Time-Delay-Estimate Based Direction-of-Arrival Estimation for Speech in Reverberant Environments

by

Krishnaraj Varma

Thesis submitted to the Faculty of
The Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering
APPROVED

Dr. A. A. (Louis) Beex, Chairman

Dr. Ira Jacobs Dr. Douglas K. Lindner

October 2002
Blacksburg, VA

KEYWORDS: Microphone array processing, Beamformer, MUSIC, GCC, PHAT, SRP-PHAT, TDE, Least squares estimate

Time-Delay-Estimate Based Direction-of-Arrival Estimation for Speech in Reverberant Environments

by
Krishnaraj Varma
Dr. A. A. (Louis) Beex, Chairman
The Bradley Department of Electrical and Computer Engineering

(Abstract)

Time delay estimation (TDE)-based algorithms for estimation of direction of arrival (DOA) have been most popular for use with speech signals. This is due to their simplicity and low computational requirements. Though other algorithms, like the steered response power with phase transform (SRP-PHAT), are available that perform better than TDE based algorithms, the huge computational load required for this algorithm makes it unsuitable for applications that require fast refresh rates using short frames. In addition, the estimation errors that do occur with SRP-PHAT tend to be large. This kind of performance is unsuitable for an application such as video camera steering, which is much less tolerant to large errors than it is to small errors.

We propose an improved TDE-based DOA estimation algorithm called time delay selection (TIDES) based on either minimizing the weighted least squares error (MWLSE) or minimizing the time delay separation (MWTDS). In the TIDES algorithm, we consider not only the maximum likelihood (ML) TDEs for each pair of microphones, but also other secondary delays corresponding to smaller peaks in the generalized cross-correlation (GCC). From these multiple candidate delays for each microphone pair, we form all possible combinations of time delay sets. From among these we pick one set based on one of the two criteria mentioned above and perform least squares DOA estimation using the selected set of time delays. The MWLSE criterion selects that set of time delays that minimizes the least squares error. The MWTDS criterion selects that set of time delays that has minimum distance from a statistically averaged set of time delays from previously selected time delays.

Both TIDES algorithms are shown to out-perform the ML-TDE algorithm in moderate signal to reverberation ratios. In fact, TIDES-MWTDS gives fewer large errors than even the SRP-PHAT algorithm, which makes it very suitable for video camera steering applications. Under small signal to reverberation ratio environments, TIDES-MWTDS breaks down, but TIDES-MWLSE is still shown to out-perform the algorithm based on ML-TDE.

Acknowledgements

I would like to express my most sincere gratitude to Dr. A. A. (Louis) Beex for his guidance during the course of this research work and my whole academic career at Virginia Tech. Without his invaluable advice, help and suggestions, this thesis work would not have been possible. Working in the DSP Research Lab at Virginia Tech has improved my technical knowledge and research skills and broadened my understanding of many aspects of electrical engineering and for this opportunity I am deeply indebted to Dr. Beex. I would also like to thank him for the financial assistantship that I was offered during the course of my MS degree.

Many thanks also to Dr. Douglas K. Lindner and Dr. Ira Jacobs for being on my committee and reviewing this work.

I would like to express my appreciation for the endless hours of discussion, technical and otherwise, that I have had with my colleague Takeshi Ikuma during my tenure at the DSPRL. Without his suggestions in MATLAB programming and invaluable help with computers in the lab, this thesis would have been very difficult.

Finally I would like to express my gratitude to my parents who have always been there for me throughout my good and bad times, always encouraging me and for making me who I am. This thesis would not have been possible without the love, affection, patience and guidance that they have provided.

Krishnaraj M. Varma

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	2 Dimensional
3D	3 Dimensional
A/D	Analog to Digital
ADSP®	Analog Devices Digital Signal Processor
ASG	Analytic Signal
D/A	Digital to Analog
DFT	Discrete Fourier Transform
DOA	Direction of Arrival
DSB	Delay and Sum Beamformer
DSP	Digital Signal Processing (Processor)
DSPRL	DSP Research Laboratory
EVD	Eigen Value Decomposition
FIR	Finite Impulse Response
GCC	Generalized Cross Correlation
GXPSD	Generalized Cross Power Spectral Density
IDFT	Inverse Discrete Fourier Transform
LS	Least Squares
LTI	Linear Time Invariant
ML	Maximum Likelihood
MUSIC	Multiple Signal Classification
MVB	Minimum Variance Beamformer
MWLSE	Minimum Weighted Least Squares Error
MWTDS	Minimum Weighted Time Delay Separation
NIST	National Institute of Standards and Technology
PHAT	Phase Transform
PSD	Power Spectral Density
SCOT	Smoothed Coherence Transform
SNR	Signal to Noise Ratio
SRP	Steered Response Power
SRR	Signal to Reverberation Ratio
TDE	Time Delay Estimate
TIDES	Tide Delay Selection
ULA	Uniform Linear Array
XPSD	Cross Power Spectral Density

1. Introduction

1.1. Motivation for Research

Direction of arrival (DOA) estimation of speech signals using a set of spatially separated microphones in an array has many practical applications in everyday life. DOA estimates from microphone arrays placed on a conference table can be used to automatically steer cameras to the speaker if the conference is part of a video conferencing session or a long distance TV based classroom [1]. In current video-conferencing systems or video classrooms, the control of the video camera is performed in one of three ways. Cameras that provide different fixed views of the room can be placed at different locations in the conference room to cover all the people in it. Secondly the system could consist of one or two cameras operated by humans. Finally the system could consist of manual switches for each user or group of users that would steer the camera in their direction when activated. The third category of systems is used commonly in long distance education that uses TV based classrooms. These systems turn out to be expensive in terms of extra hardware or manpower required to operate them effectively and reliably. It would be desirable to have one or two video cameras that can be automatically steered towards the speaker. Most conferences and classrooms typically have one person speaking at a time and all others listening. The speaker, however, could be moving around in the room. Thus there is a need to have a system that effectively and reliably locates and tracks a single speaker. Single speaker localization and tracking can be performed using either visual or acoustic data. A comprehensive tracking system using video data was developed by Wren et al. [2]. However, the algorithmic complexity and computational load required for such a system implies that a powerful computer be dedicated to performing this task. Methods based on acoustic data are typically far simpler in terms of complexity and computational load.

Another application of DOA estimation using microphone arrays is in speech enhancement for human computer interfaces that depend on speech inputs from operators [3]. Techniques used here, like superdirective beamforming, depend on accurate estimates of the DOA of the speech signals. The same is the case in hearing aids that use adaptive beamforming to capture acoustic signals in the presence of background noise and interference.

One factor that is common to all the applications mentioned above is that these involve estimation of the DOA of a sound source in a closed room. In a closed room, the sound at the microphone arrives not only directly from the source, but also because of multiple reflections from the walls of the room. This phenomenon, which is very common in conference rooms and classrooms, is called reverberation. The presence of a significant amount of reverberation can severely degrade the performance of DOA estimation algorithms. The motivation for this thesis comes from the need to find reliable algorithms that can locate and track a single speaker in a reverberant room using short signal frames from an array of microphones.

1.2. Fundamental Principles

The fundamental principle behind direction of arrival (DOA) estimation using microphone arrays is to use the phase information present in signals picked up by sensors (microphones) that are spatially separated. When the microphones are spatially separated, the acoustic signals arrive at them with time differences. For an array geometry that is known, these time-delays are dependent on the DOA of the signal. There are three main categories of methods that process this information to estimate the DOA [4].

The first category consists of the steered beamformer based methods. Beamformers combine the signals from spatially separated array-sensors in such a way that the array output emphasizes signals from a certain "look"-direction. Thus if a signal is present in the look-direction, the power of the array output signal is high and if there is no signal in the look-direction the array output power is low. Hence, the array can be used to construct beamformers that "look" in all possible directions and the direction that gives the maximum output power can be considered an estimate of the DOA. The delay and sum beamformer (DSB) is the simplest kind of beamformer that can be implemented. In a DSB, the signals are so combined that the theoretical delays computed for a particular look direction are compensated and the signals get added constructively. The minimum-variance beamformer [5] (MVB) is an improvement over simple DSB. In an MVB, we minimize the power of the array output subject to the constraint that the gain in the look-direction is unity.

The main advantage with a steered beamformer based algorithm is that with one set of computations we are able to detect the directions of all the sources that are impinging on the array. Thus it is inherently suited to detecting multiple sources. From considerations of the

eigen-values of the spatial correlation matrix, if we have *N* elements in an array, it is not possible to detect more than *N-1* independent sources. Methods like complementary beamforming [6] have been proposed to detect DOAs even when the number of sources is equal to or greater than the number of sensors. For our requirement, which is detecting and tracking a single user, the computational load involved in a steered beamformer based method is deemed to be too large. For example, if we have to perform 3-dimensional DOA estimation we have to compute the array output power using beamformers that are looking in all azimuths (0 to 360°) and all elevations (-90 to +90°). For a resolution of 1°, this involves a search space of 64,979 search points. If we add to this the condition that the source is in the near field of the array, then the set of possible ranges (distances of the sources from the array) is added to the search space.

The second category consists of high-resolution subspace based methods. This category of methods divides the cross-correlation matrix of the array signals into signal and noise subspaces using eigen-value decomposition (EVD) to perform DOA estimation. These methods are also used extensively in the context of spectral estimation. Multiple signal classification (MUSIC) is an example of one such method. These methods are able to distinguish multiple sources that are located very close to each other much better than the steered beamformer based methods because the metric that is computed gives much sharper peaks at the correct locations. The algorithm again involves an exhaustive search over the set of possible source locations.

The third and final category of methods is a two-step process. In the first step the time-delays are estimated for each pair of microphones in the array. The second step consists of combining or fusing this information based on the known geometry of the array to come up with the best estimate of the DOA. There are various techniques that can be used to compute pair-wise time delays, such as the generalized cross correlation (GCC) method [7] or narrowband filtering followed by phase difference estimation of sinusoids. The phase transform (PHAT) is the most commonly used pre-filter for the GCC. The estimated time-delay for a pair of microphones is assumed to be the delay that maximizes the GCC-PHAT function for that pair. Fusing of the pair-wise time delay estimates (TDE's) is usually done in the least squares sense by solving a set of linear equations to minimize the least squared error. The simplicity of the algorithm and the fact that a closed form solution can be obtained (as opposed to searching) has made TDE based methods the methods of choice for DOA estimation using microphone arrays.

1.3. Overview of Research

Various factors affect the accuracy of the DOA estimates obtained using the TDE based algorithm. Accuracy of the hardware used to capture the array signals, sampling frequency, number of microphones used, reverberation and noise present in the signals, are some of these factors. The hardware that is used should introduce minimum phase errors between signals in different channels. This is a requirement no matter what method is used for DOA estimation. Also, the more microphones we use in the array the better the estimates are that we get.

The sampling frequency becomes an important factor for TDE based methods especially when the array is small in terms of distance between the microphones. This is because small distances mean smaller time delays and this requires higher sampling frequencies to increase the resolution of the delay estimates. In the case of low sampling frequencies the parabolic interpolation formula [9] has been used before to come up with a more accurate sub-sample estimate of the time delay. In this thesis we look at an alternate approach to time domain interpolation by directly computing the sub-sample correlation values from the cross-power spectral density (XPSD) while computing the inverse Fourier transform.

Also for the purpose of fast tracking we study the performance of the TDE based algorithms with very short frames (32-64 *ms*) of signal data in the presence of moderate reverberation. Under such conditions the performance of the GCC-PHAT based method is only marginal compared to the performance we obtain with another method, called the steered response power (SRP) method [4]. The performance of the GCC-PHAT based method is degraded by the presence of impulsive errors in certain frames. This was caused by the algorithm picking the wrong peak in the GCC as the one corresponding to the delay. Initial work to improve these results was geared towards estimating a weighted least squares estimate [8]. The idea behind this is that while computing the least squares estimate of the DOA, we weigh those equations less which are found to be less reliable based on certain criteria. It was found that because the time-delay of arrival between two microphones was not a linear, but rather a trigonometric function of the angle of arrival, larger time-delays would give rise to less reliable angle estimates. This observation leads to one of the weighing coefficients. Also, most GCC functions were found to have multiple peaks out of which the strongest peak was assumed to

correspond to the true time-delay. Therefore this method is a maximum likelihood (ML) estimator. In the presence of reverberation, the strongest peak turns out to not always be at the correct delay. Therefore those time-delays whose second strongest peaks are close in strength to the strongest peak are also less reliable estimates. This leads to the second weighing coefficients. These two weighing coefficients can be combined to give a weighted least squares estimate of the DOA. This kind of weighting was found to reduce the number of impulsive errors in the DOA estimate, but it did not eliminate them. Impulsive errors in the DOA estimates are very undesirable in applications like video camera steering or beamforming. A unit norm constrained adaptive algorithm was suggested to remove the impulsive errors [8]. This algorithm, though slower to reach the steady-state DOA estimated, remains in the proximity of the correct DOA and does not contain impulsive errors.

From extensive studies of frame-wise GCC data, we propose an alternate method to improve the reliability of pre-adaptation estimates named Time Delay Selection (TIDES). For the frames that contained impulsive errors, it was observed that, though the wrong delay had the strongest peak, a weak peak was almost always observed at the correct delay also. Therefore it makes sense not to discard these other peaks. Since each pair of microphones could give us multiple time delay candidates, we have in our hand several candidate time-delay sets, from among which we should be choosing one based on some criterion. We propose two criteria, namely the Minimum Weighted Least Squares Error (MWLSE) and the Minimum Weighted Time Delay Separation (MWTDS), to pick one of the sets of time-delay estimates. The weighting in both cases is done so that those TDE sets that correspond to stronger GCC peaks are more likely to be picked during the search for the minimum. In the TIDES-MWLSE method we select that candidate TDE set that minimizes the weighted least squares error. In the TIDES-MWTDS method, we select that candidate TDE set that minimizes the weighted distance (separation) from a statistical average of previously selected TDE sets. Specifically, we try to find that TDE set that is closest to a median filtered TDE set from the previous five frames. We show using simulations and experiments that by just picking one extra time delay (if available) for each microphone pair, we are able to get much improved performance over the ML estimator without a great increase in computational requirement.

1.4. Organization

The remainder of the thesis is organized as follows. Chapter 2 describes in detail the three types of DOA estimation algorithms. This chapter also lays down the array conventions used throughout the remainder of the thesis and develops some basic array processing theory that is central to any multi-channel system. Chapter 3 describes the nature of sound and its behavior in a closed room with partially reflective walls that cause the effect called reverberation. The image model for approximately simulating the reverberation is developed. Finally the chapter looks at the effect that reverberation has on the DOA estimation algorithms developed in Chapter 2. Chapter 4 introduces the generalized cross-correlation with phase transform and the ML-TDE method based on the GCC-PHAT. It also describes the SRP-PHAT method and provides simulation and experimental results that show performance improvement over the methods in Chapter 2 in the presence of reverberation. Chapter 5 gives some simulation and experimental results using the GCC-PHAT based method and provides a couple of possible methods to get some improvement in performance in the form of signal to noise ratio (SNR)-based thresholding of the XPSD and symmetric extension of the frame signal data. We go on to describe the details of the algorithm based on the MWLSE and MWTDS criteria and provide both simulation and experimental results to show improvement in performance. Finally Chapter 6 briefly states the conclusions from this research work and possible avenues for future work.

2. Summary of DOA Estimation Techniques

2.1. Microphone Array Structure and Conventions

Figure 2.1 shows a 4-element uniform linear array (ULA) of microphones and a sound source in the far field of the array. We will be using the uniform linear array to develop the principles of these conventional methods. Without loss of generality, these methods can be extended to three-dimensional arrays. The array consists of 4 microphones placed in a straight line with a uniform distance, d, between adjacent microphones. The sound source is assumed to be in the far field of the array. This means that the distance of the source, S, from the array is much greater than the distance between the microphones. Under this assumption, we can approximate the spherical wavefront that emanates from the source as a plane wavefront as shown in the figure. Thus the sound waves reaching each of the microphones can be assumed to be parallel to each other. The direction perpendicular to the array is called the broadside direction or simply the broadside of the array. All DOA's will be measured with respect to this direction. Angles in the clockwise direction from the broadside (as the one shown in Figure 2.1) are assumed to be positive angles and angles in the counter clockwise direction from the broadside are assumed to be negative angles.

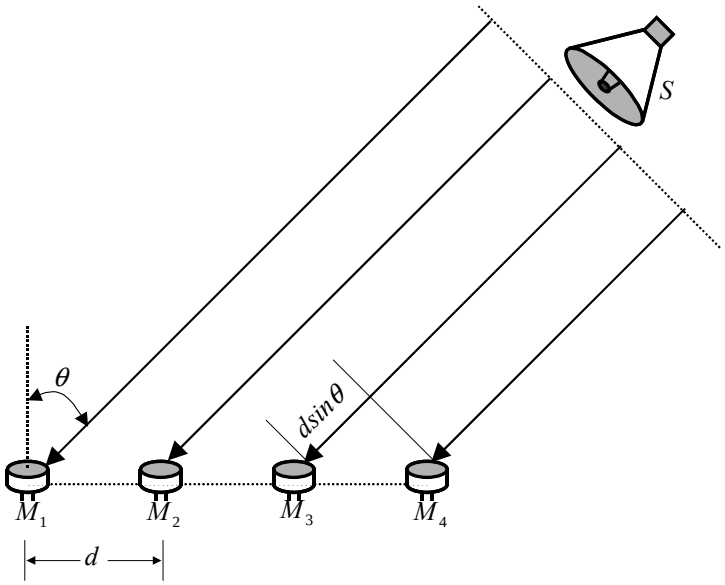

Figure 2.1 ***Uniform Linear Array with Far Field Source.***

The signal from the source reaches the microphones at different times. This is because each sound wave has to travel a different distance to reach the different microphones. For example the signal incident on microphone M_3 has to travel an extra distance of $d\sin\theta$ as compared to the signal incident on microphone M_4. This results in the signal at microphone M_3 being a time-delayed version of the signal at microphone M_4. This argument can be extended to the other microphones in the array.

