note is the very sporadic tracking that exhibited from the positive arrivals.

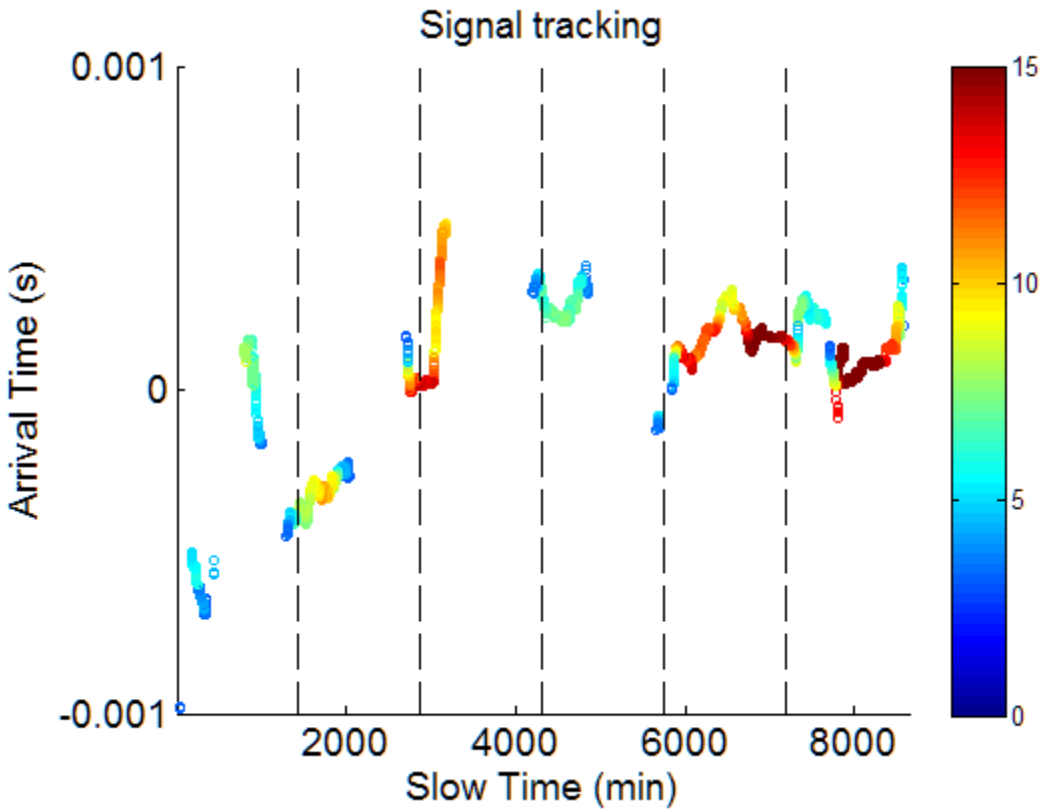

Fig. A.1 Arrival tracking of beamformed waveform, *B(t,L,N=480)*, with L stepping through each minute over 6 days with each arrival having its associated SNR denoted by the colorbar in dB for beamforming of the positive arrivals with R(t>0, 5761,1920) as a reference. Only arrivals above 3dB are presented in this plot in order to show trustworthy data trends.

APPENDIX B

MATLAB CODE

B.1 Cross-Correlation

```
%Take the cross-correlation for 1 minute of data
%Apply frequency whitening if desired
%Apply amplitude clipping if desired

%shown for CMUT ultrasound data

Fe=100e6; %sampling frequency
Ts=1/Fe;
Ntot=100e6;%%set the number of points you want to read-100e6 pts
=1second of data
Nact=Ntot/100e6;
Nstr=num2str(Nact);

for totalcount=1;
    totalcountstr=num2str(totalcount-1);
    freq_int=[21e6 28e6]; %%Selected Frequency band for data processing
    freq_int2=[21e6 28e6];


    FPATH_DATA='C:\CMUT\old\6\';

    COLOR_BIG=['brkmcgy'];
    ELT=[1:4];

    delay=1e-3;Mpoint=floor(delay/Ts);
    Tcorr=[-(Mpoint):(Mpoint)]*Ts; %set you xcorr axis
    Nc=length(Tcorr);
    %freqC=[0:Nc-1]/Nc/Ts;
    IND_ref=[nchoosek([1:4],2)];
    ZZ=zeros(length(Tcorr),length(IND_ref)); %%Save one matrix of
correlation every hour

    INIT=0;
    numofrun=floor(Ntot/5e6);
    for minute=1:numofrun
        %for minute=1
        N=5e6;
        minute;
        freq=[0:N-1]/N/Ts;
        time=[0:N-1]*Ts;
        FF=dir( [FPATH_DATA,'*.bin']);
        GG=[FF.name];
        count=1;ff=0;
        clear FNAME_TOTAL_DATA1;
        while (count<=length(GG))
            ff=ff+1;
            FILE_OK(ff)=1;
```

```
            FNAME_TOTAL_DATA1(ff,:)=GG(1+(ff-1)*10:ff*10);
            count=count+10;
        end;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%READ FILES
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        ff=0;

        %     clear xx
        if INIT==0;
            INIT=1;
            xx=zeros(N,4);
        end
        while ff<=size(FNAME_TOTAL_DATA1,1)-1
            ff=ff+1;
            ff/size(FNAME_TOTAL_DATA1,1);
            cc=0;
            FNAME_TOTAL_DATA1(ff,:);
            file1=[FPATH_DATA, FNAME_TOTAL_DATA1(ff,:)];
            fid1= fopen(file1);%%
            TEMP=fread(fid1,'int16');
            fclose(fid1);
            xx(:,ff)=TEMP((1+(minute-1)*N):N*minute);
            clear TEMP
        end
        % end
        %figure;plot(time(1:100:end),xx(1:100:end,:))

        %%PART II: Pre-processing of  noise
data%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
        %%%%%%%%%%%%Frequency whitening of the noise data in the
selected band
        FREQUENCY_EQUALIZATION='y'; %% select flag if operation desired
or not
        AMPLITUDE_CLIPPING='n';%% select flag if operation desired or
not
        SHARP_FILTERING='n';


        %%^^^^^^^Amplitude cliiping in time domain to remove spikes and
loud events
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
        for rr=1
            if AMPLITUDE_CLIPPING=='y';
                test=squeeze(xx(:,:,rr));
                test=test(:);%%Concatenate all 16 channels into on long
data vector
                %%%%%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^6
                THR_xx=3*std(test); %1200;%%Compute overall standard
deviation on all 16 channels
                %^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
                [I1 J1]=find(abs(test)>THR_xx);%%use different
threshold for each file
                test(I1)=sign(test(I1))*THR_xx;

xx(:,:,rr)=reshape(test,size(squeeze(xx(:,:,rr))));%%put back data in
time*channel matrix format
                clear test
                THR_xxALL(rr)=THR_xx; %%monitor threshold to future
analyis
            end
            NRJall_PreFilter(:,rr)=squeeze(sum(
abs(xx(:,:,rr)).^2,1));%%Compute NRJ of all hydrophone
        end
        %figure;plot(time(1:100:end),xx(1:100:end,:))

%%%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
        %%Use a  stringent filter, above 250hz to eliminate Power
        %            cycling noise issue

        %%%%%%%%%%%%Define Frequency whitening Parameters
        [BB1,AA1]=butter(4,[freq_int]/Fe*2);
        %%%%%%%%%%%%Frequency whitening Parameters
        if (INIT==1) %%set-up parameter for the first file only
            INIT=2;
            deltaf_inf=round(1e6);
            deltaf_sup=round(1e6);
            J=ceil(freq_int(1)*N/Fe):floor(freq_int(2)*N/Fe);%%Define
frequency interval for whitening

Jdebut=ceil(freq_int(1)*N/Fe):floor((freq_int(1)+deltaf_inf)*N/Fe);%rou
nd up the edges to avoid abrupt truncation in the frequency domain
            Jfin=ceil((freq_int(2)-
deltaf_sup)*N/Fe):floor(freq_int(2)*N/Fe);%round up the edges

        end
        Nj=size(xx,2);

        %figure;plot(time(1:100:end),((xx(1:100:end,:))));

        if FREQUENCY_EQUALIZATION=='y';

            rr=1;
            for ff=1:size(xx,2)
                xx(:,ff)=fft(xx(:,ff,rr),[],1);
                TEMP=zeros(size(squeeze(xx(:,ff,rr))));
                TEMP(J)=exp(1i*angle(xx(J,ff,rr)));%Take the phase only
                TEMP(Jdebut)=TEMP(Jdebut).*(sin(pi/2*[0:length(Jdebut)-
1]/(length(Jdebut)-1)).^2).';%%round up the edges
                TEMP(Jfin)=TEMP(Jfin).*(cos(pi/2*[0:length(Jfin)-
1]/(length(Jfin)-1)).^2).';

                %%Back to time-domain only if Amplitude clipping is
used after
                if AMPLITUDE_CLIPPING=='y';
                    TEMP=2*real(ifft(TEMP,[],1));
                end
```

```
                xx(:,ff,rr)=TEMP;
                clear TEMP
            end
        end
        %figure;plot(time(1:100:end),real((xx(1:100:end,:))));


        for rr=1
            if AMPLITUDE_CLIPPING=='y';
                test=squeeze(xx(:,:,rr));
                test=test(:);%%Concatenate all 16 channels into on long
data vector
                %%%%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^6
                THR_xx=3*std(test); %1200;%%Compute overall standard
deviation on all 16 channels
                %^^^^^^^^^^^^^^^^^^^^^^^^^^
                [I1 J1]=find(abs(test)>THR_xx);%%use different
threshold for each file
                test(I1)=sign(test(I1))*THR_xx;

xx(:,:,rr)=reshape(test,size(squeeze(xx(:,:,rr))));%%put back data in
time*channel matrix format
                clear test
                THR_xxALL(rr)=THR_xx; %%monitor threshold to future
analyis
            end
            NRJall_PreFilter(:,rr)=squeeze(sum(
abs(xx(:,:,rr)).^2,1));%%Compute NRJ of all hydrophone
        end



        rr=1;
        for ff=1:size(xx,2)
            NRJall(ff,rr)=squeeze(sum(
abs(xx(:,ff,rr)).^2,1));%%Compute NRJ of all hydrophone
        end

        clear TEMP;

        display('COMPUTE XCORR')
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%Final step%%%%Compute the pairwise CROSS-CORRELATIONS
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        delay=1e-3;Mpoint=floor(delay/Ts);
        Tcorr=[-(Mpoint):(Mpoint)]*Ts;
        Nc=length(Tcorr);
        freqC=[0:Nc-1]/Nc/Ts;
        %IND_ref=nchoosek([1:4],2);


        %for cc=1:length(PAIR);

        for ii=1:length(IND_ref)

            if AMPLITUDE_CLIPPING=='y';
```

```
                xx=fft(xx,[],1);
            end
            %%%%%%%%%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
            %INDelt(cc,1)=1; INDelt(cc,2)=2;
            %%1->2 t<0;; 2->1; t>0
            %%It is the same as
CC=xcorr(DATA_VLA1(:,1),DATA_VLA2(:,2),Mpoint);

%ZZ(:,ii)=xcorr(xx(:,IND_ref(ii,1)),xx(:,IND_ref(ii,2)),Mpoint,'coeff')
;

            %%%%Do correlations ourselves in the frequeny domain- This
is  faster than calling xcor as we alrady have samve the
data%%%%%%%%%%%%%^^^^^^^^^^^^^^^^^^^^^^^^^
            CC=xx(:,IND_ref(ii,1)).*conj(xx(:,IND_ref(ii,2))) ;

CC=real(ifft(CC))*N/sqrt(NRJall(IND_ref(ii,1)))/sqrt(NRJall(IND_ref(ii,
2)));
            CC=cat(1,CC([end-Mpoint+1:end]), CC([1:Mpoint+1]));
            ZZ(:,ii)=CC+ZZ(:,ii);

            %find the time windows that will be used for SNR
calculations
            search=[1.0024e5 1.0029e5];
            searchnoise=[1.005e5 1.008e5];
            %searchleft=[9.975e4 9.985e4];
            searchleft=[9.972e4 9.976e4];
            searchnoiseleft=[.992e5 .995e5];
            maxright=max(ZZ2(search(1):search(2),:));
            maxleft=max(ZZ2(searchleft(1):searchleft(2),:));
            noiseright=std(ZZ2(searchnoise(1):searchnoise(2),:));

noiseleft=std(ZZ2(searchnoiseleft(1):searchnoiseleft(2),:));
            SNRleft=maxleft./noiseleft;
            SNRright=maxright./noiseright;
            SNRleftav(ii)=mean(SNRleft);
            SNRrightav(ii)=mean(SNRright);

        end

    end

    figure;;hold on
    plot(Tcorr,ZZ)
    xlim([-1 1]*45e-6)
    ylim([-1 1 ]*0.08)
    title([FPATH_DATA 'XCor ' Nstr  's'])
    xlabel('time (s)')
    ylabel('XCor')

    [BB2,AA2]=butter(4,[freq_int2]/Fe*2);
    ZZ2=filtfilt(BB2,AA2,ZZ);

    figure;;hold on
    %plot(Tcorr,ZZ(:,3)/max(abs(ZZ(:,3))),'r')
```

```
SCALE1=0.08;
for jj=1:size(ZZ2,2)
    plot(Tcorr,ZZ2(:,jj)/SCALE1+jj,'r')
    xlim([-1 1]*45e-6)
    %
end
ylim([0 size(ZZ2,2)+1 ])
title([FPATH_DATA ' Individual XCor ' Nstr  's'])
xlabel('Time (s)')
ylabel('channel/Xcor')

figure;;hold on
plot(Tcorr,ZZ2)
xlim([0 1]*6e-6)
ylim([-1 1 ]*0.08)
title([FPATH_DATA ' XCor (refiltered) ' Nstr  's'])
xlabel('Time (s)')
ylabel('XCor')


%Can used if needing to find the best band
% Determine optimal bandwidth
Ipos=find(Tcorr>1.8e-6 & Tcorr <2.4e-6);%find arrivals
Ineg=find(Tcorr>1.8e-6 & Tcorr <2.4e-6);
Istd=find(abs(Tcorr)>2e-5 & abs(Tcorr)<5e-5);

clear Noise*, clear SNR*,clear Bw;clear Freq2, clear Freq1
i1=0;i2=0;
for f1=[freq_int(1):1e6:freq_int(2)]
    i1=i1+1;i2=i1;

    for f2=[f1+1e6:1e6:freq_int(2)]
        i2=i2+1;
        freq_int2=[f1 f2];[BB2,AA2]=butter(4,[freq_int2]/Fe*2);
        ZZ2=filtfilt(BB2,AA2,ZZ(:,3));
        NoiseFL(i1,i2)=std(ZZ2(Istd));
        ZZ2=abs(hilbert(ZZ2));
        MMpos=max(ZZ2(Ipos));
        MMneg=max(ZZ2(Ineg));

        Freq2(i2)=f2;
        Freq1(i1)=f1;
        SNRpos(i1,i2)=MMpos/NoiseFL(i1,i2);
        SNRneg(i1,i2)=MMneg/NoiseFL(i1,i2);
        Bw(i1,i2)=f2-f2;

    end
end

figure(13)
subplot(1,2,1)
imagesc(Freq1/1e6,Freq2/1e6,SNRpos')
xlabel('Lower Frequency (Mhz)')
ylabel('Upper Frequency (Mhz)')
axis xy
```

```
    colorbar


    subplot(1,2,2)
    imagesc(Freq1/1e6,Freq2/1e6,SNRneg')
    xlabel('Lower Frequency (Mhz)')
    ylabel('Upper Frequency (Mhz)')
    axis xy
    colorbar

    %save(savename,'Tcorr', 'ZZ', 'ZZ2', 'Fe','freq_int',
'freq_int2','xx')
end
ZZfilt21_28=ZZ;
```