As a convention we will fix microphone M_1 as the reference microphone. Let the signal incident on M_1 be $s(t)$. Then the signal incident on M_2 is a time-advanced version of $s(t)$ and the advance is equal to $\frac{d\sin\theta}{v}$ where v is the velocity of sound (355 ms^{-1}). In other words, the signal incident on M_2 is a time-delayed version of $s(t)$ with the delay being $-\frac{d\sin\theta}{v}$. Thus positive values of θ give negative delays and negative values of θ give positive delays. To summarize, the signals picked up by the array at each of the microphones are given below.

$$\begin{aligned} x_{M1} &= s(t) \\ x_{M2} &= s(t-\tau_{21}) \\ x_{M3} &= s(t-\tau_{31}) \\ x_{M4} &= s(t-\tau_{41}) \end{aligned} \tag{2.1}$$

where

$$\tau_{ij} = -\frac{d_{ij}\sin\theta}{v} \tag{2.2}$$

Consider the pair of microphones shown in Figure 2.2. These microphones form part of a uniform linear array with a distance *d* between adjacent microphones. Also shown are two sources that are incident on the array at an angle of θ with respect to the broadside. The angles made by the sources are measured with respect to two different broadsides, one in front of the array and the other behind it. The extra distance traveled by either source signal to reach M_1 as compared to M_2 is $d\sin\theta$. Thus the pair-wise time delays associated with either source will be the same. This is under the assumption that the microphones are omni-directional, which means

that the gain of the microphone does not change the direction of the acoustic wavefront. What this means is that the ULA is only capable of distinguishing that the source is at an angle with respect to the line of the array, but not where exactly it is around the line. This is referred to as front-back ambiguity of the array. A ULA can uniquely distinguish angles between – 90° and + 90° with respect to the broadside of the array.

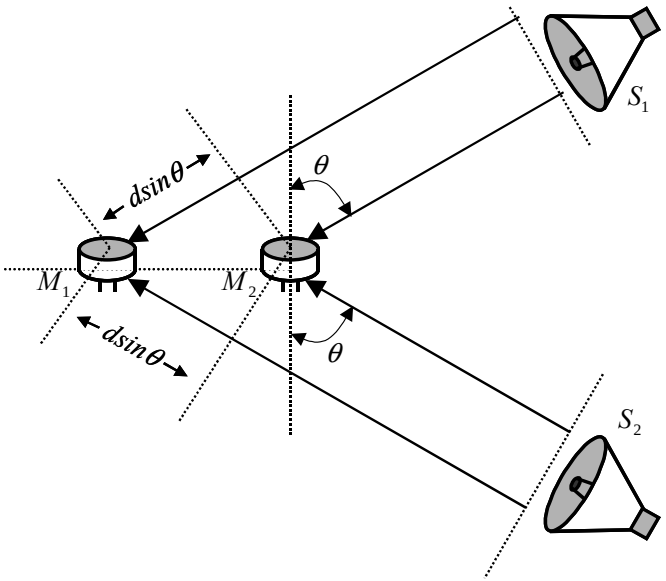

Figure 2.2 ***Uniform Linear Array shown with front-back ambiguity.***

2.2. Restrictions on the Array

There is a relationship between the frequency content of the incident signal and the maximum allowed separation between each pair of microphones in the array. Consider two sinusoids of the same frequency, but with a phase difference of ϕ between them. This phase difference is restricted to be between $-\pi$ and π. A phase lag of ϕ which is greater than π cannot be distinguished from a phase lead of $2\pi - \phi$ and vice-versa. For example consider the sinusoid shown in Figure 2.3(a) with the second sinusoid having a phase lead of $\frac{5\pi}{4}$. In Figure 2.3(b) we have a sinusoid with the second sinusoid having a phase lag of $2\pi - \frac{5\pi}{4} = \frac{3\pi}{4}$. It is clearly seen that these two situations are identical. Thus any phase difference out of the range of $-\pi$ and π will be wrapped around to within that range.

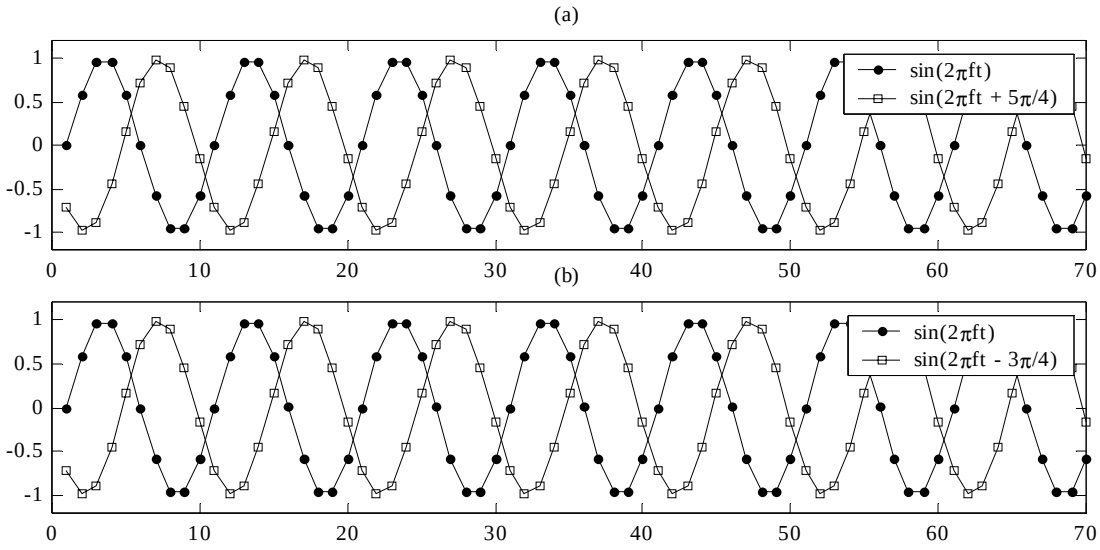

Figure 2.3 ***Two pairs of sinusoids with different phase differences appear identical.***

This fact places an important restriction on the array geometry to prevent spatial aliasing, when performing narrowband DOA estimation. Spatial aliasing happens when the phase delay, at the frequency of interest, between signals from a pair of microphones, exceeds π. This causes the time delays to be interpreted wrongly, which in the end results in wrong DOA estimates. Consider a signal incident on a ULA at an angle θ. Let this broadband signal have a maximum frequency of $f_{\max}$. If we would like to restrict the phase difference, at this frequency, between signals of any pair of microphones to be less than or equal to π, then we require $2\pi f_{\max}\tau \le \pi$, where τ is the signal time delay between the two microphones and $\tau = \frac{d\sin\theta}{v}$, where d is the distance between the microphones, θ is the incident angle and v is the velocity of sound. Rearranging these terms, we have $d \le \frac{1}{2}\left(\frac{v}{f_{\max}}\right)\frac{1}{\sin\theta}$. Since we do not have any control over the incident direction, we take the worst-case scenario, which is $\theta = 90^{\circ}$. Also the term $\frac{v}{f_{\max}}$ is the same as $\lambda_{\min}$, the smallest wavelength present in the signal. Thus we have the condition $d \le \frac{\lambda_{\min}}{2}$, which means that the distance between any pair of microphones in the array should not exceed half the smallest wavelength present in the signal. When this condition is satisfied, spatial aliasing is avoided and correct DOA estimates can be obtained. Note that this

consideration becomes important only when we are performing TDE from phase difference estimates of narrowband signals. Algorithms that directly compute the time delays of broadband signals using cross-correlations are not restricted in this manner.

2.3. Steered Beamformer Based Methods

The property of beamformers to enhance signals from a particular direction and attenuate signals from other directions can be used to perform DOA estimation. A beamformer can be constructed for each direction of interest (hereafter referred to as the look direction of the beamformer) and the power of the array output can be computed. The look directions that give large power outputs can then be taken as the estimated DOA's of the incident signals. When the power is plotted against the look directions, it exhibits a peak for each look direction that has a signal present. Depending on the type of beamformer used, many different methods can be used.

2.3.1. Beamformer Concept

The concept of a beamformer is to use a set of spatially separated microphones and select a direction from which to accept signals, while rejecting signals from other directions. Beamformers can be narrowband or broadband depending on the bandwidth of the signals that they deal with. Almost all DOA estimation algorithms use narrowband beamforming techniques to get separate DOA estimates for the many different frequency bands. These separate estimates are then combined to get one estimate based on feasible statistical observations.

Narrowband beamformers assume that the incident signal that the beamformer is trying to capture has a narrow bandwidth centered at a particular frequency. If the signal does not satisfy this condition, then it can be bandpass filtered to convert it into a narrowband signal. In this case it should be ensured that the same bandpass filter is used on all channels of the array so that the relative phase information between channels is not altered. Let *s(t)* be such a narrowband source signal with a center frequency f_c. Consider any arbitrary *N* element microphone array on which this source signal is incident from an unknown angle. Let the vector $\mathbf{x}(k) = [x_0(k) \quad x_1(k) \quad \cdots \quad x_{N-1}(k)]^T$ represent the set of signal samples from the *N* microphones at time-sample *k*. If microphone M_0 is fixed as the reference microphone, then the vector $\mathbf{x}$ can be rewritten as

$$\mathbf{x}(k)=\begin{bmatrix}a_0\,s(k) & a_1 s(k-\tau_{10}) & \cdots & a_{N-1}s\left(k-\tau_{(N-1)0}\right)\end{bmatrix}^T+\begin{bmatrix}v_0(k) & v_1(k) & \cdots & v_{N-1}(k)\end{bmatrix}^T \quad (2.3)$$

where τ_{i0} is the sample delay of the signal at microphone M_i with respect to the signal at microphone M_0, a_i is a gain factor associated with each microphone and $v_i(k)$ represents the noise in each microphone. For the case of a linear array $\tau_{i0}=\left(-\frac{d_{i0}\sin\theta}{v}\right)$ where d_{i0} is the distance from microphone M_i to microphone M_0. Note that in many cases, the delays between microphones falls in between samples, in which case they will have to be rounded to the nearest sample delay. The frequency domain representation of the vector $\mathbf{x}(k)$ can be obtained by taking the Fourier transform of (2.3).

$$\begin{aligned}\mathbf{X}(\omega)=&\begin{bmatrix}a_0S(\omega)e^{-j\omega\tau_{00}} & a_1S(\omega)e^{-j\omega\tau_{10}} & \cdots & a_{N-1}S(\omega)e^{-j\omega\tau_{(N-1)0}}\end{bmatrix}^T\\&+\begin{bmatrix}V_0(\omega) & V_1(\omega) & \cdots & V_{N-1}(\omega)\end{bmatrix}^T\end{aligned} \quad (2.4)$$

Alternatively

$$\mathbf{X}(\omega)=S(\omega)\mathbf{d}(\omega)+\mathbf{V}(\omega) \quad (2.5)$$

where

$$\mathbf{d}(\omega)=\begin{bmatrix}a_0e^{-j\omega\tau_{00}} & a_1e^{-j\omega\tau_{10}} & \cdots & a_{N-1}e^{-j\omega\tau_{(N-1)0}}\end{bmatrix}^T \quad (2.6)$$

$\mathbf{d}(\omega)$ is called the array steering vector or the array manifold [5]. If all the microphones are assumed to be identical and the distances between the source and the microphones are assumed to be large (far-field assumption), then the gains in each term of the array-manifold vector are identically equal to unity. Thus the array-manifold vector can be re-written as

$$\mathbf{d}(\omega)=\begin{bmatrix}e^{-j\omega\tau_{00}} & e^{-j\omega\tau_{10}} & \cdots & e^{-j\omega\tau_{(N-1)0}}\end{bmatrix}^T \quad (2.7)$$

For the narrowband case, we will be dealing with only the center frequency and so (2.5) can be written as

$$\mathbf{X}(\omega_c) = S(\omega_c)\mathbf{D}(\omega_c) + \mathbf{V}(\omega_c) \tag{2.8}$$

Figure 2.4 shows the structure of a frequency domain narrowband beamformer [5]. The signal picked up at each microphone is first transformed to the frequency domain by taking its discrete Fourier transform (DFT). From among the frequency bins, the DFT values corresponding to the center frequency are picked. These frequency coefficients are multiplied by appropriate complex weights and then summed to get the frequency domain representation of the array output. Thus we have

$$Y(\omega_c) = \mathbf{W}^H \mathbf{X}(\omega_c) \tag{2.9}$$

where

$$\mathbf{W} = \begin{bmatrix} W_0 & W_1 & \cdots & W_{N-1} \end{bmatrix}^T \tag{2.10}$$

The inverse Fourier transform (IDFT) of $Y(\omega_c)$ is the array output signal. The weights of the beamformer are chosen to impart proper gain and phase changes (delays) to the signals in each channel so that when they add coherently, the array passes, with high gain, signals from the look direction and attenuates signals from other directions.

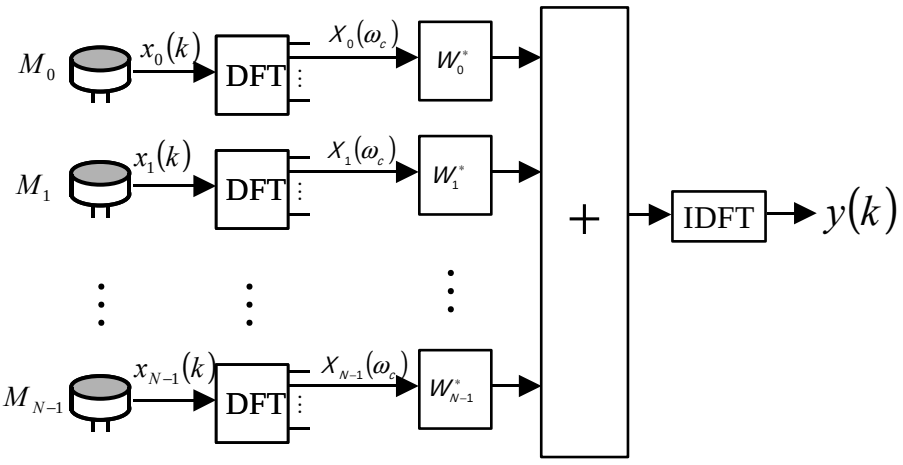

Figure 2.4 ***Frequency Domain Narrowband Beamformer Structure.***

A time domain version of such a beamformer would involve analytic signal generators (ASG's) for each of the N channels. These ASG's generate complex signals that have power

only in the positive frequencies. They transfer the power from the negative frequencies to the corresponding positive frequencies. The real part of the analytic signal is called the in-phase component and the imaginary part of the analytic signal is called the quadrature component. The in-phase part and the quadrature part of the analytic signal have a phase difference of $\frac{\pi}{2}$ at all frequencies of the signal. Implementation of ASG's is done using finite impulse response (FIR) filters [10]. A Hilbert transformer FIR filter imparts the $\frac{\pi}{2}$ phase lag to generate the quadrature component. The in-phase component of the analytic signal is generated by passing the microphone signal through a delay filter that imparts to it, a delay equal to the delay of the Hilbert transformer filter. The advantage of using analytic signals over the raw microphone signals is that it becomes easy to impart any arbitrary delay to these signals in order to perform beamforming. By multiplying these complex signals with appropriate weights of the form $a_i e^{-j\omega_c \tau_i}$ we can impart any gain and any delay to these signals. Such delayed signals can then be summed to generate the array output. The time domain equation for the narrowband beamformer is

$$y(n) = \mathbf{w}^H \mathbf{x}(n) \tag{2.11}$$

where **w** is the vector of complex beamformer weights and $\mathbf{x}(n)$ is the vector of analytic signals from the *N* channels.

The power of the beamformer output is an important parameter. In the frequency domain, the array output power spectral density (PSD) can be written as

$$\begin{aligned} \Phi_{YY}(\omega_c) &= Y(\omega_c) Y^*(\omega_c) \\ &= \left(\mathbf{W}^H \mathbf{X}(\omega_c)\right)\left(\mathbf{W}^H \mathbf{X}(\omega_c)\right)^* \\ &= \left(\mathbf{W}^H \mathbf{X}(\omega_c)\right)\left(\mathbf{X}^H(\omega_c)\mathbf{W}\right) \\ &= \mathbf{W}^H \left(\mathbf{X}(\omega_c)\mathbf{X}^H(\omega_c)\right)\mathbf{W} \\ &= \mathbf{W}^H \mathbf{\Phi}_{\mathbf{XX}}(\omega_c)\mathbf{W} \end{aligned} \tag{2.12}$$

where $\Phi_{\mathbf{XX}}(\omega_c)$ is an N by N matrix representing the cross power spectral densities of the channel input signals. In all the expressions above ω_c represents the frequency of the narrowband input signal.

Another important parameter of the array is the array response function. This function is the more general form of the frequency response. The array response function, $R(\theta,\omega)$, of a ULA represents the response of the array to a complex exponential at frequency ω incident on the array at an angle of θ. Consider the noiseless case where the Fourier transform of the array output is given by

$$\begin{aligned} Y(\omega) &= \mathbf{W}^H \mathbf{X}(\omega) \\ &= \mathbf{W}^H S(\omega) \mathbf{D}(\omega) \end{aligned} \tag{2.13}$$

In general the τ_{i0} terms present in the expression for $\mathbf{D}(\boldsymbol{\omega})$ are functions of the incident angle θ. Thus the equation above can be rewritten as

$$Y(\theta,\omega) = \mathbf{W}^H S(\omega) \mathbf{D}(\theta,\omega) \tag{2.14}$$

Then the array response function is given by

$$R(\theta,\omega) = \frac{Y(\theta,\omega)}{S(\omega)} = \mathbf{W}^H \mathbf{D}(\theta,\omega) \tag{2.15}$$

For narrowband beamformers, the frequency will be fixed to the center frequency and the array response becomes a function only of the incident angle. Thus for narrowband beamformers, we have the array response

$$R(\theta) = \mathbf{W}^H \mathbf{D}(\theta) \tag{2.16}$$

2.3.2. Steered Delay and Sum Beamformer Based Method

The delay and sum beamformer (DSB) is the simplest type of beamformer. Here the signals of each channel are given delays that compensate for the delays caused by the signal

arriving at the array from the look direction. Therefore the weights for the delay and sum beamformer with a look direction θ_{look} are given by [5]

$$\mathbf{W} = \mathbf{D}\left(\theta_{look}\right) \tag{2.17}$$

Figure 2.5 shows the magnitude of the simulated array response at 800 Hz $\left(\lambda = 44.375\,cm\right)$ for a 10 element ULA with an inter-element distance of 10 cm steered to a look direction of 0°. Sidelobes observed at – 40° and + 40° are significant at – 13 dB. Also the mainlobe is very broad to cover between – 20° and + 20°. Thus, this beamformer, though simple, is not very good at focusing onto a look direction and at rejecting all other directions.

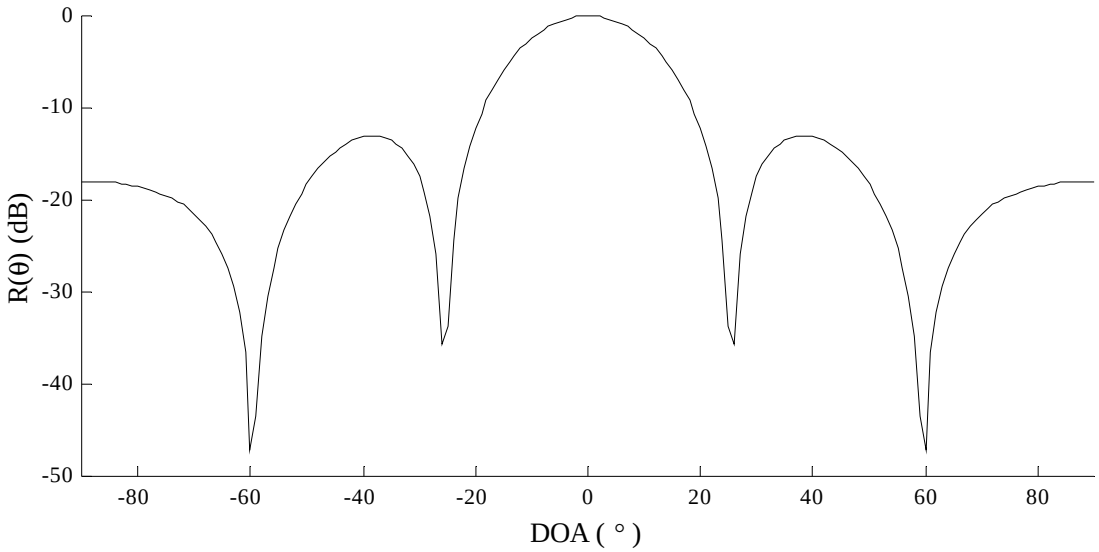

Figure 2.5 ***Magnitude of Array Response for a DSB with a 10-element ULA and a look angle of*** 0° ***at F = 800 Hz.***

To perform DOA estimation with this type of beamformer, we search all the angles of interest between – 90° and + 90° by constructing delay and sum beamformers for each of these directions. We can compute the output PSD at the frequency of interest for each direction and the directions that give high power outputs can be assumed to be directions of impinging signals. Figure 2.6 shows the output PSD for a 4-element ULA with a sinusoidal input signal of 800 Hz coming in at 30° incident angle. The plot exhibits a peak at 30° and can be used to estimate the DOA.

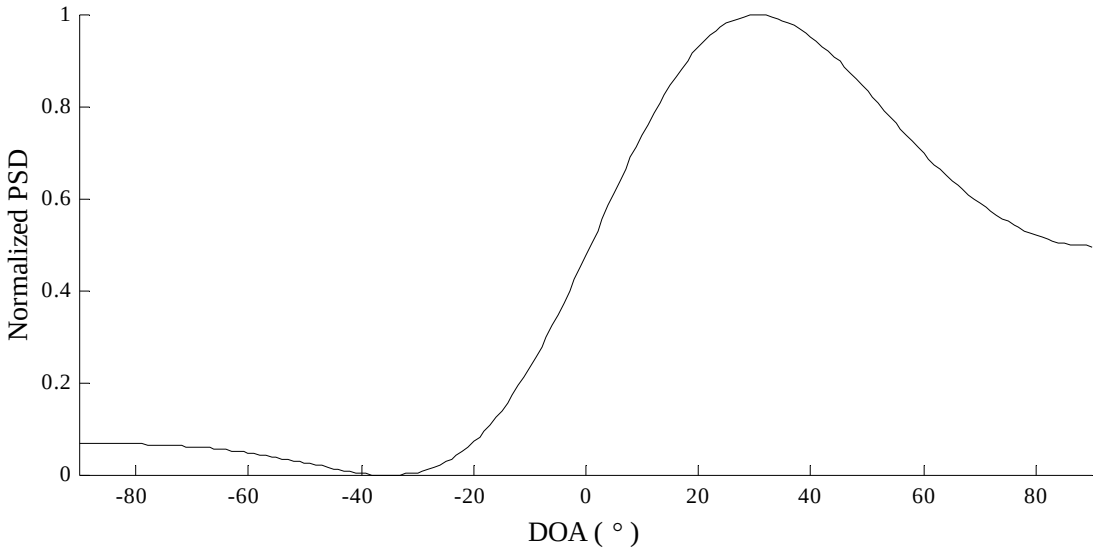

Figure 2.6 ***Output PSD against incident angle for a 4-element ULA with DSB at F = 800 Hz.***

2.3.3. Broadband Signal Considerations

Another important consideration is how the algorithm would perform in the presence of broadband signals, for example, speech signals. The spectrogram of a typical speech signal is shown in Figure 2.7. It can be seen that speech signals have significant power over a wide range of frequencies. Also speech signals exhibit formant frequencies. These are specific frequencies that exhibit higher power when compared to surrounding frequencies. In Figure 2.7 such frequencies appear as horizontal bands in the spectrogram. Since these frequencies have significant power, it makes sense to use these frequencies to perform DOA estimation. First we perform a DFT on the speech frame to get the frequency domain coefficients. Then we set a threshold power and pick up the frequency coefficients that are above that power. For the simulations performed here, we set the threshold at 15 dB so that all frequency bins that were more than 15 dB below the power of the strongest frequency were rejected. A peak-picking algorithm was run on these coefficients to pick up the dominant frequencies. Frequency domain narrowband DOA estimation is performed at each of these frequencies. The mean of these estimates formed a good approximation to the true DOA.

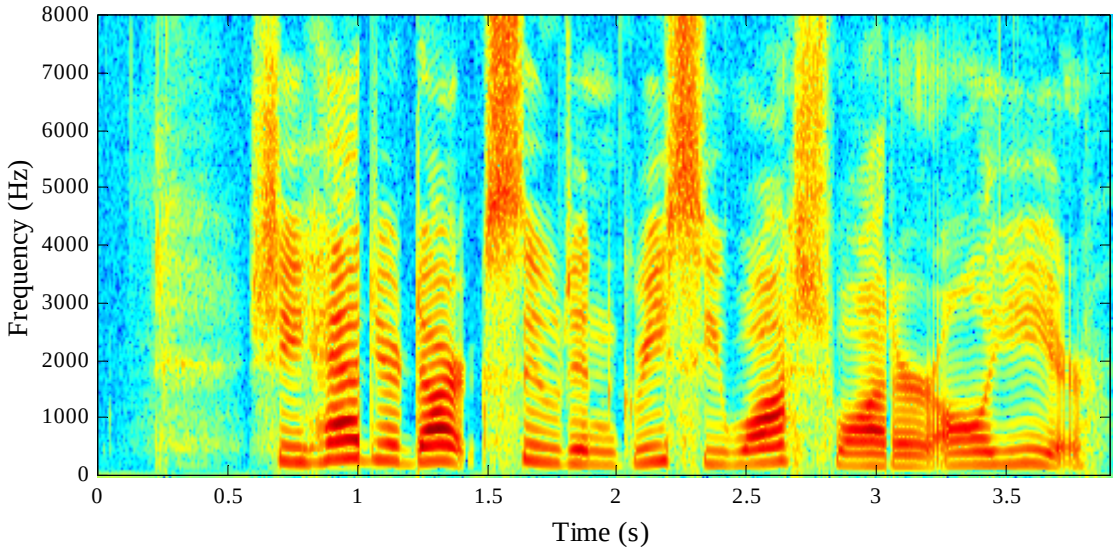

Figure 2.7 ***Spectrogram of a typical speech signal.***

Simulations were performed using a speech frame from the DARPA-TIMIT database available from the National Institute of Standards and Technology (NIST) [11]. These speech signals were sampled at 16 kHz. The simulations were again done for a 4-element ULA with a spacing of 10 cm, for a female speech signal from an incident direction of 22° and signal to noise ratio (SNR) of 30 dB. In order to simulate fine delays the signal was interpolated 10 times. After computing the finely delayed microphone signals they were decimated back to the original sampling frequency. A 4000-point DFT was computed using 2000 samples of the simulated microphone signals. The peak-picking algorithm found 13 frequencies and narrowband DOA estimation was done at these 13 frequencies. The mean DOA estimate was computed to be 21.85°. Figure 2.8 gives a plot of the estimated DOA against the peak-picked frequencies. Notice that not all of the picked frequencies give the same DOA estimate. The average over all the picked frequencies gives a fairly good estimate of the DOA. Another method to pick the correct DOA would be to add the output PSDs obtained for all the picked frequencies and search for the angle at which this sum maximizes. Figure 2.9 shows the cumulative PSD over the 13 picked frequencies plotted against the incident DOA. This cumulative PSD maximizes at 22°.

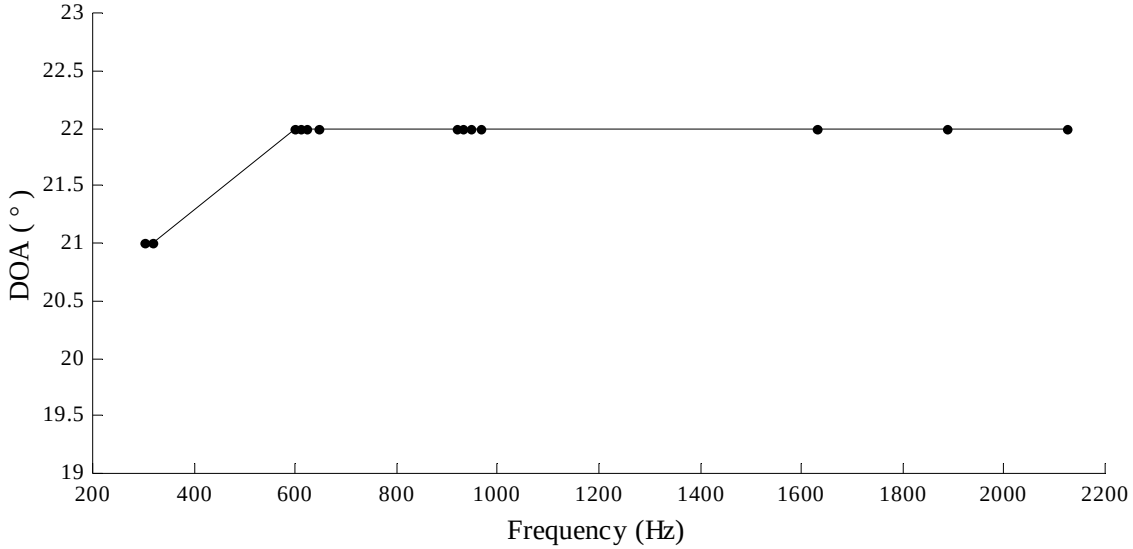

Figure 2.8 ***Estimated DOA against chosen formant frequency using DSB based method.***

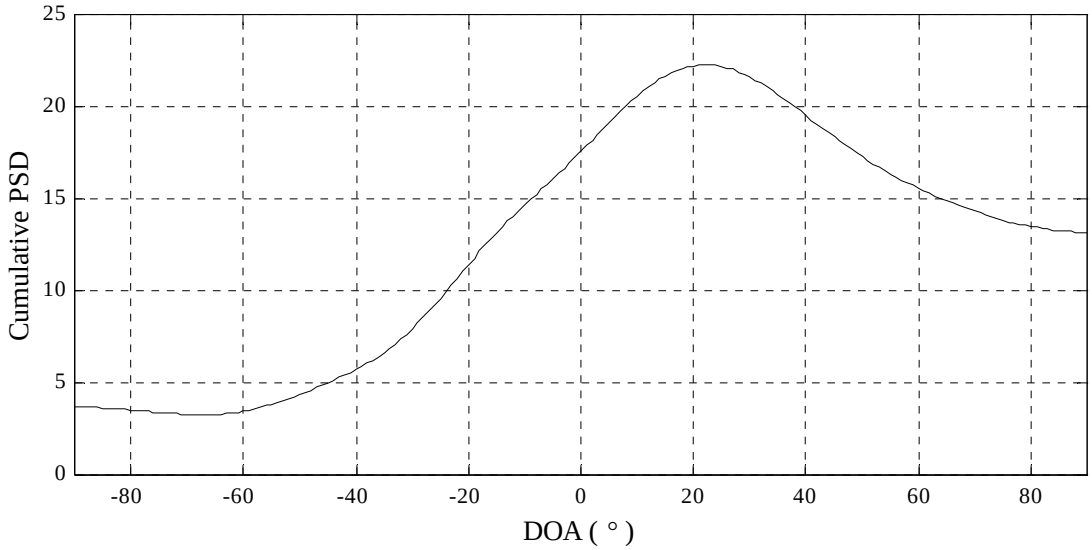

Figure 2.9 ***Cumulative PSD over all picked frequencies plotted against incident angle shows a peak at the correct DOA = 22°.***

2.4. Subspace Based DOA Estimation

Subspace based methods first decompose the cross-correlation matrix of the array signals into signal and noise subspaces using eigen-value decomposition. Then a search is performed using either the noise subspace or the signal subspace over all possible DOAs to determine the most likely one. The Multiple Signal Classification (MUSIC) introduced by Schmidt [12] is one

of the most popular subspace based narrowband methods. MUSIC is also extensively used in spectral estimation to estimate the frequency and other parameters of incident signals.

Consider a microphone array of M microphones and let K source signals be incident on it. Let $\mathbf{X}$ be a $N \times M$ matrix, each column of which is a snapshot (of length N) of the signal incident at a microphone. Moreover, let the signals in $\mathbf{X}$ be complex analytic signals constructed, as described in Section 2.3.1, from the real incident signals. The source signals are all assumed to be narrowband signals with a center frequency ω_c. The spatial correlation matrix of these array signals is an $M \times M$ matrix given by

$$\mathbf{R} = \mathbf{X}^H \mathbf{X} \tag{2.18}$$

An eigen-value decomposition of $\mathbf{R}$ [12] decomposes the M-dimensional space of the matrix into a K-dimensional signal subspace and a (M-K)-dimensional noise subspace. The highest K eigen-values determine the signal subspace, $\mathcal{S}$, which is spanned by the corresponding eigen-vectors. The other (M-K) eigen-values determine the noise subspace, $\mathcal{N}$, which is spanned by the corresponding eigen-vectors. In fact, theoretically, if the signal arriving at the microphones is corrupted by un-correlated white noise, these M-K eigen-values are equal to the variance of the noise in the incident signals. The signal and noise subspaces are the orthogonal complements of each other. The two sets of eigen-vectors span the respective subspaces.

Now consider any arbitrary vector, $\mathbf{s}$. The Euclidean distance of $\mathbf{s}$ from the signal subspace is the length of the projection of $\mathbf{s}$ in the noise subspace. Thus the squared magnitude of this distance is given by

$$\left| d(\mathbf{s}) \right|^2 = \sum_{i=K+1}^{M} \left| \mathbf{e}_i^H \mathbf{s} \right|^2 \tag{2.19}$$

where $\mathbf{e}_i$ represents the i^{th} eigen-vector of $\mathbf{R}$. Note that here we have used the eigen-vectors that span the noise subspace to compute the distance of $\mathbf{s}$ from the signal subspace. A signal that belongs to the signal subspace minimizes this squared distance. Another way to express minimizing the latter is by maximizing the reciprocal of the squared distance.

$$P(\mathbf{s}) = \frac{1}{\left|d(\mathbf{s})\right|^2} = \frac{1}{\sum_{i=K+1}^{M} \left|\mathbf{e_i^H s}\right|^2} \tag{2.20}$$

If $\mathbf{s}$ belongs to the signal subspace, then the distance is zero and the metric $P(\mathbf{s})$ approaches infinity. In practice, when $\mathbf{s}$ belongs to the signal subspace, $P(\mathbf{s})$ goes to a very large value.