B.2 CMUT Spherical Beamforming

```
%Nearfield Scattering BF for CMUT
%makes an image with the cross correlation waveforms

INIT=1;

savenameBF=['15PTwhiteBFSNRcrossrange'];
peakwhitening='y'; %additional frequency whitening to enhance peaks
SNRcheck='y'; %Only use waveforms that have an acceptable SR
numofsettouse=8;

for setoffour=1:numofsettouse

    totalcountstr=num2str(setoffour-1);
    savename=['15PT' totalcountstr ];
    load(savename)
    SenN=floor(size(Tcorr,2)/2)+1;
    SensorData=ZZ2((SenN):end,:); %take only positive arrivals
    freq=((1:SenN)-1)*Fe/SenN; %freq axis
    timeaxis=((1:SenN)-1)/Fe;
    if peakwhitening=='y'
        xx=zeros(size(SensorData));
        xx(0.0016e5:0.0022e5,:)=SensorData(0.0016e5:0.0022e5,:);

        [BB1,AA1]=butter(4,[freq_int]/Fe*2);
        %%%%%%%%%%%%Frequency whitening Parameters
        if (INIT==1) %%set-up parameter for the first file only
            INIT=2;
            deltaf_inf=round(1e6);
            deltaf_sup=round(1e6);
            [I,J]=find(freq>freq_int(1) & freq<freq_int(2));%%Define
frequency interval for whitening
            [Idebut,Jdebut]=find(freq>freq_int(1) &
freq<(freq_int(1)+deltaf_inf));%round up the edges to avoid abrupt
truncation in the frequency domain
            [Ifin,Jfin]=find(freq>(freq_int(2)-deltaf_sup) &
freq<freq_int(2));%round up the edges
            fftsignal=zeros(size(SensorData),numofsettouse);
        end

        for ff=1:size(xx,2)
            xx(:,ff)=fft(xx(:,ff),[],1);
            TEMP=zeros(size(squeeze(xx(:,ff))));
            TEMP(J)=exp(1i*angle(xx(J,ff)));%Take the phase only
            TEMP(Jdebut)=TEMP(Jdebut).*(sin(pi/2*[0:length(Jdebut)-
1]/(length(Jdebut)-1)).^2).';%%round up the edges
            TEMP(Jfin)=TEMP(Jfin).*(cos(pi/2*[0:length(Jfin)-
1]/(length(Jfin)-1)).^2).';
            fftsignal(:,ff,setoffour)=TEMP;
        end
    else
        fftsignal=fft(SensorData,[],1);
```

```
fftsignal(:,:,setoffour)=[fftsignal(1:floor(SenN/2),:);zeros(floor(SenN
/2+1):SenN,size(fftsignal,2))];
    end

    %it is very important to have the accurate locations of the sensors
for
    %Beamfomring
    numSen=4;
    choose=nchoosek(1:numSen,2);
    ang=linspace(0,2*pi,numSen+1)+ones(1,numSen+1)*(setoffour-
1)*90/8*pi/180;
    sensorPos(:,:,setoffour)=[362.5e-6*cos(ang(1:(end-1))); 362.5e-
6*sin(ang(1:(end-1))) ; zeros(1,numSen)];

end
%end
clear ZZ ZZ2 xx

%Set the parameters for the Spherical Beamforming
fs=Fe;
c=1500;
dt=1/fs;
xlow=-.5e-3;
xhigh=.5e-3;
xspacing=1e-5;
ylow=-.5e-3;
yhigh=.5e-3;
yspacing=1e-5;
zlow=1.3e-3;
zhigh=1.3e-3;
zspacing=1e-5;
xaxis=xlow:xspacing:xhigh;
yaxis=ylow:yspacing:yhigh;
zaxis=zlow:zspacing:zhigh;

maxm=zeros(length(xlow:xspacing:xhigh),length(ylow:yspacing:yhigh),leng
th(zlow:zspacing:zhigh));
%maxn=maxm;maxo=maxm;maxp=maxo;

%Spherical Beamforming
%Step through all pixels desired and obtain a value
for x=xlow:xspacing:xhigh
    x1=round(x/xspacing+1-xlow/xspacing);
    for y=ylow:yspacing:yhigh
        y1=round(y/yspacing+1-ylow/yspacing);
        for z=zlow:zspacing:zhigh
            z1=round(z/zspacing+1-zlow/zspacing);
            d=[x;y;z];
            setoffoursig=0;
            for setoffour=1:numofsettouse
                %for setoffour=countsetoffour
                allsensors=0;
                for count=1:numSen
                    clear Ind J
                    [Ind J]=find(choose==count);
```

```
                    fullSignal=0;
                    clear sig
                    for ind=1:(numSen-1)
                        if SNRcheck=='y'

SensorDatanew=real(ifft(fftsignal(:,Ind(ind),setoffour)));
                            xx=zeros(size(SensorDatanew));

xx(0.0016e5:0.0022e5,:)=SensorDatanew(0.0016e5:0.0022e5);
                            maxsig=max(abs(xx));
                            noisepart=SensorDatanew(0.0020e5:0.0030e5);
                            noisestd=std(noisepart);
                            SNRsig=maxsig./noisestd;
                        else
                            SNRsig=Inf;
                        end
                        if SNRsig<5.5
                            continue
                        end
                        rone=sensorPos(:,choose(Ind(ind),1),setoffour);
                        rtwo=sensorPos(:,choose(Ind(ind),2),setoffour);
                        delta=norm(d-rone)/c+norm(d-rtwo)/c;

tempfftsig=squeeze(fftsignal(:,Ind(ind),setoffour));
                        sig=2*pi*real(ifft(tempfftsig.*exp(-
1j*2*pi*freq*delta)'));
                        fullSignal=fullSignal+sig;
                    end

%figure(1);subplot(2,2,count);plot(time,sig);xlim([0 8e-8])

%figure(2);subplot(2,2,count);plot(time,fullSignal);xlim([0 8e-8])
                    allsensors=allsensors+fullSignal;
                end
                setoffoursig=setoffoursig+allsensors;
            end
            placehold=abs(hilbert(setoffoursig));
            maxm(x1,y1,z1)=placehold(1);

        end
    end
end
save(savenameBF,'maxm', 'xaxis', 'yaxis', 'zaxis')
%Plot the resulting image
figure
maxmsq=abs(squeeze(maxm));
imagesc(yaxis,xaxis,20*log10(maxmsq./(max(max(maxmsq))*ones(size(maxmsq
)))))
colorbar
caxis([-10 0])
```