Now consider the *M*-dimensional array-manifold vector, $\mathbf{d}(\theta, \omega_c)$, that was introduced in Section 2.3.1. This vector represents the spatial sampling of a narrowband complex exponential of frequency ω_c arriving from an angle θ. Thus if θ happens to be the incident angle of arrival, $\mathbf{d}(\theta, \omega_c)$ belongs to the signal subspace and thus $P(\mathbf{d}(\theta, \omega_c))$ approaches a large value. The MUSIC algorithm can now be defined as follows. Compute $P(\mathbf{d}(\theta, \omega_c))$ (or $P(\theta)$ for brevity) for all possible angles of arrival.

$$P(\theta) = \frac{1}{\sum_{i=K+1}^{M} \left|\mathbf{e_i^H d}(\theta)\right|^2} \tag{2.21}$$

Here we have removed the explicit dependence on ω_c because it is a fixed frequency. The true angle of arrival produces a sharp peak in $P(\theta)$ and this feature can be used to determine the DOA. Figure 2.10 shows a plot of $P(\theta)$ for a 4-element ULA with a spacing of 10 *cm*. This was simulated for a source signal of 800 *Hz* coming in at a direction of 30°. The spatial correlation matrix was computed using 200 samples of the array signal. When compared to the PSD of the delay and sum beamformer shown in Figure 2.6, MUSIC exhibits a much sharper peak at the true DOA. Thus subspace-based methods like MUSIC provide higher resolution to facilitate separating the DOAs of multiple sources that are located very close to each other.

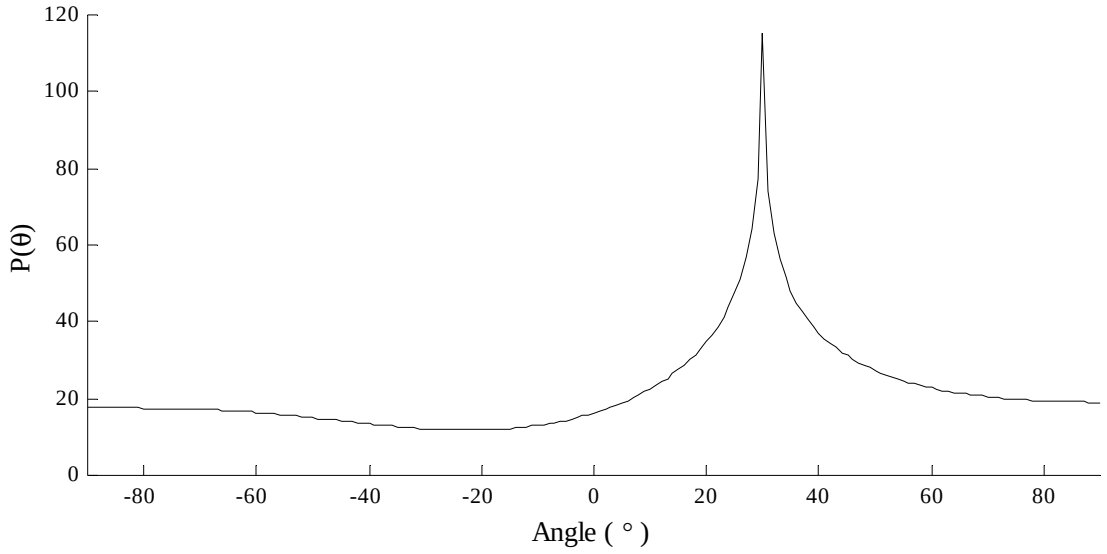

Figure 2.10 ***The*** $P(\theta)$ ***metric of MUSIC plotted against all possible angles of arrival showing a sharp peak at the correct DOA = 30°.***

2.4.1. Broadband Signal Considerations

The MUSIC algorithm can be extended so that it can be used for broadband signals in a manner very similar to what was done with the DSB based method. The dominant frequencies present in a broadband signal like speech can be picked and separate narrowband MUSIC algorithms can be run on each frequency to arrive at separate estimates. The mean of these independent estimates can be used as a good estimate of the DOA. Simulations were performed using a 4-element ULA with a spacing of 10 cm . The source signal was speech incident to the array at 22° with an SNR of 30 dB. The threshold used to pick dominant frequencies was 15 dB, which picked 13 frequencies. The mean value of the estimated DOA was 21.6°. To run the narrowband MUSIC algorithm, very narrow band-pass filters were used to extract the signals at each of the dominant frequencies. The pass-band of the filters was centered at the chosen frequencies and the width of the pass-band was set to 10 % of the frequency. This meant that the two –3 *dB* points on either side of the center frequency were separated by a distance of approximately 10 % of the center frequency. The magnitude response of one such filter centered at 2123 *Hz* is shown in Figure 2.11. The filter used was a simple second order filter of the form

$$H(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2.22}$$

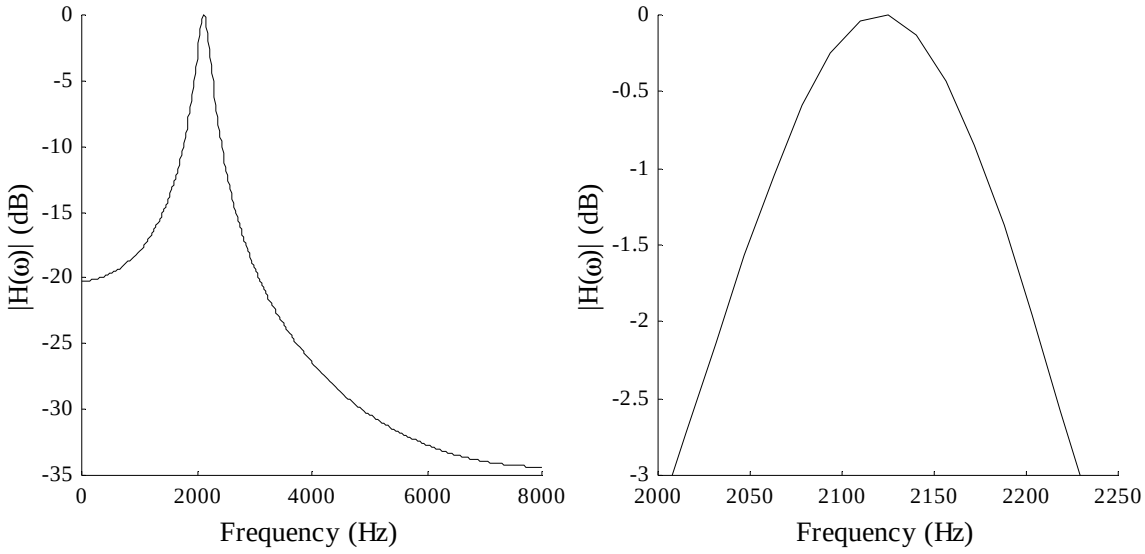

Figure 2.11 ***The narrow band-pass filter used to extract signals at F_c = 2123 Hz showing a pass-band of width approximately 220 Hz.***

The DOA estimates obtained from 13 dominant frequencies is shown in Figure 2.12. The estimates range from 19° to 23° and the mean value of the estimates is 21.69°.

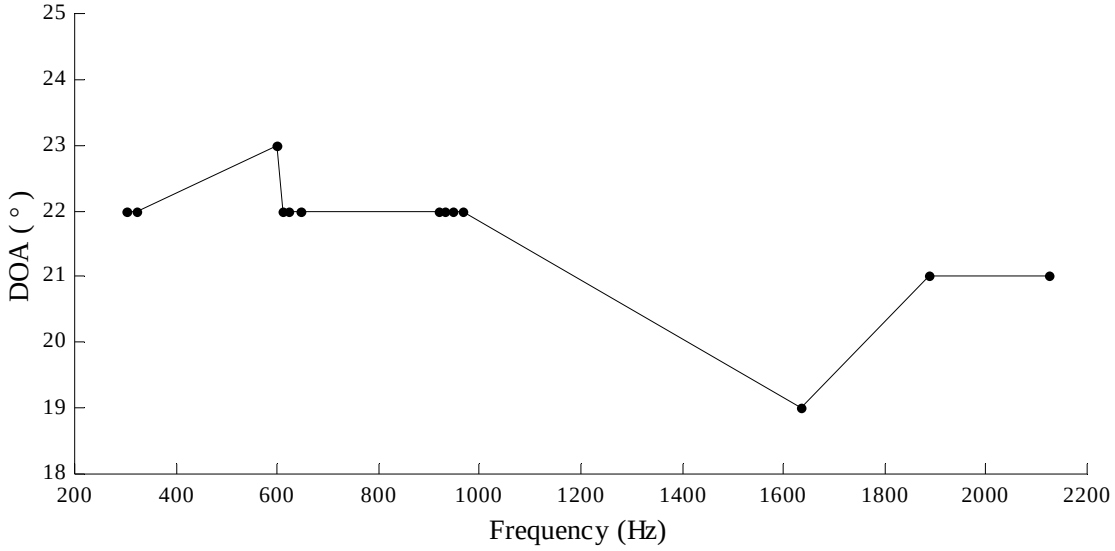

Figure 2.12 ***Estimated DOA against chosen formant frequency using MUSIC.***

Another method to combine the independent estimates obtained from the different frequencies is to sum the $P(\theta)$ metric across all the frequencies and pick the angles at which this

cumulative metric maximizes. Figure 2.13 shows the cumulative $P(\theta)$ plotted against possible angles and shows that it maximizes at the correct DOA of 22°.

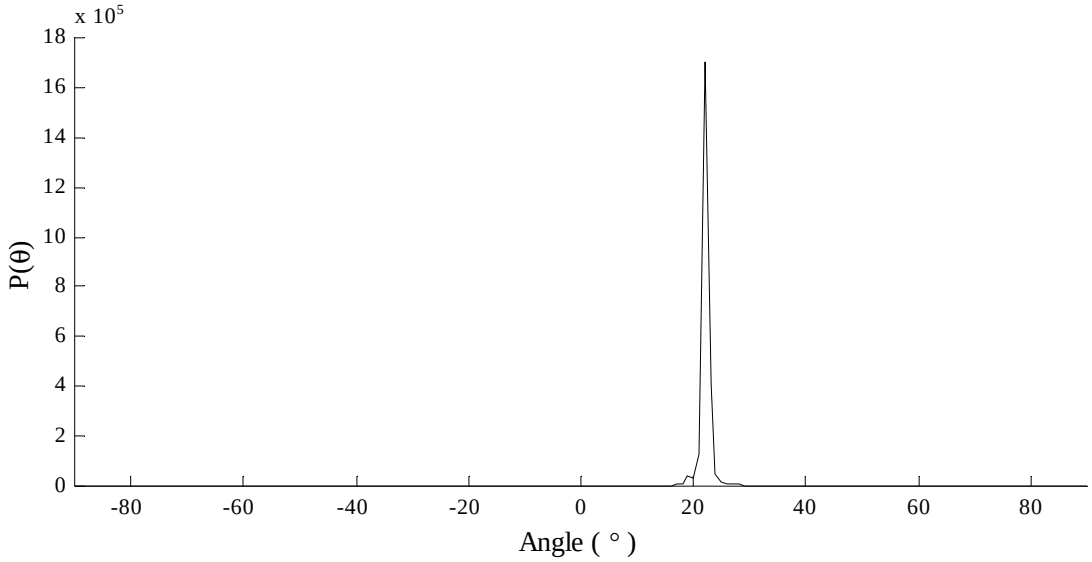

Figure 2.13 ***Cumulative $P(\theta)$ against possible angles showing a sharp peak at 22°.***

2.5. Time Delay Estimate Based Method

The third and final type of DOA estimation method consists of first computing the time delay estimates (TDE) between all pairs of microphones and then combining them, with the knowledge of the array geometry, to obtain the DOA estimate. In terms of computational requirements, the TDE based methods are the most efficient because they do not involve an exhaustive search over all possible angles. Also, TDE based methods are applicable directly to broadband signals. On the flip side, TDE based methods are useful only for the case of a single source impinging on the array. Computation of the time delay between signals from any pair of microphones can be performed by first computing the cross-correlation function of the two signals. The lag at which the cross-correlation function has its maximum is taken as the time delay between the two signals. Consider a ULA of N microphones with spacing between microphones equal to d. This array has a total number of microphone pairs equal to $\binom{N}{2}$ which is the number of combinations of N taken 2 at a time.

$$\binom{N}{2} = \frac{N!}{2!(N-2)!} \tag{2.23}$$

Consider any two microphones, *i* and *j*. Let the signals on these two microphones be $x_i(n)$ and $x_j(n)$ where *n* is a time-sample index. Let DFT samples of these signals be represented by $X_i(k)$ and $X_j(k)$ where *k* is a frequency-sample index. The cross-power spectral density (XPSD) between these signals is given by

$$\Phi_{x_i x_j}(k) = X_i(k) X_j^*(k) \tag{2.24}$$

The cross-correlation between these signals is given by the inverse DFT of the XPSD.

$$R_{x_i x_j}(l) = \frac{1}{M} \sum_{k=0}^{M-1} \Phi_{x_i x_j}(k) e^{j\frac{2\pi kl}{M}} \tag{2.25}$$

Here *M* is the length of the XPSD and *l* is a lag. $R_{x_i x_j}(l)$ can be computed for the range of possible negative and positive lags. The lag at which $R_{x_i x_j}(l)$ maximizes is the number of samples of delay between the two signals. Hence the time delay is given by

$$\tau_{ij} = \frac{1}{F_s} \arg\max\left(R_{x_i x_j}(l)\right) \tag{2.26}$$

Figure 2.14 shows the cross-correlation between speech signals impinging on two microphones from an angle of –60° from broadside. The microphones were separated by 10 *cm*, which results in a time delay of 251.02 μs.

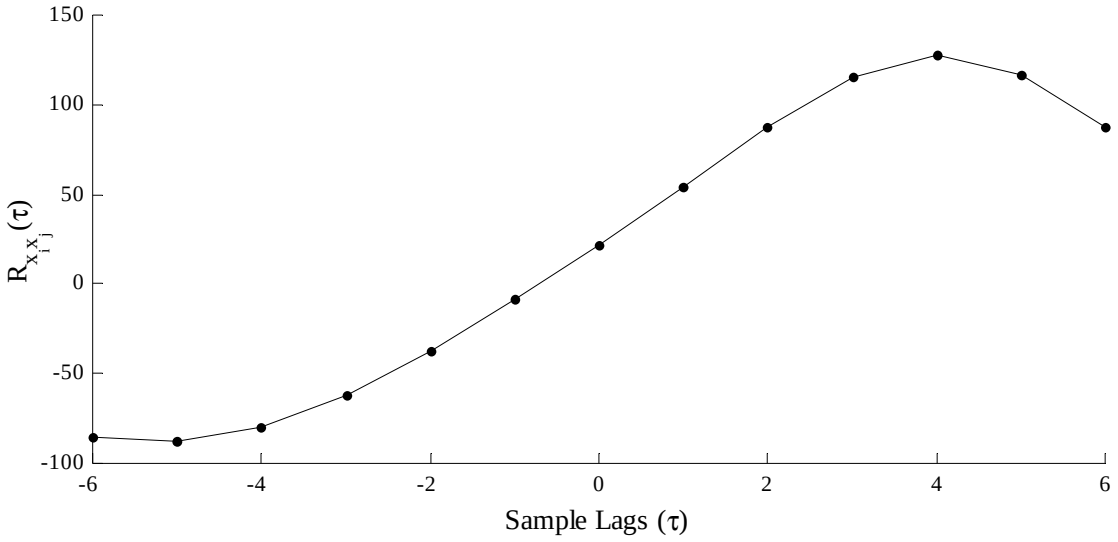

Figure 2.14 ***Cross correlation between two microphone signals with the source at -60°.***

The figure shows that the cross correlation maximizes at a delay of 4 samples, which corresponds to 250 μs at 16 *kHz* sampling. The figure only shows delays between –6 to +6 samples because the maximum delay that can be expected for this microphone separation is ±5 samples.

Time delays can be computed in a similar manner for all the possible microphone pairs. These time delays can be combined in a least squares sense to obtain the DOA. Let $\boldsymbol{\tau}$ be a $\binom{N}{2}\times 1$ vector that contains the time delays for all the microphone pairs. From (2.2), for each pair of microphones, *i* and *j*, the distance-time relationship is given by

$$d_{ij}\sin\theta = -v\tau_{ij} \tag{2.27}$$

Putting together this equation for all pairs of microphones, we get

$$\mathbf{d}\sin\theta = -v\boldsymbol{\tau} \tag{2.28}$$

Here **d** is, in general, a $\binom{N}{2}\times 1$ vector that contains the distances between each pair of microphones. This equation represents $\binom{N}{2}$ different equations that can be solved individually

to obtain DOA estimates. It is an over-determined system of equations where we have $\binom{N}{2}$ equations and one unknown. This system can be solved to obtain a least squares solution.

$$\sin\theta = \left(\mathbf{d}^T\mathbf{d}\right)^{-1}\mathbf{d}^T\left(-v\hat{\boldsymbol{\tau}}\right) \tag{2.29}$$

or

$$\theta = \sin^{-1}\left[\left(\mathbf{d}^T\mathbf{d}\right)^{-1}\mathbf{d}^T\left(-v\hat{\boldsymbol{\tau}}\right)\right] \tag{2.30}$$

We can solve for θ values between –90° and +90°. Table 2.1 shows the expected and estimated time delays for a simulated scenario.