B.3 Ocean Monitoring

```
%Ocean Monitoring Code
%Takes the inputs of R(t,L,N) and a refernce matrix
%Uses the reference matrix with SVD and tracks arrivals

function savename=BFprojFinenew(days,refday)

%load reference matrix
refdaystr=num2str(refday);
refdaytwo=refday+1;
refdaytwostr=num2str(refdaytwo);
if refday==99

loadname=['/home/slani/.matlab/MATLAB/NC09/Dayref/dayALLnT01ref.mat']
elseif refday==88

loadname=['/home/slani/.matlab/MATLAB/NC09/Dayref/BestSNRGrowthRefPosRE
F.mat']
else
    loadname=['/home/slani/.matlab/MATLAB/NC09/Dayref/day' refdaystr
refdaytwostr 'p0to7T01ref500.mat']
end
load(loadname)

%choose parameters to be used
thresholdcheck='n'; %remove minuts above threshold
window='y';         %window around the expected arrival
runningaverage='n'; %used to get plots for SNR growth over one day
arrival='p';        %p=positive n=negative
planewaverun='n';   %use unitary weights for plane wave BF

%load the data to be processed
INIT=1;
for day=days
    daytwo=day+1;
    daystr=num2str(day);
    daytwostr=num2str(daytwo);

    ZZavfull=zeros(1000,16,16,1920);
    load(['/mnt/sdb1/NC09DayXcor/day' daystr 'p0to7.mat'])
    ZZavfull(:,:,:,1:480)=ZZavp;
    load(['/mnt/sdb1/NC09DayXcor/day' daystr 'p8to15.mat'])
    ZZavfull(:,:,:,481:960)=ZZavp;
    load(['/mnt/sdb1/NC09DayXcor/day' daystr 'p16to23.mat'])
    ZZavfull(:,:,:,961:1440)=ZZavp;
    load(['/mnt/sdb1/NC09DayXcor/day' daytwostr 'p0to7.mat'])
    ZZavfull(:,:,:,1441:1920)=ZZavp;
    clear ZZavp
    ZZavp=ZZavfull;
    clear ZZavfull

    if arrival=='p'
        timecorr=time_corrp;
        tmin=.295;
```

```
        tmax=.306;
    elseif arrival=='n'
        timecorr=time_corrm;
        tmin=-.303;
        tmax=-.298;
    end

    N=length(timecorr); %number of points
    Fs=10000;
    freqaxis=(0:(N-1))*Fs/N;
    Ts=1/Fs;
    Tfine=timecorr(1):Ts/100:timecorr(end); % define time for
interpolation
    INDfreq=find(freqaxis>250 & freqaxis<1500);
    savename=['BFday' daystr daytwostr 'ref' refdaystr refdaytwostr
'p0to7tightwinNoT_check'];

    %%%%%%%% Set up the averaging variables
    if runningaverage=='y'
        N_dur=1;
    else
        N_dur=480;
    end         %num of minutes to average
    totalmin=size(ZZavp,4);
    ll=totalmin-N_dur+1;
    N_sum=ll; %nombre total de sommation que je veux atteindre
    %%%%%%%%%
    % threshold checking
    if thresholdcheck=='y'
        threshold=.01;
        for totalcount=1:totalmin
            maxcheck=max(squeeze(ZZavp(:,1,1,totalcount)));
            if maxcheck>=threshold
                ZZavp(:,:,:,totalcount)=zeros(1000,16,16);
            end
        end
    end
    %gather for SNR calcultions
    ZZav=zeros(N,N_sum); ZZavsq=zeros(N,N_sum);
    for pp=1:ll
        if runningaverage=='y'
            ZZav(:,pp)=sum(ZZavp(:,4,2,1:pp),4);
            ZZavsq(:,pp)=sum(ZZavp(:,4,2,1:pp).^2,4);
        else
            ZZav(:,pp)=sum(ZZavp(:,4,2,pp:N_dur+pp-1),4);
            ZZavsq(:,pp)=sum(ZZavp(:,4,2,pp:N_dur+pp-1).^2,4);
        end
    end

    %first window around expected arrival
    if window=='n'
        ZZavphw=ZZavp;
    else
        IND1=find(timecorr>tmin & timecorr<tmax);
        ZZavphw=zeros(size(ZZavp));
        for ii=1:16
```

```
            for jj=1:16
                for mincount=1:totalmin

%ZZavphw(IND1,ii,jj,mincount)=ZZavp(IND1,ii,jj,mincount).*hann(length(I
ND1));

ZZavphw(IND1,ii,jj,mincount)=ZZavp(IND1,ii,jj,mincount);
                end
            end
        end
    end
    clear ZZavp
    ZZavphw_f=fft(ZZavphw);
    clear ZZavphw

    %SVD INIT
    Uref=zeros(N,16,16);
    Vref=zeros(N,16,16);
    Sref=zeros(N,16,16);
    ZZav_sum_f_proj_PR=zeros(N,16,16); ZZav_sum_f_proj_BF=zeros(N,1);
    ZZav_sum_t_proj_PR_all=zeros(N,16,16,totalmin);
ZZav_sum_t_proj_BF_all=zeros(N,totalmin);
    %SVD filter waveforms and bring back to time domain
    for stepwise=1:totalmin
        ZZav_sum_f=ZZavphw_f(:,:,:,stepwise);
        %%Projecting
        for ii=1:N
            if INIT==1
                [U,S,V] = svd(squeeze(ZZav_ref_f(ii,:,:)));
                Uref(ii,:,:)=U;
                Vref(ii,:,:)=V;
                Sref(ii,:,:)=S;
            end
            %first singular vectors
            Un=Uref(ii,:,1).';
            Vn=Vref(ii,:,1).';
            if planewaverun=='y'
                Un=ones(size(Un));
                Vn=ones(size(Vn));
            end
            P1=Un*Un';
            Q1=Vn*Vn';

ZZav_sum_f_proj_PR(ii,:,:)=P1*squeeze(ZZav_sum_f(ii,:,:))*Q1; %PR
matrix

ZZav_sum_f_proj_BF(ii,1)=Un'*squeeze(ZZav_sum_f(ii,:,:))*Vn; %BF matrix
        end
        INIT=INIT+1;

ZZav_sum_t_proj_PR_all(:,:,:,stepwise)=real(ifft(ZZav_sum_f_proj_PR,[],
1));

ZZav_sum_t_proj_BF_all(:,stepwise)=real(fftshift(ifft(ZZav_sum_f_proj_B
F,[],1),1));
    end
    clear ZZav_sum_f_proj_PR ZZav_sum_f_proj_BF ZZavphw_f
```

```
    %Sum both the PR data and the BF data
    %ZZavPR=zeros(N,16,16,N_sum); ZZavPRsq=zeros(N,16,16,N_sum);
    ZZavPR=zeros(N,N_sum); ZZavPRsq=zeros(N,N_sum);
    ZZavBF=zeros(N,N_sum); ZZavBFsq=zeros(N,N_sum);
    for pp=1:ll
        if runningaverage=='y'
            ZZavPR(:,pp)=sum(ZZav_sum_t_proj_PR_all(:,4,2,1:pp),4);

ZZavPRsq(:,pp)=sum(ZZav_sum_t_proj_PR_all(:,4,2,1:pp).^2,4);
            ZZavBF(:,pp)=sum(ZZav_sum_t_proj_BF_all(:,1:pp),2);
            ZZavBFsq(:,pp)=sum(ZZav_sum_t_proj_BF_all(:,1:pp).^2,2);
        else
            ZZavPR(:,pp)=sum(ZZav_sum_t_proj_PR_all(:,4,2,pp:N_dur+pp-
1),4);

ZZavPRsq(:,pp)=sum(ZZav_sum_t_proj_PR_all(:,4,2,pp:N_dur+pp-1).^2,4);
            ZZavBF(:,pp)=sum(ZZav_sum_t_proj_BF_all(:,pp:N_dur+pp-
1),2);
            ZZavBFsq(:,pp)=sum(ZZav_sum_t_proj_BF_all(:,pp:N_dur+pp-
1).^2,2);
        end
    end
    clear ZZav_sum_t_proj_PR_all ZZav_sum_t_proj_BF_all

    %Initialization for figures