Table 2.1 ***Expected and estimated time delays for a 4-element ULA and source at -60°.***

Pairs (*i, j)*	Expected Time Delay (μs)	Estimated Time Delay (μs)
1, 2	251.02	250.0
1, 3	502.04	500.0
1, 4	753.07	750.0
2, 3	251.02	250.0
2, 4	502.04	500.0
3, 4	251.02	750.0

Here a 4-element ULA with a spacing of 10 *cm* was used. The source was a speech signal coming from – 60° with respect to broadside. The signal was sampled at 48 *kHz* and had an SNR of 30 *dB*. The errors in the TDE are minimal and are a result of the discrete nature of the cross-correlation function. The least squares result obtained from this simulation was 59.6°. These time delays were computed from a signal frame of length 6000 samples. Zero padding was done on these array signals to make the DFTs twice the length of the signals.

3. Nature and Effects of Room Reverberation

In Chapter 2 we looked at several methods used to estimate the DOA of an acoustic source using a microphone array. These methods were developed with the assumption that there was no multipath in the received signal. The effects of multipath are encountered in a received signal when the source signal reflects off of surrounding objects and gets added to the direct path signal with a delay. The larger the number of surrounding objects, the more reflected signals are added to the direct path signal. For acoustic sources and microphone arrays placed inside a room, this effect can be quite large. The sound reflects off the walls, floor and ceiling of the room, multiple times, and these reflected signals get added to the direct signal. This effect is called room reverberation. Reverberation causes drastic changes to the time delay estimates derived from signals at the different microphones of an array. These changes are of a local nature with respect to time, which means that at certain instants of time there could be strong reflections and at certain other instants the reflections could be weak. Because of this, if we estimate the time delays using a short frame of signal data, the estimates keep changing over time. This introduces a significant challenge to algorithms performing DOA estimation.

3.1. Sound Generation and Propagation

Sound may be considered as a traveling wave that is generated by the vibrations of a plane surface that is in contact with a medium. The vibrations of the plane surface cause the layer of molecules of the medium close to the surface to compress and expand alternately. These compressions and expansions are then transferred to the next layer of molecules and so on. This way the sound generated by a vibrating body is transferred through a medium. At any point in time, the space surrounding the vibrating plane will consist of waves of compressed or expanded molecules of the medium. Such a space, which has moving sound in it, is called a sound field. The compressions and expansions of the medium at any point cause the pressure at that point to keep changing instantaneously. This variation in pressure at any point in the medium is what is heard as the sound signal. If the pressure varies in a purely sinusoidal manner, a single tone is heard. The sound is then said to have a single frequency. For pure sinusoidal sound, the distance between successive crests or troughs of the sinusoid is called the wavelength. The

wavelength is the distance traveled by the sound signal during one cycle of the sinusoid. For any propagating sinusoidal signal, the relationship between wavelength and frequency is given by

$$\lambda = \frac{v}{f} \tag{3.1}$$

Here λ is the wavelength in *m*, v is the velocity of sound in ms^{-1} and f is the frequency of the signal in *Hz*. The velocity of sound is, in general, a function of the characteristics of the medium such as its density, temperature and steady state pressure. Generally, sound is slowest in air and fastest in solids. At 20° C and at normal atmospheric pressure of 101 *kPa*, sound has a velocity of 344 ms^{-1} in air [14]. Another important property of sound is the amplitude of the signal. For a single tone this is the maximum change in pressure from the steady state value. All real sound signals can be thought of as being made up of a sum of sinusoids of varying frequencies, amplitudes and phases.

Consider a single tone sound wave that is propagating only in a single direction. This direction can be taken as the positive direction of the *x*-axis. Such a wave is called a plane wave because if we join all the points of equal pressure in the wave, we get a plane. Strictly speaking, plane waves can be generated only in controlled environments like narrow tubes and even then, only as an approximation. Most real waves are spherical waves where the sound waves emanate in all directions from the source. By joining all the points of equal pressure for such a wave, we get a sphere. A small section of a spherical wave that has propagated for a sufficient distance can be approximated as a plane wave because the curvature of the wave-front can be approximated by a plane. The wave equation for such a plane sound wave can be written as [15]

$$p(x,t) = p_0 \cos(\omega t - kx) \tag{3.2}$$

The sound pressure , $p(x,t)$, has been expressed as a function of both spatial location, *x,* and time, *t*. Here p_0 is the amplitude of the wave, ω is the radial frequency $(2\pi f)$ and k is the propagation constant given by

$$k = \frac{\omega}{v} \tag{3.3}$$

From (3.1) and (3.3) the relationship between the propagation constant (also called wave number) and the wavelength can be expressed as

$$\lambda = \frac{2\pi}{k} \tag{3.4}$$

3.2. Reflection of Sound from Rigid Surfaces

We will now consider the mechanics of the reflection of a planar sound wave from a flat rigid surface like a wall. Here we will assume that the wall is rough, but that the dimensions of this roughness are negligible compared to the wavelengths of the sound wave. Under such assumptions, the roughness of the wall can be neglected. When a sound wave hits a wall and reflects back the amplitude and phase of the wave change. Thus the reflection coefficient of the wall can be expressed as a complex quantity [15]

$$R = |R| e^{j\gamma} \tag{3.5}$$

The amplitude and phase of this complex reflection coefficient depend on the nature of the surface, frequency of the wave and the angle of incidence. The intensity (energy) of a plane wave is proportional to the square of the pressure amplitude of the wave. Thus the intensity of the reflected wave will be smaller by a factor $|R|^2$. The fraction of energy that was lost in the reflection is $1-|R|^2$. This quantity is called the absorption coefficient of the wall.

$$\alpha = 1-|R|^2 \tag{3.6}$$

A wall with R = 0 is said to be totally absorbent or "matched to the sound field". A wall with R = 1 is said to be "hard" and one with R = -1 (phase reversal) is said to be soft. Now consider a single tone plane sound wave moving in the positive x direction towards a perpendicular rigid wall. From (3.2) the equivalent complex analytic sound wave can be expressed as

$$p_i(x,t) = p_0 e^{j(\omega t - kx)} \tag{3.7}$$

The reflected sound wave can now be written as [15]

$$p_r(x,t) = Rp_0 e^{j(\omega t + kx)} \tag{3.8}$$

The effect of the reflection is incorporated in the multiplication by the complex reflection coefficient. The change in direction is incorporated by a change in sign for the spatial term within the exponent. Now consider a plane wave that is traveling at an angle θ to the x-axis as shown in Figure 3.1. This wave can be expressed as

$$p_i(x,y,t) = p_0 e^{j\left(\omega t - k\left[x\cos(\theta) + y\sin(\theta)\right]\right)} \tag{3.9}$$

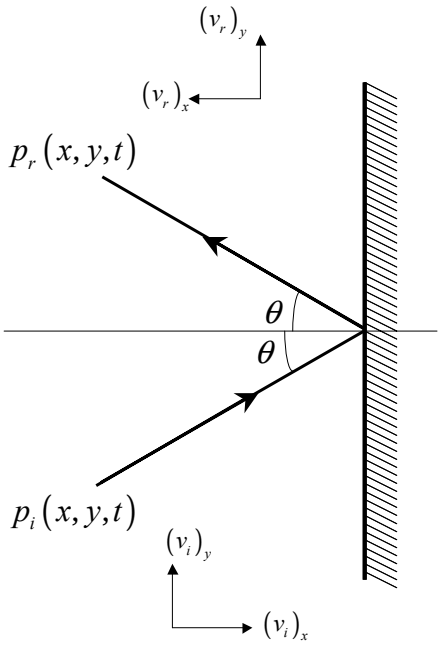

Figure 3.1 ***Plane wave reflecting at an angle to the wall.***

This expression can be obtained by rotating the x-axis by θ to line it up with the wave and following a co-ordinate transformation procedure [15]. The reflected wave for this case can be written as

$$p_r(x,y,t) = Rp_0 e^{j\left(\omega t - k\left[-x\cos(\theta) + y\sin(\theta)\right]\right)} \tag{3.10}$$

Again the effect of the reflection is taken care of by the multiplication with R and the change in direction is taken care of by the change in the sign of the x term within the exponent.

Note that the reflection of sound waves follows the well-known law of reflection where the angle of incidence is equal to the angle of reflection.

3.3. Geometrical Room Acoustics

The discussion in Section 3.2 was based on the wave model of sound. When considering the sound field in an enclosed room, the use of the wave model can become quite challenging. Apart from considering the effect of superposition of numerous reflected waves, one also needs to take into account the particle velocity normal to the wall of reflection. This effect, which is characterized by the specific impedance of the wall, has not been considered in the discussion in Section 3.2. A simpler approach to take is to take the limiting case of very small wavelengths (high frequencies) and thus replace the sound wave with a sound ray and then use geometrical acoustics. This simplification is justified for wavelengths that are arbitrarily small when compared to the dimensions of the room and distances traveled by the sound wave. For frequencies around the medium range (1000 *Hz*, 34 *cm* wavelength) this approximation is valid for typical rooms. Several other assumptions are made when using this approach. The sound ray originates from a certain point and has a well-defined direction of propagation. It has a finite velocity of propagation and follows the law of reflection when it encounters a rigid wall. The medium in the room is assumed to be homogeneous, i.e. there are no sudden changes in density in the medium, and thus refraction is assumed to be non-existent and the sound rays travel in straight lines until they encounter reflecting walls. Also, since sound rays do not change directions while traveling in the medium, diffraction is also assumed to be non-existent.

Under these circumstances there are three effects that determine the acoustics of a room, viz. finite velocity of sound, absorption of sound energy by the walls during each reflection and absorption of sound energy by the medium. The finite velocity of sound causes reflected signals to arrive at the listener with finite delays and these signals get added to the original source signal. The reflections at walls can be simplified by making the reflection coefficients real valued and independent of either frequency or angle of incidence. A mean value can be used to represent reflection coefficients for all frequencies and all angles. The propagation of sound in a medium is not ideal. Sound is transmitted from one layer of the medium to the next by mechanical collisions between adjoining molecules. These collisions are not ideal and some energy is lost as heat, which goes towards increasing the temperature of the medium. This loss of energy is

proportional to the distance traveled by the ray. In other words, the intensity of the sound ray is inversely proportional to the square of the total distance traveled.

We can now look at a source and a listener (or a microphone) placed inside a room together as forming a linear time invariant (LTI) system. The source signal is the input to the system and the signal picked up by the microphone is the output of the system. Any LTI system is characterized by its impulse response. We now attempt to model the room system using an impulse response. If the walls of the room were completely absorbent, then the source signal would arrive at the microphone after a delay corresponding to the time taken by the sound to travel from the source to the microphone. In this case the impulse response can be represented by a single impulse at the appropriate delay. The amplitude of this impulse would be inversely proportional to the distance between the source and the microphone. Note that since intensity (energy) is inversely proportional to the square of the distance traveled, amplitude should be inversely proportional to the distance traveled. Real walls reflect the sound rays, which also end up arriving at the microphone with different delays. This gives rise to reverberation. Reverberation is the effect felt by listening to a succession of the same signals arriving at different delays with gradually decreasing intensities. In the impulse response, each arrival of a reflected ray can be represented by an impulse at the appropriate delay. The delays at which these impulses occur depend on the total path lengths of the reflected rays. The amplitudes of these impulses depend again on the total path length and also on the number of reflections that the reflected rays had to go through. In the next section we discuss an efficient method to compute these path lengths and the number of reflections.

3.4. Image Model of The Source

Figure 3.2 shows a sound source S located near a rigid reflecting wall. The destination D gets two signals, one from the direct path and a second one from the reflection. The path length of the direct path can be directly calculated from the known locations of the source and the destination. Also shown is an image of the source, S', located behind the wall at a distance equal to the distance of the source from the wall. Because of symmetry, the triangle SRS' is isosceles and therefore the path length $SR + RD$ is the same as $S'D$.

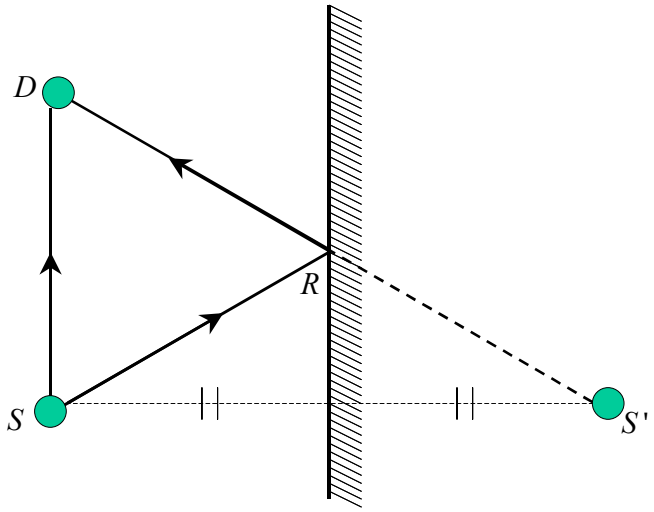

Figure 3.2 ***A source and its image.***

To compute the path length of the reflected path, we can construct an image source and compute the distance between destination and image source. Also, the fact that we are computing the distance using an image means that there was one reflection in the path. Figure 3.3 shows a path involving two reflections. The length of this path can be obtained from the length of $S"D$. In Figure 3.4 the length of a path involving three reflections is obtained from the length of $S'''D$. These figures can also be extended to three dimensions to take into account reflections from the ceiling and the floor.

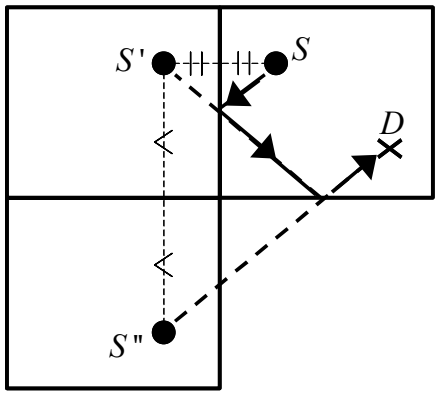

Figure 3.3 ***Path involving two reflections obtained using two levels of images.***

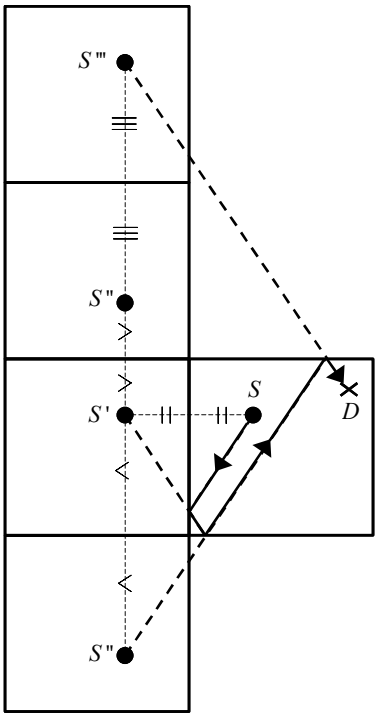

Figure 3.4 ***Path involving three reflections obtained using three levels of images.***

In general the path lengths (and thus the delays) of reflections can be obtained by computing the distance between the source images and the destination. The strength of the reflection can be obtained from the path length and the number of reflections involved in the path. The number of reflections involved in the path is equal to the level of image that was used to compute the path.

3.5. Simulation of Reverberation

The image model can be used to simulate the reverberation in a room for a given source and microphone location. The system is treated as an LTI system whose impulse response consists of a set of delayed impulses of gradually decreasing amplitudes. Allen and Berkley developed an efficient method [16], using the image model, to compute such an impulse response for rectangular rooms. Consider a rectangular room with length, width and height given by L_x, L_y and L_z. Let the sound source be at a location represented by the vector $\mathbf{x_s} = [x_s \quad y_s \quad z_s]$ and let the microphone be at a location represented by the vector $\mathbf{x_m} = [x_m \quad y_m \quad z_m]$. Both vectors are with respect to the origin, which is located at one of the

corners of the room. The vector joining the microphone to any of the first level images can be written as

$$\mathbf{R_p} = \left[x_s - x_m + 2qx_m \quad y_s - y_m + 2jy_m \quad z_s - z_m + 2kz_m \right] \tag{3.11}$$

Each of the elements in the triplet $p = (q, j, k)$ can take on values 0 or 1. When the value of p is 1 in any dimension, then an image of the source in that dimension is considered. To consider images of any level, we add the vector $\mathbf{R_r}$ to $\mathbf{R_p}$ where

$$\mathbf{R_r} = 2\left[nL_x \quad lL_y \quad mL_z \right] \tag{3.12}$$

Each of the elements of the triplet $r = (n, l, m)$ take on values between -N and +N, depending on the maximum level of images that we would like to consider. In all the simulations performed for this research the value of N was set to 5. The distance between any source image and the microphone can be written as

$$d = \left| \mathbf{R_p} + \mathbf{R_r} \right| \tag{3.13}$$

The time delay of arrival of the reflected sound ray corresponding to any image source can thus be expressed as

$$\tau = \frac{\left| \mathbf{R_p} + \mathbf{R_r} \right|}{v} \tag{3.14}$$

The impulse response for this source and microphone location can now be written as [16]

$$h(t, \mathbf{x_s}, \mathbf{x_m}) = \sum_p \sum_r R_{x1}^{|n-q|} R_{x2}^{|n|} R_{y1}^{|l-j|} R_{y2}^{|l|} R_{z1}^{|m-k|} R_{z2}^{|m|} \frac{\delta\left(t - \left(\frac{\left| \mathbf{R_p} + \mathbf{R_r} \right|}{v} \right) \right)}{4\pi \left| \mathbf{R_p} + \mathbf{R_r} \right|} \tag{3.15}$$

The quantities R_{x1}, R_{x2}, R_{y1}, R_{y2}, R_{z1} and R_{z2} are the reflection coefficients of the six walls. The elements of the triplet p are 0 or 1, which means that there are 8 different combinations, (0 0 0) to (1 1 1). The elements of the triplet r range from –N to +N, which means

that there are $(2N+1)^3$ combinations. Therefore, for a given N, this method computes $8(2N+1)^3$ different paths. The delays of the impulses corresponding to these paths are computed using (3.14) and the strengths of these impulses are multiplied by reflection-coefficients as many times as there are reflections. Once the impulse response has been computed this way, the source signal can be convolved with the impulse response to simulate the signal picked up by the microphone.