    Corr_nonp_rec=zeros(N,N_sum);
    Corr_p_rec_PR=zeros(N,N_sum); Corr_p_rec_PR2=zeros(N,N_sum);
Corr_p_rec_BF=zeros(N,N_sum); Corr_p_rec_BF2=zeros(N,N_sum);
    noise1 =zeros(1,N_sum);
    noise2_PR=zeros(1,N_sum); noise2_PR2=zeros(1,N_sum);
noise2_BF=zeros(1,N_sum); noise2_BF2=zeros(1,N_sum);
    signal1=zeros(1,N_sum);
    signal2_PR=zeros(1,N_sum); signal2_PR2=zeros(1,N_sum);
signal2_BF=zeros(1,N_sum); signal2_BF2=zeros(1,N_sum);
    SNR_nonp=zeros(1,N_sum);
    SNR_p_PR=zeros(1,N_sum); SNR_p_PR2=zeros(1,N_sum);
SNR_p_BF=zeros(1,N_sum); SNR_p_BF2=zeros(1,N_sum);
    maximum1=zeros(1,N_sum);
    maximum2_PR=zeros(1,N_sum); maximum2_PR2=zeros(1,N_sum);
maximum2_BF=zeros(1,N_sum); maximum2_BF2=zeros(1,N_sum);
    signal1_fine=zeros(1,N_sum);
    signal2_fine_PR=zeros(1,N_sum); signal2_fine_PR2=zeros(1,N_sum);
signal2_fine_BF=zeros(1,N_sum); signal2_fine_BF2=zeros(1,N_sum);
    maximum1_fine=zeros(1,N_sum);
    maximum2_fine_PR=zeros(1,N_sum); maximum2_fine_PR2=zeros(1,N_sum);
maximum2_fine_BF=zeros(1,N_sum); maximum2_fine_BF2=zeros(1,N_sum);



    if arrival=='p'
        INDnoise=find(timecorr>0.35 & timecorr<0.38);
    elseif arrival=='n'
        INDnoise=find(timecorr>-.38 & timecorr<-.35);
    end

    %interval where the maximum of beamforming is seeked
    dt=Ts;
```

```
    Tcorr_BF=(1:N)*dt-dt*N/2;
    TfineBF=Tcorr_BF(1):Ts/100:Tcorr_BF(end);
    INDnoiseBF=find(Tcorr_BF>0.03 & Tcorr_BF<0.05);
    tBF_min=-.01;
    tBF_max=.01;
    tPR_min=tmin;
    tPR_max=tmax;
    %interval where the maximum of raw matrix is seeked
    IND1=find(timecorr>tmin & timecorr<tmax);
    %interval where the maximum of projected matrix is seeked
    IND2_PR=find(timecorr>tPR_min & timecorr<tPR_max);
    %interval where the maximum of beamformed matrix is seeked
    IND2_BF=find(Tcorr_BF>tBF_min & Tcorr_BF<tBF_max);
    %fine interval where the maximum of raw matrix is seeked
    IND1_fine=find(Tfine>tmin & Tfine<tmax);
    %fine interval where the maximum of projected matrix is seeked
    IND2_fine_PR=find(Tfine>tPR_min & Tfine<tPR_max);
    %fine interval where the maximum of beamformed matrix is seeked
    IND2_fine_BF=find(TfineBF>tBF_min & TfineBF<tBF_max);

    %get SNR for each method
    newSNRStd=zeros(N,N_sum); newSNRPR=zeros(N,N_sum);
newSNRBF=zeros(N,N_sum);
    for n=1:ll

        %SNR for sensor pair 4-2
        Corr_nonp_rec(:,n)=abs(hilbert(squeeze(ZZav(:,n))));
        [signal1(n),maximum1(n)]=max(Corr_nonp_rec(:,n));
        noise1(n)=std(squeeze(ZZav(INDnoise,n)));

Corr_nonp_rec_fine=abs(spline(timecorr,Corr_nonp_rec(:,n),Tfine));
        [signal1_fine(n),maximum1_fine(n)]=max(Corr_nonp_rec_fine);
        SNR_nonp(n)=signal1(n)/noise1(n);

        %new SNR for sensor pair 4-2
        meanSqStd=((1/N_dur)*ZZav(:,n)).^2;
        avEnStd=(1/N_dur)*ZZavsq(:,n);
        varStd=N_dur*(avEnStd-meanSqStd);
        newSNRStd(:,n)=abs(hilbert(ZZav(:,n))./(varStd.^(0.5));

        %PR
        Corr_p_rec_PR(:,n)=abs(hilbert(squeeze(ZZavPR(:,n))));
        [signal2_PR(n),maximum2_PR(n)]=max(Corr_p_rec_PR(:,n));
        noise2_PR(n)=std(squeeze(ZZavPR(INDnoise,n)));

Corr_p_rec_fine_PR=abs(spline(timecorr,Corr_p_rec_PR(:,n),Tfine));

[signal2_fine_PR(n),maximum2_fine_PR(n)]=max(Corr_p_rec_fine_PR);
        SNR_p_PR(n)=signal2_PR(n)/noise2_PR(n);

        %new SNR for sensor pair 4-2
        meanSqPR=((1/N_dur)*ZZavPR(:,n)).^2;
        avEnPR=(1/N_dur)*ZZavPRsq(:,n);
        varPR=N_dur*(avEnPR-meanSqPR);
        newSNRPR(:,n)=abs(hilbert(ZZavPR(:,n))./(varPR.^(0.5));
```

```
        %BF
        Corr_p_rec_BF(:,n)=abs(hilbert(squeeze(ZZavBF(:,n))));
        [signal2_BF(n),maximum2_BF(n)]=max(Corr_p_rec_BF(:,n));
        noise2_BF(n)=std(ZZavBF(INDnoiseBF,n));

Corr_p_rec_fine_BF=abs(spline(Tcorr_BF,Corr_p_rec_BF(:,n),TfineBF));

[signal2_fine_BF(n),maximum2_fine_BF(n)]=max(Corr_p_rec_fine_BF);%Temp
        SNR_p_BF(n)=signal2_BF(n)/noise2_BF(n);

        %new SNR
        meanSqBF=((1/N_dur)*ZZavBF(:,n)).^2;
        avEnBF=(1/N_dur)*ZZavBFsq(:,n);
        varBF=N_dur*(avEnBF-meanSqBF);
        newSNRBF(:,n)=abs(hilbert(ZZavBF(:,n)))./(varBF.^(0.5));
    end

    %%%%%%%%%%%% FIGURES

    %prep data

    tmin_aff1=tmin;
    tmax_aff1=tmax;
    tmin_aff_PR=tmin;
    tmax_aff_PR=tmax;
    tmin_aff_BF=-.01;
    tmax_aff_BF=.01;

    Corr_nonp=abs(Corr_nonp_rec(:,1:n)');
    IND_plot1=find(timecorr>tmin_aff1 & timecorr<tmax_aff1);
    max_plot1=max(max(Corr_nonp));

    Corr_p_PR=abs(Corr_p_rec_PR(:,1:n)');
    IND_plot_PR=find(timecorr>tmin_aff_PR & timecorr<tmax_aff_PR);
    max_plot_PR=max(max(Corr_p_PR));

    Corr_p_BF=abs(Corr_p_rec_BF(:,1:n)');
    IND_plot_BF=find(Tcorr_BF>tmin_aff_BF & Tcorr_BF<tmax_aff_BF);
    max_plot_BF=max(max(Corr_p_BF));

    croix=10;

    %Ref matrix
    figure;
    subplot(1,2,1)
    imagesc(timecorr,1:n,20*log10(Corr_nonp/max_plot1)); hold on;
    for gg=1:croix:N_sum
        plot(timecorr(maximum1(gg)),gg,'black+') %% rejoute une croix
sur le peak
        title('MA / Standard','Fontsize',20)
    end
    %xlim([tmin_aff1 tmax_aff1]);
    xlabel('Fast Time (s)','Fontsize',20); ylabel('Slow Time
(min)','Fontsize',20);
```

```
    caxis([-40 0]);
    colorbar
    set(gca,'Fontsize',20)

    subplot(1,2,2)
    plot(20*log10(SNR_nonp(1:n)),'b--','linewidth',2);
    xlabel('Slow Time (min)','Fontsize',20); ylabel('SNR
(dB)','Fontsize',20);
    set(gca,'Fontsize',20)
    xlim([1 n])


    %%PR
    figure;
    subplot(2,2,1)
    imagesc(timecorr,1:n,20*log10(Corr_nonp/max_plot1)); hold on;
    for gg=1:croix:N_sum
        plot(timecorr(maximum1(gg)),gg,'black+') %% rejoute une croix
sur le peak
        title('MA / Standard','Fontsize',20)
    end
    %xlim([tmin_aff1 tmax_aff1]);
    xlabel('Fast Time (s)','Fontsize',20); ylabel('Slow Time
(min)','Fontsize',20);
    caxis([-40 0]);
    colorbar
    set(gca,'Fontsize',20)

    subplot(2,2,2)
    imagesc(timecorr,1:n,20*log10(Corr_p_PR/max_plot_PR)); hold on;
    for gg=1:croix:N_sum
        %plot(timecorr(IND2_PR(maximum2_PR(gg))),gg,'black+') %%
rejoute une croix sur le peak
        plot(timecorr(maximum2_PR(gg)),gg,'black+')
    end
    title('MA / SVD Projection','Fontsize',20)
    %xlim([tmin_aff_PR tmax_aff_PR]);
    xlabel('Fast Time (s)','Fontsize',20); ylabel('Slow Time
(min)','Fontsize',20); caxis([-60 0])
    colorbar
    set(gca,'Fontsize',20)

    subplot(2,2,3)
    plot(Tfine(maximum1_fine(1:n)),'b.','linewidth',2); hold on;

%plot(Tfine(IND2_fine_PR(maximum2_fine_PR(1:n))),'r.','linewidth',2)
    plot(Tfine(maximum2_fine_PR(1:n)),'r.','linewidth',2)
    title('Signal tracking','Fontsize',25)
    legend('Standard','SVD Projected')
    %ylim([tPR_min tPR_max])
    xlabel('Slow Time (min)','Fontsize',20); ylabel('Arrival Time
(s)','Fontsize',20);
    set(gca,'Fontsize',20);

    subplot(2,2,4)
    plot(20*log10(SNR_nonp(1:n)),'b.','linewidth',2);
```

```
    hold on;
    SNRstdpts=ceil(maximum1_fine(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRStd(SNRstdpts(jj),jj);
    end
    plot(20*log10(plotSNR),'y.','linewidth',2)
    plot(20*log10(SNR_p_PR(1:n)),'r.','linewidth',2);
    SNRPRpts=ceil(maximum2_fine_PR(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRPR(SNRPRpts(jj),jj);
    end
    plot(20*log10(plotSNR),'g.','linewidth',2)
    legend('Standard','Standard new','SVD Projected','SVD new')
    % title('Compared SNR (dB)','Fontsize',25);
    %ylim([0 35]);
    xlabel('Slow Time (min)','Fontsize',20); ylabel('SNR
(dB)','Fontsize',20);
    set(gca,'Fontsize',20)

    %%BF
    figure;
    subplot(2,2,1)
    imagesc(timecorr,1:n,20*log10(Corr_nonp/max_plot1)); hold on;
    for gg=1:croix:N_sum
        plot(timecorr(maximum1(gg)),gg,'black+') %% rejoute une croix
sur le peak
    end
    title('MA / Standard','Fontsize',20)
    %xlim([tmin_aff1 tmax_aff1]);
    xlabel('Fast Time (s)','Fontsize',20); ylabel('Slow Time
(min)','Fontsize',20);
    caxis([-60 0]);
    colorbar
    set(gca,'Fontsize',20)

    subplot(2,2,2)
    imagesc(Tcorr_BF,1:n,20*log10(Corr_p_BF/max_plot_BF)); hold on;
    for gg=1:croix:N_sum
        %plot(timecorr(IND2_BF(maximum2_BF(gg))),gg,'black+') %%
rejoute une croix sur le peak
        plot(Tcorr_BF(maximum2_BF(gg)),gg,'black+') %% rejoute une
croix sur le peak
    end
    title('MA / SVD Beamforming','Fontsize',25)
    %xlim([tmin_aff_BF tmax_aff_BF]);
    xlabel('Fast Time (s)','Fontsize',20); ylabel('Slow Time
(min)','Fontsize',20); caxis([-60 0])
    colorbar
    set(gca,'Fontsize',20)

    subplot(2,2,3)
    plot(Tfine(maximum1_fine(1:n)),'b.','linewidth',2); hold on;

%plot(Tfine(IND2_fine_BF(maximum2_fine_BF(1:n))),'r.','linewidth',2)

plot(TfineBF(maximum2_fine_BF(1:n))+(tmax+tmin)/2,'r.','linewidth',2)
```

```
    title('Signal tracking','Fontsize',25)
    legend('Standard','SVD Beamforming')
    xlabel('Slow Time (min)','Fontsize',20); ylabel('Arrival Time
(s)','Fontsize',20);
    set(gca,'Fontsize',20);

    subplot(2,2,4)
    plot(20*log10(SNR_nonp(1:n)),'b.','linewidth',2);
    hold on;
    SNRstdpts=ceil(maximum1_fine(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRStd(SNRstdpts(jj),jj);
    end
    plot(20*log10(plotSNR),'y.','linewidth',2)
    plot(20*log10(SNR_p_BF(1:n)),'r.','linewidth',2);
    SNRBFpts=ceil(maximum2_fine_BF(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRBF(SNRBFpts(jj),jj);
    end
    plot(20*log10(plotSNR),'k.','linewidth',2)
    legend('Standard','Standard new','SVD Beamforming','BF new')
    title('Compared SNR (dB)','Fontsize',25);
    %ylim([4 35]);
    xlabel('Slow Time (min)','Fontsize',20); ylabel('SNR
(dB)','Fontsize',20);
    set(gca,'Fontsize',20)

    figure
    plot(20*log10(SNR_nonp),'b--','linewidth',2)
    hold on;
    SNRstdpts=ceil(maximum1_fine(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRStd(SNRstdpts(jj),jj);
    end
    plot(20*log10(plotSNR),'y.','linewidth',2)
    plot(20*log10(SNR_p_PR(1:n)),'g.','linewidth',2);
    SNRPRpts=ceil(maximum2_fine_PR(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRPR(SNRPRpts(jj),jj);
    end
    plot(20*log10(plotSNR),'g.','linewidth',2)
    plot(20*log10(SNR_p_BF(1:n)),'r.','linewidth',2);
    SNRBFpts=ceil(maximum2_fine_BF(1:n)./100);
    for jj=1:length(SNRstdpts)
        plotSNR(jj)=newSNRBF(SNRBFpts(jj),jj);
    end
    plot(20*log10(plotSNR),'k.','linewidth',2)
    legend('std','newstd','PR','newpr','BR','newbf')
    title('SNR comp')
    xlabel('slowtime')
    xlim([1 n])

    save(
savename,'tmax','tmin','timecorr','Tcorr_BF','n','N_sum','Tfine','Tfine
BF',...
```

```
'Corr_nonp','maximum1','maximum1_fine','SNR_nonp','max_plot1','tmin_aff
1','tmax_aff1','IND1','IND1_fine',...

'Corr_p_PR','maximum2_PR','maximum2_fine_PR','SNR_p_PR','max_plot_PR','
tmin_aff_PR','tmax_aff_PR','IND2_PR','IND2_fine_PR','tPR_min','tPR_max'
,...

'Corr_p_BF','maximum2_BF','maximum2_fine_BF','SNR_p_BF','max_plot_BF','
tmin_aff_BF','tmax_aff_BF','IND2_BF','IND2_fine_BF','tBF_min',
'tBF_max',...
        'newSNRStd','newSNRPR','newSNRBF')

    clear tmax tmin timecorr Tcorr_BF n N_sum Tfine TfineBF ...
        Corr_nonp maximum1 maximum1_fine SNR_nonp max_plot1 tmin_aff1
tmax_aff1 IND1 IND1_fine ...
        Corr_p_PR maximum2_PR maximum2_fine_PR SNR_p_PR max_plot_PR
tmin_aff_PR tmax_aff_PR IND2_PR IND2_fine_PR tPR_min tPR_max ...
        Corr_p_BF maximum2_BF maximum2_fine_BF SNR_p_BF max_plot_BF
tmin_aff_BF tmax_aff_BF IND2_BF IND2_fine_BF tBF_min  tBF_max ...
        newSNRStd newSNRPR newSNRBF

    %'Corr_p_PR2','maximum2_PR2','maximum2_fine_PR2','SNR_p_PR2'
    %'Corr_p_BF2','maximum2_BF2','maximum2_fine_BF2','SNR_p_BF2',
end
```