An important consideration while simulating the discrete version of this impulse response using a computer is that the delays given by (3.14) do not always fall at sampling instants. One way to get around this problem is to compute the discrete impulse response at a much higher sampling frequency, convolve the interpolated source signal with it, and then decimate the resultant signal to the original sampling frequency. This increases the computational load since the convolution at the higher sampling frequency involves larger data sequences. Peterson suggested a modification to this method [17]. In this approach, each impulse in (3.15) is replaced by the impulse response of a Hanning-windowed ideal low pass filter of the form

$$h_{lp}(t) = \begin{cases} \frac{1}{2}\left[1+\cos\left(\frac{2\pi t}{T_w}\right)\right]\mathrm{sinc}(2\pi F_c t), & -\frac{T_w}{2} < t < \frac{T_w}{2} \\ 0, & \textit{otherwise} \end{cases} \tag{3.16}$$

T_w is the width (in time) of the impulse response and F_c is the cut-off frequency of the low-pass filter. For the simulations performed in this research T_w was set to 15 *ms* and F_c was set to 90 % of the Nyquist frequency. Each impulse in (3.15) is replaced by $h_{lp}(t)$ centered at the true delay as shown in Figure 3.5. Only the samples of $h_{lp}(t)$ at the sampling instants are used here, but $h_{lp}(n)$ still has its centroid located at the correct delay of 20.3 samples. By doing this, true delays of arrival of the reflected signals are simulated accurately even at the original low sampling frequency.

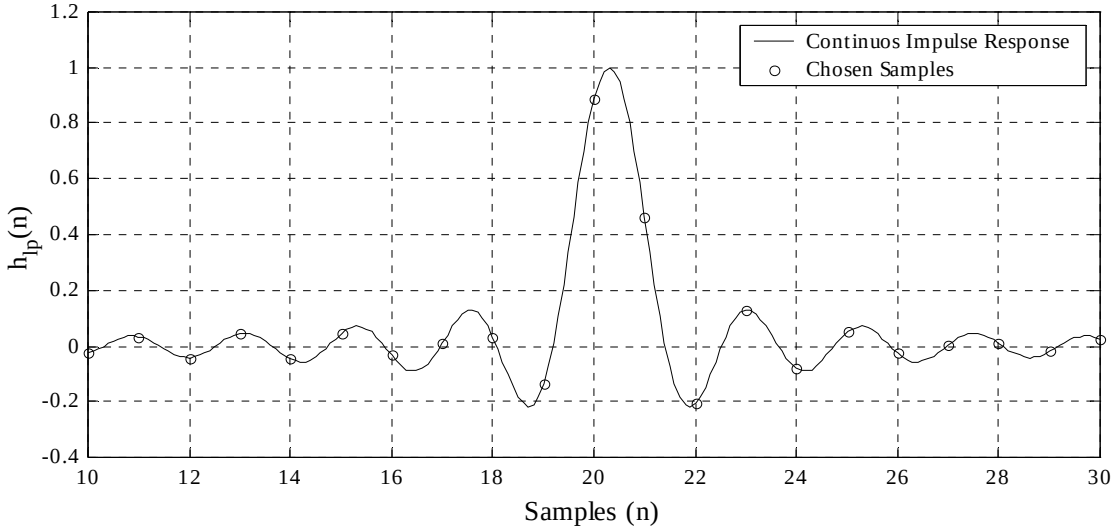

Figure 3.5 ***Peterson's low-pass impulse response centered at a delay of 20.3 samples.***

The other consideration while simulating reverberation for a room is the duration of reverberation or the reverberation time. Formally, the reverberation time is defined as the time required for the intensities of reflected sound rays to be down 60 dB from the direct path sound ray. An empirical formula, known as Eyring's formula [18] can be used to relate the reverberation time, T_r, to the reflection coefficient, R, of the walls.

$$T_r = \frac{-13.82}{\left[v\left(L_x^{-1} + L_y^{-1} + L_z^{-1}\right) \ln\left(R\right) \right]} \tag{3.17}$$

In (3.17) the reflection coefficients of all the walls and the floor and the ceiling are assumed to be the same. Figure 3.6 shows the simulated speech signals arriving at two microphones that are placed in a room without any reverberation. The room is $5m \times 3m \times 3m$ in dimensions and the speaker was placed at $\begin{bmatrix} 0.5 & 0.05 & 1.5 \end{bmatrix}^T$ with respect to one of the corners of the room. Mic-1 was located at $\begin{bmatrix} 4.5 & 1.45 & 1.5 \end{bmatrix}^T$ and Mic-2 was located at $\begin{bmatrix} 4.5 & 1.65 & 1.5 \end{bmatrix}^T$, which is a distance of 20 *cm* from Mic-1. All these vectors are expressed in meters.

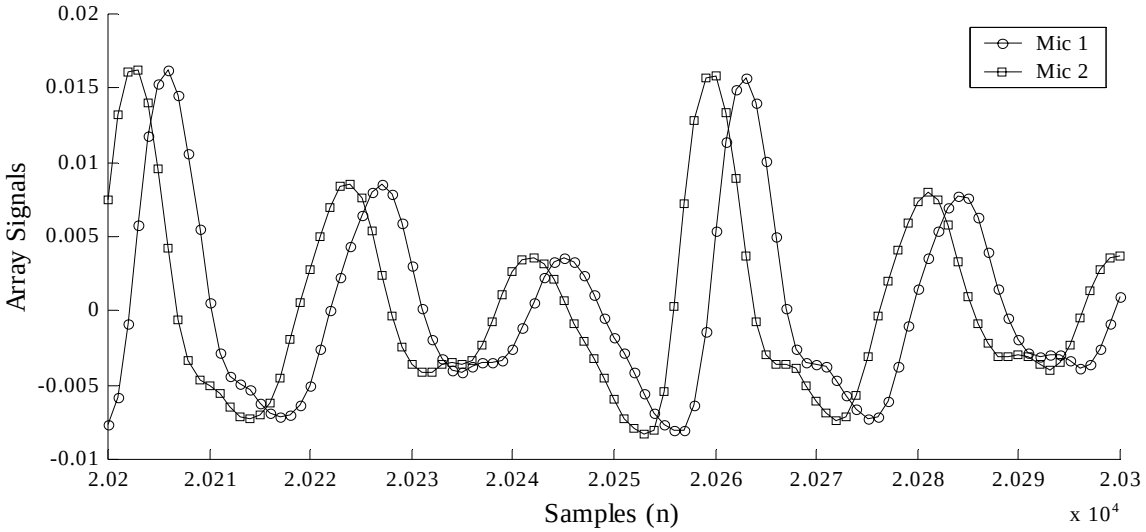

Figure 3.6 ***Signals at two microphones simulated without reverberation.***

Figure 3.7 shows the simulated speech signals arriving at the microphones when the room was assumed to have a reverberation time of 100 *ms.* The reflection coefficients of all the walls and the floor and ceiling were assumed to be the same and was computed using (3.17). The same sections of the signals without reverberation are shown as in Figure 3.6 for comparison purposes.

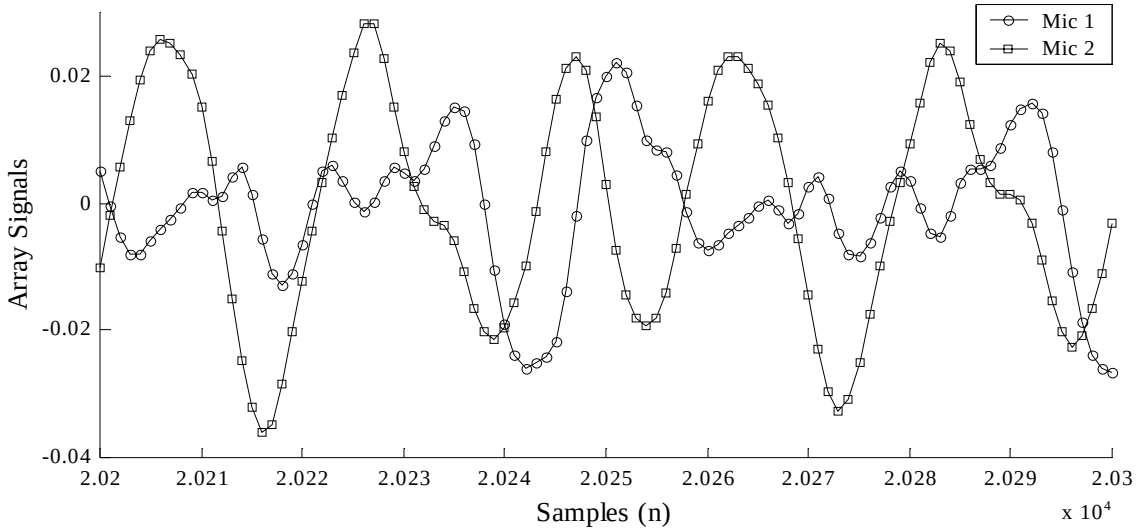

Figure 3.7 ***Signals at two microphones simulated with 100 ms reverberation.***

The most important feature of the reverberant signals is that the uniform time delay between the two signals has been lost. The time delay between the two signals appears to be not

constant. Figure 3.8 shows the simulated impulse response for Mic-1. The impulse response shows strong impulses at delays where reflected sound rays arrive at the microphone. The strengths of these reflections are seen to gradually decrease over time.

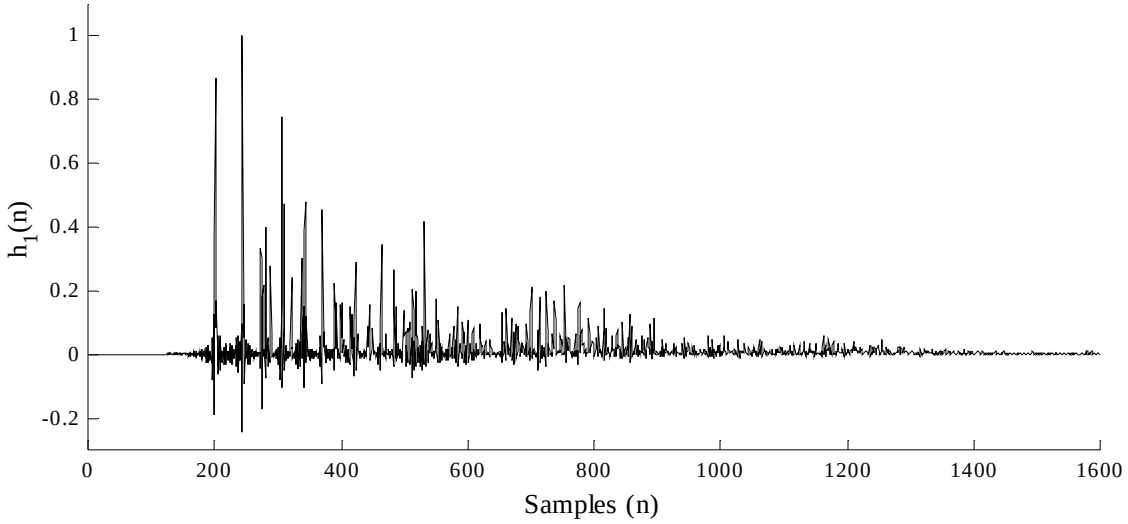

Figure 3.8 ***Simulated impulse response for Mic-1.***

3.6. Measurement of Room Reverberation

Now we look at how we can measure the acoustics of a room. As was discussed in Section 3.5, the path between an acoustic source and a microphone, both of which are located in a reverberant room, can be modeled as a linear system with a certain impulse response. In general the impulse response is a function of the source location, microphone location, the dimensions of the room and the reflection properties of the walls of the room. The measurement can be performed in two ways. In the first method we directly record the signal picked up by the microphone after sounding a short pulse approximating an impulse. In the second method white noise is sounded and the signal picked up on the microphone is correlated with the source signal.

3.6.1. Measurement Using Narrow Pulses

In this method, narrow pulses are sounded through a loudspeaker. The microphone in the array records the actual response between the source and itself to that narrow pulse. Under the assumption that the narrow pulse that was sounded was a good approximation of an impulse, the recorded response can be assumed to the impulse response of the system. The loudspeaker and the microphone were placed in a room in the DSP Research Lab. (DSPRL). The room had

approximate dimensions of $3.85m \times 2.8m \times 2.45m$. The narrow pulse that was sounded was generated on a computer and was of a width of one sample. The signal was played at 4 *kHz* sampling frequency. Figure 3.9 shows one such measured impulse response as recorded by one of the microphones. Let $\mathbf{s}$ be a vector representing the location of the source and let $\mathbf{m}_i$ be a vector representing the location of the i^{th} microphone of the array. The impulse response from the source to the i^{th} microphone can be written as $h_{\mathbf{s},\mathbf{m}_i}(n)$.

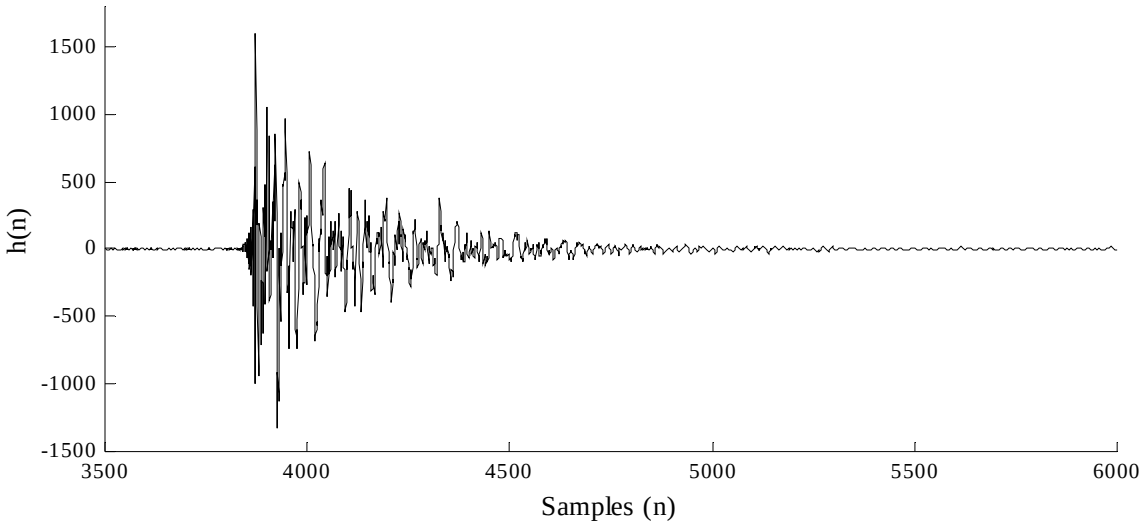

Figure 3.9 ***Recorded impulse response.***

Reverberation time of a room is formally defined as the time taken for the reflections to go down 60 *dB* in energy. Figure 3.10 shows the energy of the recorded impulse response in *dB*. The figure shows a clear noise floor of *–50 dB*. So in this case we will take the reverberation time as the time required for the reflections to go down to –50 *dB*. By this definition the reverberation time was measured to last for 2527 samples, which, at 8 *kHz* sampling, amounts to a reverberation time of approximately 316 *ms*.

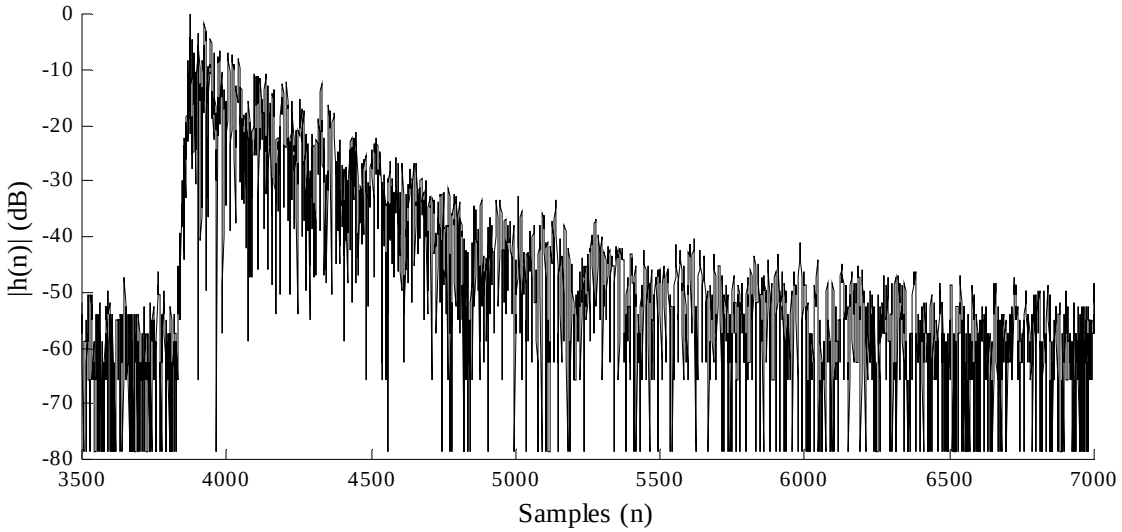

Figure 3.10 ***Energy of the recorded impulse response in dB.***

3.6.2. Measurement Using White Noise Input

As we have seen before, a reverberant room can be modeled as a linear time invariant system with impulse response $h(n)$, and our aim is to estimate $h(n)$. Let the system be excited with white noise - $e(n)$ and let the measured response of the system be $y(n)$. In all these expressions the index n is a discrete time-sample index. The response can now be written as a convolution of the input white noise with the system impulse response, as follows.

$$y(n) = e(n) * h(n) \tag{3.18}$$

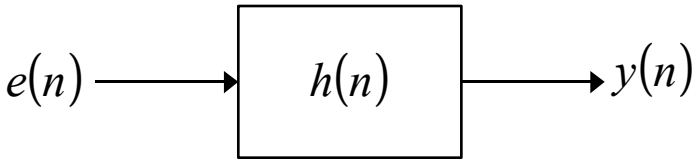

Figure 3.11 ***A linear time invariant system excited with white noise.***

Consider the cross-correlation of the output response with the input excitation given by

$$R_{ye}(\tau) = E\left[y(n) e(n-\tau) \right] \tag{3.19}$$

From a finite observation window of M samples of the input and output signals, we can compute an estimate of the cross-correlation given in (3.19) as

$$\hat{R}_{ye}(\tau) = \frac{1}{M}\sum_{n=0}^{M-1} y(n)e(n-\tau) \tag{3.20}$$

Substituting (3.18) in (3.20) we get

$$\begin{aligned}
\hat{R}_{ye}(\tau) &= \frac{1}{M}\sum_{n=0}^{M-1}\Big[h(n)*e(n)\Big]e(n-\tau) \\
&= \frac{1}{M}\sum_{n=0}^{M-1}\left[\sum_{k=0}^{N-1} h(k)e(n-k)\right]e(n-\tau) \\
&= \sum_{k=0}^{N-1} h(k)\left[\frac{1}{M}\sum_{n=0}^{M-1} e(n-k)e(n-\tau)\right] \\
&= \sum_{k=0}^{N-1} h(k)\hat{R}_{ee}(\tau-k)
\end{aligned} \tag{3.21}$$

$\hat{R}_{ee}(\tau)$ is an estimate of the auto-correlation of the white noise input from an observation window of *M* samples. Theoretically, $\hat{R}_{ee}(\tau)$ can be replaced with a delta function at τ. Thus we have

$$\begin{aligned}
\hat{R}_{ye}(\tau) &= \sum_{k=0}^{N-1} h(k)\delta(\tau-k) \\
&= h(\tau)*\delta(\tau) \\
&= h(\tau)
\end{aligned}$$

Thus we have

$$h(n) = \hat{R}_{ye}(n) \tag{3.22}$$

Equation (3.22) states that we can estimate the impulse response of an LTI system by measuring the response of the system to a white noise input excitation and computing an estimate of the cross-correlation of the response with that input excitation. This principle can be used to measure the impulse response of a reverberant room. White noise is sounded through a speaker and one microphone is placed right in front of the speaker. This microphone measures the input excitation of the LTI system. At the same time the microphones that form the array measure the signals reaching them. The source and each microphone in the array form separate

LTI systems. Thus the signals reaching each microphone in the array act as the responses of the respective LTI systems. The impulse response from the source to the i^{th} microphone in the array can be written as

$$h_{\mathbf{s},\mathbf{m_i}}(n) = \hat{R}_{x_i,s}(n) \tag{3.23}$$

where $\hat{R}_{x_i,s}(n)$ is the cross-correlation between the signal, $x_i(n)$, recorded by the i^{th} microphone and $s(n)$, which is the signal recorded by the microphone close to the source. Figure 3.12 shows the measured impulse response at one of the microphones in the array. The figure shows impulses at delays when reflected signals are received. This impulse response was computed using 2048 samples of recorded white noise. Hence the impulse response could only be computed for 1024 samples. The first impulse is at a delay of 33 samples (4.125 *ms*), which corresponds to the time taken by sound to travel the direct path length, which is the distance between the source and the microphone (1.46 *m*). This calculated distance agrees well with the distance that was measured using a ruler (1.43 *m*).

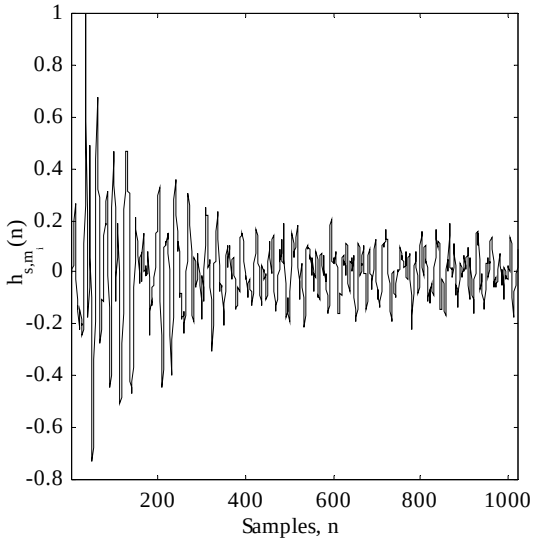

Figure 3.12 ***Impulse response measured with white noise.***

3.6.3. Comparison of Measurements

Now we will attempt to compare the results obtained from the two methods of measurement and thus try to find possible sources of error in these measurements. Figure 3.13

shows a plot of the first 350 samples of the estimated impulse responses measured by the two methods. We observe that the locations of the peaks (which correspond to reflections) more or less match up. The relative strengths of the peaks do not always match up. This difference can be attributed to two imperfections in the measurement setup. Figure 3.14 shows the source impulse signal and the Fourier transform of the noise signal that was used to make these measurements. Notice that the impulse signal is not an ideal impulse. It is not even an ideal sinc pulse, which is what one would expect from a D/A conversion. It is a sinc pulse, of which one side has been distorted. We also observe that the noise signal that was sounded for the measurement was not truly white. This can cause the measurement to contain errors. In summary, we have measured the room impulse response using two different methods and have obtained results that are fairly consistent with each other. Some inconsistencies were observed between the two measurements, and attributed to the non-ideal nature of the sources used for making the measurements.

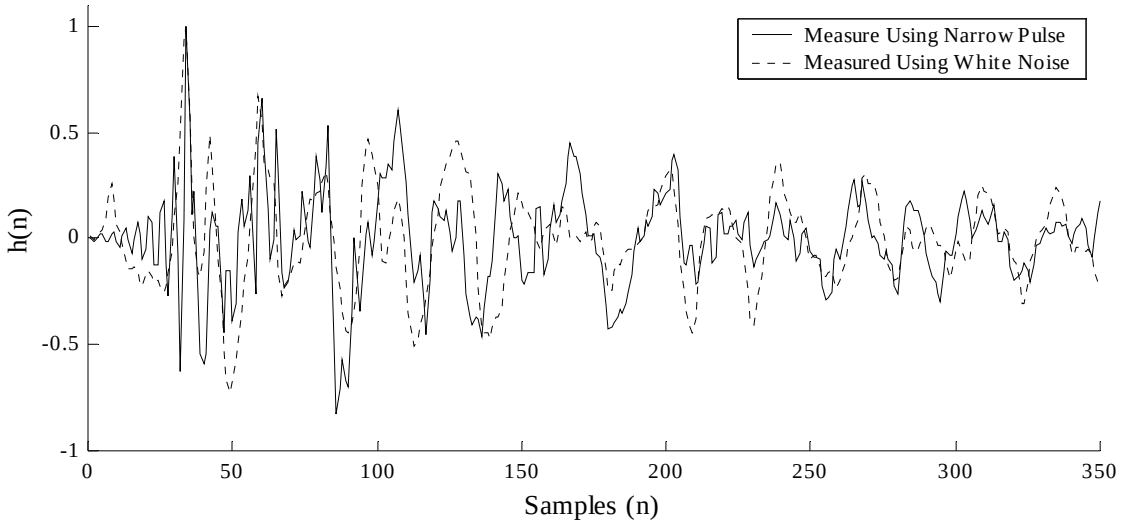

Figure 3.13 ***Normalized impulse responses measured by the two methods.***

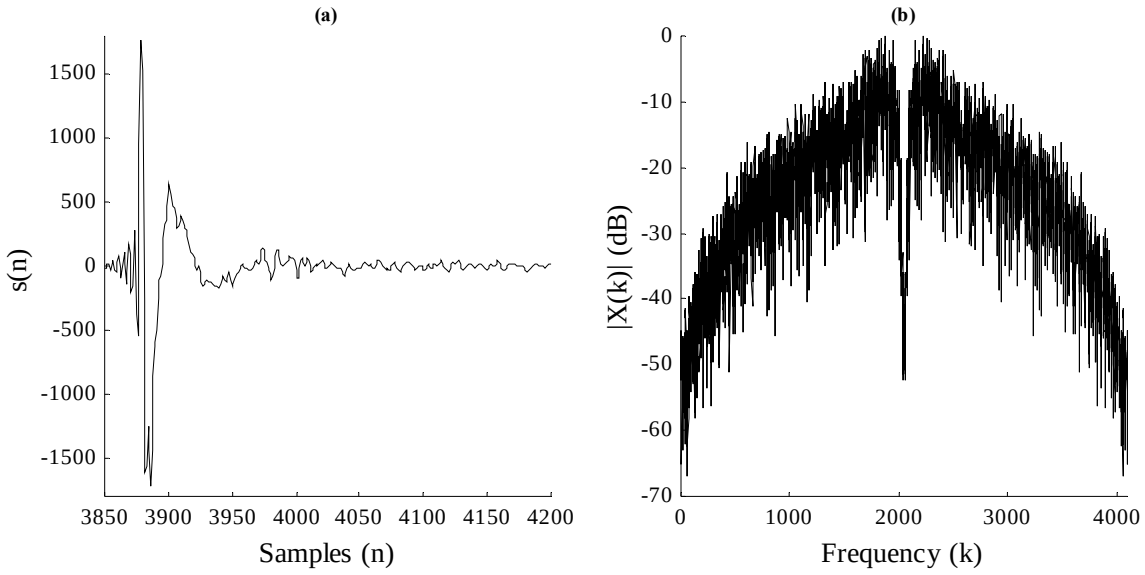

Figure 3.14 ***Imperfections in the measurement setup (a) Non-ideal impulse, (b) Non-white noise source.***

3.7. Effect of Reverberation on DOA Estimation Techniques

We can now look at the effect of reverberation on the DOA estimation techniques that were discussed in Chapter 2. We will again use the same room that was used in Section 3.5 with dimensions $5m \times 3m \times 3m$. The source is again placed at $\begin{bmatrix} 0.5 & 0.05 & 1.5 \end{bmatrix}^T$ with respect to one of the corners of the room. We will be using a 4-element ULA with a spacing of 10 *cm* to perform the DOA estimation. The microphones are located at the following locations:

$$Mic-1: \quad \begin{bmatrix} 4.5 & 1.35 & 1.5 \end{bmatrix}^T$$

$$Mic-2: \quad \begin{bmatrix} 4.5 & 1.45 & 1.5 \end{bmatrix}^T$$

$$Mic-3: \quad \begin{bmatrix} 4.5 & 1.55 & 1.5 \end{bmatrix}^T$$

$$Mic-4: \quad \begin{bmatrix} 4.5 & 1.65 & 1.5 \end{bmatrix}^T$$

This setup sets the true DOA to -19.93°.

A speech signal sampled at 16 *kHz* was used as a source and the signals at each microphone were simulated assuming a reverberation of 100 *ms* for the room. The SNR of the simulated signal was set to 30 *dB*. The signals from the microphones were divided into frames

of 512 samples each (32 *ms*) and DOA estimates were performed using all three methods for all the frames. For the case of MUSIC, multiple metrics were obtained for several different frequencies. These metrics were added and the angle at which the sum maximized was used as an estimate of the DOA. The same procedure was followed for the DSB based method where the PSD obtained for several different frequencies were added and the angle at which this cumulative PSD maximized was used as an estimate of the DOA.

Figure 3.15 shows the results obtained from this simulation. For comparison purposes Figure 3.16 shows the results of the simulation for the same setup except that this time the room was assumed to have no reverberation. Only frames with energy greater than or equal to 1×10^{-6} were used for the simulations. This is the reason why we get 91 frames for the case with reverberation and 58 frames for the case without reverberation. The presence of multiple reflections increases the power of the reverberated signals.

Clearly the presence of reverberation degrades the performance of all three methods. Table 3.1 lists the standard deviations and means of the estimated DOA for all three methods for both scenarios. The table shows two things. First there is far more variation around the mean in the presence of reverberation. Second, there is a bias towards 0 in the estimates in the presence of reverberation.

Table 3.1 ***Standard deviations and means of DOA estimates over all frames.***

Method	Standard Deviation		Mean	
	$T_r = 100$ *ms*	$T_r = 0$ *ms*	$T_r = 100$ *ms*	$T_r = 0$ *ms*
MUSIC	15.51	5.03	-11.88	-18.41
DSB	14.67	5.18	-11.79	-19.23
TDE	6.75	0.59	-12.69	-20.17

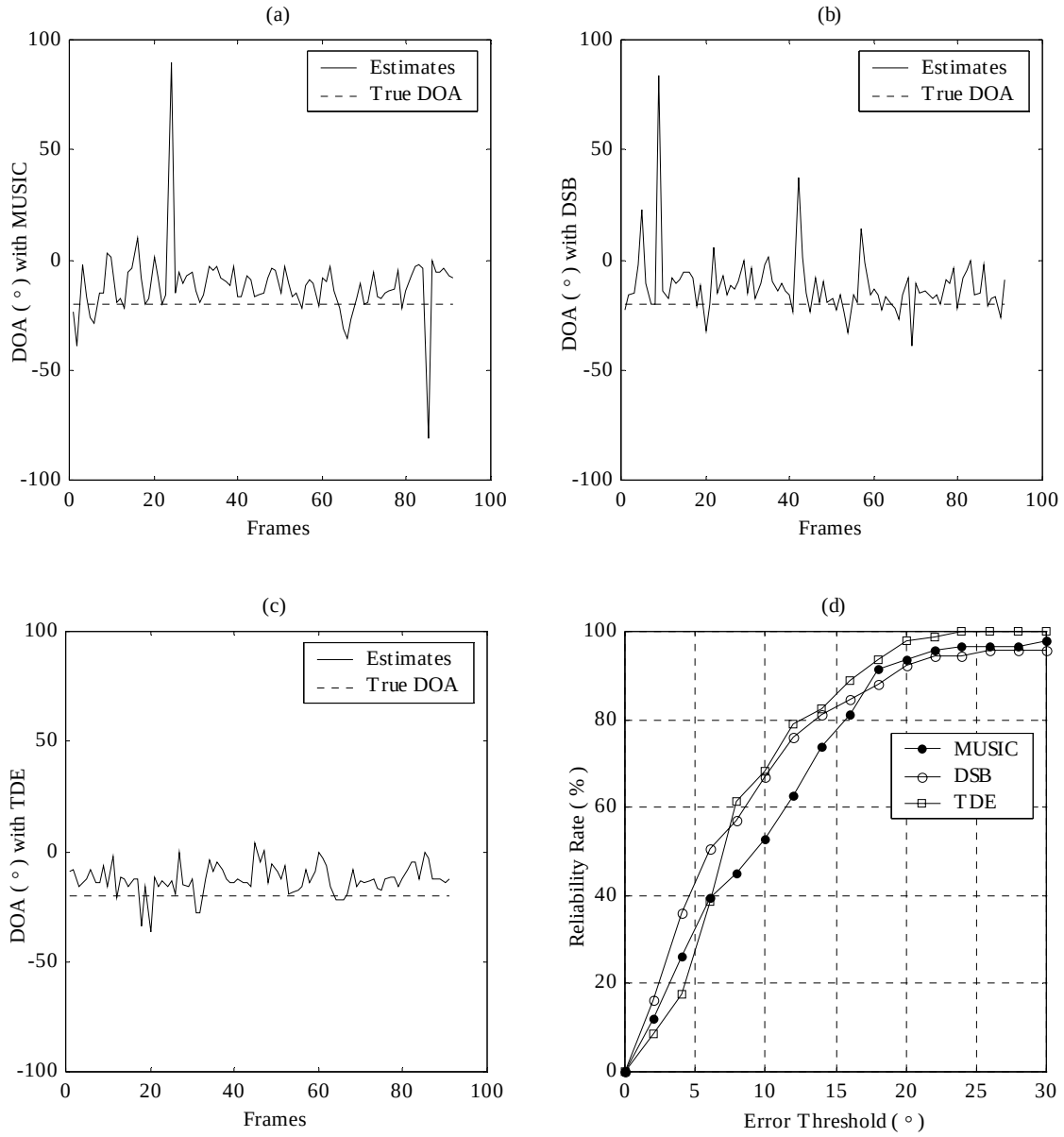

Figure 3.15 ***Framewise DOA estimates using (a) MUSIC, (b) DSB and (c) TDE for 100 ms reverberation time and (d) reliability-rates.***