REFERENCES

[1] Anderson, S., "Space-time-frequency processing from the analysis of bistatic scattering for simple underwater targets," Master's Thesis Georgia Institute of Technology, 2012.

[2] Bendat, J. S. and Piersol, A. G. *Random Data: Analysis & Measurement Procedures* pp. 272-286, 357-359 (Wiley-Interscience, 2000).

[3] Blakeslee, S. "Lost on Earth: Wealth of Data Found in Space," The New York Times (Online) Pub. Mar. 20, 1990 Accessed Sept. 22, 2012.

[4] Brekhovskikh, L. M. and Lysanov, Y. P., *Fundamentals of Ocean Acoustics.* Springer, pp. 1-34, 2003.

[5] Brown, R., "A brief account of microscopical observations made in the months of June, July, and August, 1827 on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies," *Philos. Mag.*, vol. 4, pp. 161–173, 1828.

[6] Campillo, M., and Paul, A., "Long-range correlations in the diffuse seismic coda," Science, 299, 547–549, 2003.

[7] Cohen, L., "History of Noise" IEEE Signal Processing Magazine, no. 1053-5888 pp 20-45, 2005

[8] Degertekin, F., Guldiken, R., and Karaman, M., "Annular-ring CMUT arrays for forward-looking IVUS: transducer characterization and imaging," Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on, vol. 53, no. 2, pp. 474 –482, Feb. 2006.

[9] Duroux, A., Sabra, K., Ayers, J., and Ruzzene, M., "Extracting guided waves from cross-correlations of elastic diffuse fields: Applications to remote structural health monitoring," J. Acoust. Soc. Am. 127, 204-215, 2010.

[10] Eccardt, P. C., Lohfink, A., and Garssen, H. G., "Analysis of crosstalk between fluid coupled CMUT membranes," in Ultrasonics Symposium, 2005 IEEE, vol. 1, Sept. 2005, pp. 593–596.

[11] Einstein, A., "Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen," (On the movement of small particles suspended in a stationary liquid demanded by the

molecular-kinetic theory of heat), *Annalen der Physik*, vol. 17, pp. 549–560, 1905.

[12] Fried, S., Kuperman, W., Sabra, K. G., and Roux, P., "Extracting the local Green's function on a horizontal array from ambient ocean noise," J. Acoust. Soc. Am., vol. 124, pp. 8393-8399, 2008.

[13] Godin, O. A., Emergence of the acoustic Green's function from thermal noise, *J. Acoust. Soc. Am.*, 121, no.2, p. EL96–EL102, 2007.

[14] Gurun, G., Hasler, P., and Degertekin, F., " Front-end receiver electronics for high-frequency monolithic CMUT-on-CMOS imaging arrays," IEEE Trans. Ultrason., Ferroelectr. Freq. Control 58, 1658, 2011.

[15] Gurun, G., Hochman, M., Hasler, P., and Degertekin, F., "Thermal-mechanical-noise-based- CMUT characterization and sensing," IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 59, no. 6, June 2012.

[16] Gurun, G., Zahorian, J., Hasler, P., and Degertekin, F., "Thermal mechanical noise based characterization of CMUTs using monolithically integrated low noise receiver electronics" in 2010 IEEE Ultrasonics Symposium (IUS) (IEEE International, San Diego, CA, 2010), pp. 567–570.

[17] Hochman, M., "Investigation of acoustic crosstalk effects in CMUT arrays," Master's Thesis, Georgia Institute of Technology, 2011.

[18] Hochman, M., Zahorian, J., Satir, S., Gurun, G., Xu, T., Karaman, M., Hasler, P., and Degertekin, F. L., "CMUT-on-CMOS for forward-looking IVUS: Improved fabrication and real-time imaging," in Ultrasonics Symposium (IUS), 2010 IEEE, Oct. 2010, pp. 555 –558.

[19] Johnson, D. H. and Dudgeon, D. E., *Array Signal Processing: Concepts and Techniques*. Prentice Hall, pp. 111-198, 1993.

[20] Kinsler, L., Frey, A., Coppens, A., and Sanders, J., *Fundamentals of Acoustics*. Wiley, pp. 210-245, 2000.

[21] Lani, S., Satir, S., Gurun, G., Sabra, K.G, Degertekin, F.L., "High Frequency Ultrasonic Imaging Using Thermal Mechanical Noise Recorded on Capacitive Micromachined Transducer Arrays," Appl. Phys. Lett., vol. 99, p 224103, (2011)

[22] Lani, S., Satir, S., Gurun, G., Sabra, K.G, Degertekin, F.L., "Ultrasonic sensing using thermal-mechanical noise recorded on monolithic CMUT-on-CMOS arrays," in International Ultrasonics Symposium (IUS), 2011 IEEE, Oct. 2011, pp. 104–107.

[23] Leroy, C., Lani, S., Sabra, K. G., Hodgkiss, W., Kuperman, W. A., Walker, S., Roux, P. "Enhancing the emergence rate of coherent wavefronts from ambient shipping noise correlations using spatio-temporal filters," J. Acoust. Soc. Am. 132, 883-893, 2012.

[24] Leroy, C., "Using ocean ambient noise cross-correlations for passive acoustic tomography," Master's Thesis, Georgia Institute of Technology, 2010.

[25] Lobkis, O. I. and Weaver, R. L., "On the emergence of the Green's function in the correlations of a diffuse field," J. Acoustic. Soc. Am., vol. 110, no. 6, pp. 3011-3017, 2001.

[26] Munk, W. H., "The Heard Island experiment," International Science Lecture Series, 1991.

[27] R. Brown, "Additional remarks on active molecules," *Philos. Mag.*, vol. 6, pp. 161–166, 1829.

[28] Roux, P., Kuperman, W., and NPAL Group, "Extracting coherent wave fronts from acoustic ambient noise in the ocean," J. Acoustic. Soc. Am., vol. 116, no. 4, pp. 1995-2003, 2004.

[29] Roux, P., Sabra, K., and Kuperman, W., "Ambient noise cross-correlation in free-space: theoretical approach," J. Acoust. Soc. Am., vol. 117, pp. 79-84, 2005.

[30] Sabra, K. G., Roux, P., and Kuperman, W. A., "Arrival-time structure of the time-averaged ambient noise cross-correlation function in an oceanic waveguide," J. Acoustic. Soc. Am., vol. 117, pp. 164-174, 2005.

[31] Sabra, K., "Emergence rate of the time-domain green's function from the ambient noise cross-correlation function," J. Acoust. Soc. Am., vol. 118, pp. 3524-3531, 2005.

[32] Sabra, K.G. and Archer, A. (2009), Tomographic elastography imaging of skeletal muscles from their natural vibrations, Appl. Phys. Lett., vol. 95, p203701, 2009.

[33] Schuster, G. T., *Seismic Interferometry* Cambridge University Press, 2009.

[34] Shapiro, N. M. and Campillo, M., "Emergence of broadband Rayleigh waves form correlations of the ambient seismic noise," Geophysical Research Letters, vol. 31, no. L07614, 2004.

[35] Siderius, M., "Adaptive passive fathometer processing," J. Acoust. Soc. Am., vol. 127, pp. 2193-2200, 2010.

[36] Skinner, J. D. and Hodgkiss, W. S. "A networked/autonomous receiving array system," in Proceedings of the OCEANS 2008, IEEE, Quebec City, 2008.

[37] Snieder, R., and Wapenaar, K., "Imaging with ambient noise," Physics Today, vol. 63, no. 9, p. 44, 2010.

[38] Son, J., Thomas, G., and B.C., F., Range-Doppler Radar Imaging and Motion Compensation. Boston: Artech House, 2001.

[39] Wapenaar, K., Draganov, D., and Robertson, J., editors, Seismic Interferometry: *History and Present Status*, Soc. of Exp. Geophys., Geophys. Rep. Series, No. 26, (SEG, Tulsa, OK, 2008).

[40] Weaver, R. L. and Lobkis, O. I., "Ultrasonics without a source: Thermal fluctuation correlations at MHz frequencies," Physical Review Letter, vol. 87, no. 134301, 2001.

[41] Weaver, R. L. and Lobkis, O. I., "The mean and variance of diffuse field correlations in finite bodies," J. Acoust. Soc. Am., vol. 118, pp. 3447–3456, 2005.

[42] Wenz, G. M., "Acoustic ambient noise in the ocean: Spectra and sources," J. Acoust. Soc. Am., vol. 34, pp. 1936-1956, 1962.

[43] Personal correspondence between Karim Sabra, William Hodgkiss, and Jeffrey Skinner via email, June 4, 2012.