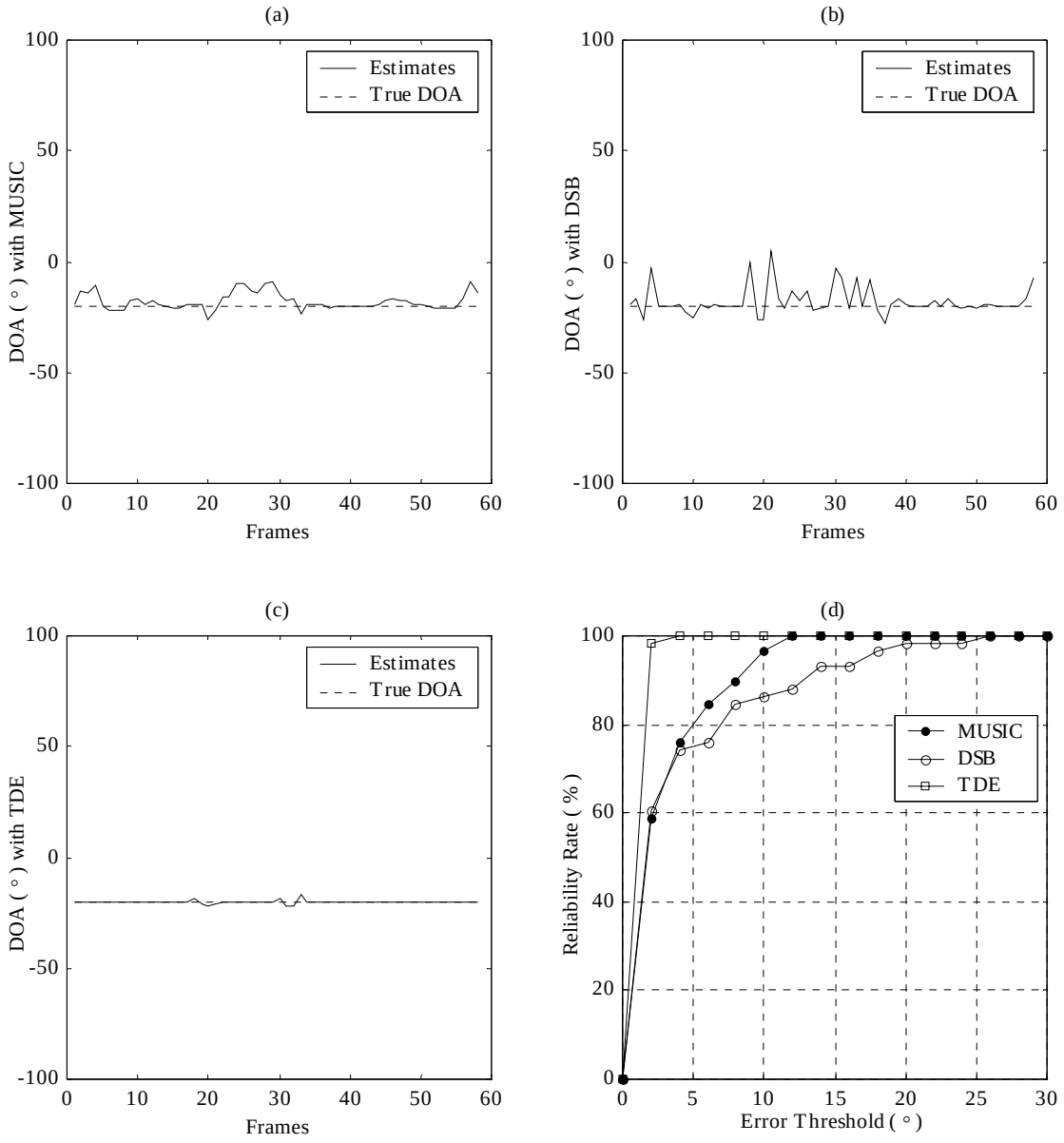

Figure 3.16 ***Framewise DOA estimates using (a) MUSIC, (b) DSB and (C) TDE and (d) reliability-rates with no reverberation.***

The reliability-rate plots that are shown for both these cases give us a useful tool to determine the reliability of the different methods. The *x*-axis represents various error thresholds and the *y*-axis represents the percentage of DOA estimates that were within the error threshold. For example, in the case of 100 *ms* reverberation time we can see that approximately 50 % of the estimates using MUSIC have errors less than 10° whereas approximately 70 % of the estimates using either DSB or TDE have errors less than 10°. Compare this with the case with no

reverberation where approximately 95 % of the results with MUSIC, 85 % of the results with DSB and 100 % of the results with TDE have errors less than 10°.

These results clearly show that there is a need to improve upon the traditional DOA estimation techniques to get reliable results in the presence of reverberation. In the next chapter we show how such improvements can be achieved using the phase transform.

4. Application of the Phase Transform to DOA Estimation

In Chapter 3 we looked at the nature of reverberation and the effects it has on the DOA estimation techniques that were discussed in Chapter 3. In this chapter we look at an ad-hoc pre-filtering technique called the Phase Transform (PHAT) and discuss how it can be used to improve the reliability of DOA estimation techniques.

4.1. The Generalized Cross-Correlation with Phase Transform

Consider the signals picked up by two microphones *i* and *j,* which belong to a microphone array. Let these signals be named $x_i(n)$ and $x_j(n)$. The cross-correlation between the two signals is defined as

$$R_{x_i x_j}(\tau) = E\left[x_i(n)\, x_j(n-\tau)\right] \tag{4.1}$$

The operator $E\left[f(n)\right]$ is called the expected value of $f(n)$ and for an observation window of N samples of $f(n)$, an estimate of the expected value can be written as

$$\hat{E}\left[f(n)\right] = \frac{1}{N}\sum_{i=1}^{N} f(i) \tag{4.2}$$

Section 2.5 talks about computing such a cross-correlation estimate between two signals, though there the computation was done in the frequency domain. Computation of the time delay now boils down to picking the delay that maximizes the cross-correlation function. Knapp and Carter introduced a more general version of (4.1) [7] called the Generalized Cross-Correlation (GCC). The GCC can be defined as

$$R_{x_i x_j}^{(g)}(\tau) = E\left[\left(h_i(n) * x_i(n)\right)\left(h_j(n-\tau) * x_j(n-\tau)\right)\right] \tag{4.3}$$

As stated in (4.3), to compute the GCC, the array signals are first pre-filtered and then the cross-correlation between the filtered signals is computed. The GCC can also be computed in

the frequency domain. The generalized cross power spectral density (GXPSD) [7] can be computed as

$$\Phi_{x_i x_j}^{(g)}(\omega) = \left[H_i(\omega) X_i(\omega)\right]\left[H_j(\omega) X_j(\omega)\right]^* \tag{4.4}$$

For the case of *M* discrete samples in the frequency domain, we can use a frequency index such that

$$\omega_k = \frac{2\pi k}{N} \tag{4.5}$$

Equation (4.4) can be re-written in terms of the discrete frequency index as

$$\Phi_{x_i x_j}^{(g)}(k) = \left[H_i(k) X_i(k)\right]\left[H_j(k) X_j(k)\right]^* \tag{4.6}$$

The two pre-filters can be combined and represented as a single filter, $\Psi_{ij}(k)$.

$$\Phi_{x_i x_j}^{(g)}(k) = \Psi_{ij}(k) X_i(k) X_j^*(k) \tag{4.7}$$

where

$$\Psi_{ij}(k) = H_i(k) H_j^*(k) \tag{4.8}$$

An approximation of the GCC can be computed as the inverse discrete Fourier transform (IDFT) of the discrete GXPSD given in (4.7).

$$R_{x_i x_j}^{(g)}(\tau) = \frac{1}{M}\sum_{k=0}^{M-1} \Phi_{x_i x_j}^{(g)}(k)\, e^{j\frac{2\pi k\tau}{M}} \tag{4.9}$$

The choice of the pre-filtering term $\Psi_{ij}(\omega)$ is dependent on the type of spectral weighting that is required by a situation. For example, when the aim of the pre-filters is to accentuate those frequencies that have high power and suppress those frequencies that have low power when compared to the noise power, $\Psi_{ij}(\omega)$ could be a function of the SNR spectra, which would either be known a priori or could be estimated from the given signal window.

Many different pre-filtering transforms have been studied [7]. Some examples are the Roth filter, the smoothed coherence transform (SCOT), the phase transform (PHAT), the Eckart filter, and the maximum likelihood filter. Of all these pre-filters, the PHAT offers the most interesting properties. Before we look in detail at PHAT let us take a closer look at the cross-correlation function.

Consider the signals arriving at two microphones with a delay of D samples between them

$$\begin{aligned} x_1(k) &= s(k) + n_1(k) \\ x_2(k) &= s(k+D) + n_2(k) \end{aligned} \tag{4.10}$$

The term $n_i(k)$ represents noise at the microphones. The Fourier transforms of these microphone signals can be expressed as

$$\begin{aligned} X_1(\omega) &= S(\omega) + N_1(\omega) \\ X_2(\omega) &= S(\omega)\, e^{j\omega D} + N_2(\omega) \end{aligned} \tag{4.11}$$

Thus the cross power spectral density with no pre-filtering is given by

$$\begin{aligned} \Phi_{x_1x_2}(\omega) &= X_1(\omega)\, X_2^*(\omega) \\ &= \left[S(\omega) + N_1(\omega) \right] \left[S(\omega) e^{j\omega D} + N_2(\omega) \right]^* \\ &= S(\omega) S^*(\omega) e^{-j\omega D} + S(\omega) N_2^*(\omega) + S^*(\omega) N_1(\omega) e^{-j\omega D} + N_1(\omega) N_2(\omega) \end{aligned} \tag{4.12}$$

The term $S(\omega)S^*(\omega)$ in (4.12) can be replaced by $\Phi_{ss}(\omega)$ which represents the power spectral density of the source signal. Thus we have

$$\Phi_{x_1x_2}(\omega) = \Phi_{ss} e^{-j\omega D} + S(\omega) N_2^*(\omega) + S^*(\omega) N_1(\omega) e^{-j\omega D} + N_1(\omega) N_2(\omega) \tag{4.13}$$

Since the noise signals are assumed to be uncorrelated with the signal and each other, the last three terms do not contribute towards the cross-correlation computation, which is an inverse Fourier transform operation.

$$\begin{aligned} R_{x_1x_2}(\tau) &= F^{-1}\left[\Phi_{ss}(\omega)e^{-j\omega D}\right] \\ &= R_{ss}(\tau) * \delta(\tau - D) \\ &= R_{ss}(\tau - D) \end{aligned} \tag{4.14}$$

Thus we find that the cross-correlation function is just the incident signal's auto-correlation centered at delay *D*. Ideally we would have liked the cross-correlation function to be a delta function at *D* so that the peak can be easily picked out. In reality, the auto-correlation of the incident signal ends up spreading the cross-correlation function around the delay *D*. When pre-filtering is used, as in GCC, the choice of $\Psi_g(\omega)$ must be geared towards minimizing this spread. If there happen to be multiple delays, as is the case in rooms with reverberation, the cross-correlation is given by

$$\begin{aligned} R_{x_1x_2}(\tau) &= R_{ss}(\tau) * \sum_i \alpha_i \delta(\tau - D_i) \\ &= \sum_i R_{ss}(\tau - D_i) \end{aligned} \tag{4.15}$$

This will result in peaks in the cross-correlation function at each delay. If the delays are very close to each other and if the spread in the signal auto-correlation is very large, then the peaks may merge thus decreasing the resolution of the estimate.

4.1.1. The Phase Transform

The pre-filter used in the phase transform (PHAT) [7] is

$$\Psi_g(\omega) = \frac{1}{\left|\hat{\Phi}_{x_1x_2}(\omega)\right|} \tag{4.16}$$

where $\hat{\Phi}_{x_1x_2}(\omega)$ is an estimate of the cross power spectral density of the two signals computed from estimates of the signal spectra $\hat{X}_1(\omega)$ and $\hat{X}_2(\omega)$.

Thus the estimated generalized cross-correlation with phase transform (GCC-PHAT) is given by

$$\hat{R}_{x_1x_2}^{(PHAT)}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\frac{1}{\left|\hat{\Phi}_{x_1x_2}(\omega)\right|}\right)\hat{\Phi}_{x_1x_2}(\omega)\,e^{j\omega\tau}d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\frac{1}{\left|\hat{\Phi}_{ss}(\omega)e^{-j\omega D}\right|}\right)\left(\hat{\Phi}_{ss}(\omega)e^{-j\omega D}\right)e^{j\omega\tau}d\omega$$

$$= \frac{1}{2\pi}\int_{\pi}^{\pi}\left(e^{-j\omega D}\right)e^{j\omega\tau}d\omega$$

The integral on the LHS reduces to a delta function at delay D. Thus we have

$$\hat{R}_{x_1x_2}^{(PHAT)}(\tau) = \delta(\tau - D) \tag{4.17}$$

where $\left|\hat{\Phi}_{ss}(\omega)e^{-j\omega D}\right| = \left|\hat{\Phi}_{ss}(\omega)\right| = \hat{\Phi}_{ss}(\omega)$ since the power spectral density of a signal is a real non-negative function of frequency. Thus, under the assumption of completely uncorrelated noise, the spread caused by the auto-correlation of the incident signal disappears and we can get a much sharper delta function for the cross-correlation. The phase transform is an ad-hoc technique to pre-whiten the signals before computing the cross-correlations in order to get a sharp peak. The time delay information is present in the phases of the various frequencies and these are not affected by the transform. Because the transform tends to enhance the true delay and suppress all spurious delays, it turns out to be very effective in rooms with moderate reverberation and one can easily pick the strongest peak as the true delay. One disadvantage of the phase transform is that it tends to enhance the effect of frequencies that have low power when compared to the noise power. This can cause the estimates to be corrupted by the effect of uncorrelated noise.

Figure 4.1 shows the cross-correlation between two speech signals, one of which was delayed from the other by 4 samples. Figure 4.1 (a) shows the regular cross-correlation and Figure 4.1 (b) shows the GCC-PHAT. Both cross-correlations show a peak at –4 samples, but notice that the GCC-PHAT has a much sharper and better-defined peak. The regular cross-correlation is an approximation of the auto-correlation of the source signal centered at a delay of $\tau = -4$ samples. On the other hand, the GCC-PHAT is an approximation of an impulse centered at a delay of $\tau = -4$ samples.

Now consider that in addition to the delay, the second signal also has an interfering component. This interfering component can be the result of a single reflection from the wall of a room. As an example for the interfering component, consider that the source signal is added again at a delay of 9 samples. Figure 4.2 shows the regular cross-correlation and the GCC-PHAT for this situation. These simulations were performed with a speech sample selected from a speech database from NIST [11].

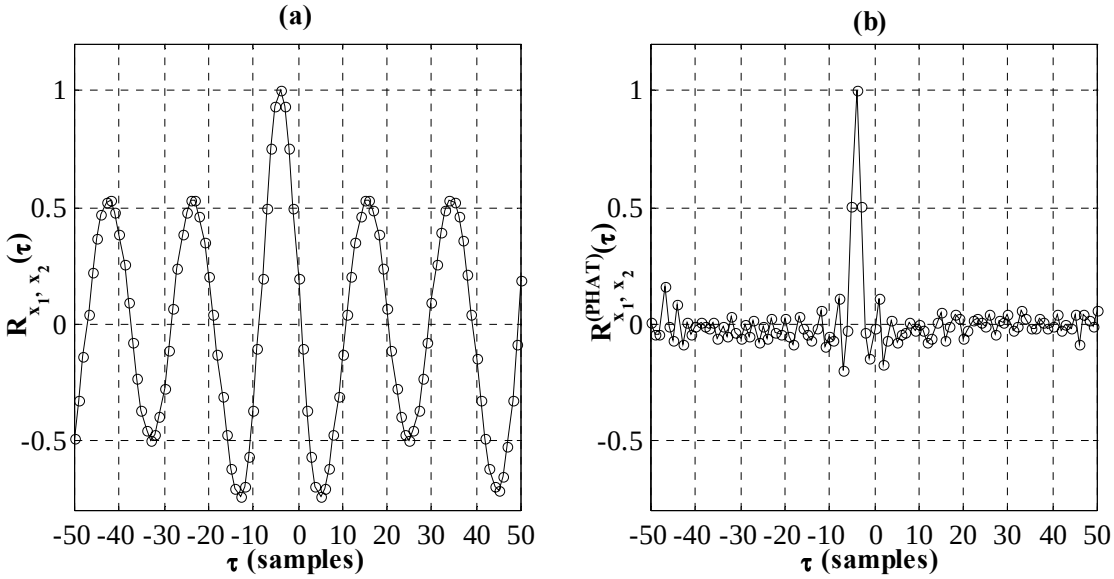

Figure 4.1 ***(a) Regular cross-correlation and (b) GCC-PHAT for two speech signals that have a delay of 4 samples between them.***

Notice, how in Figure 4.2 (a), the cross-correlation fails to separate the true delay from the reflection. Instead, the regular cross-correlation just gives a single peak at a delay of $\tau = -6$ samples. This delay is neither the true delay nor the delay at which the reflection occurs.

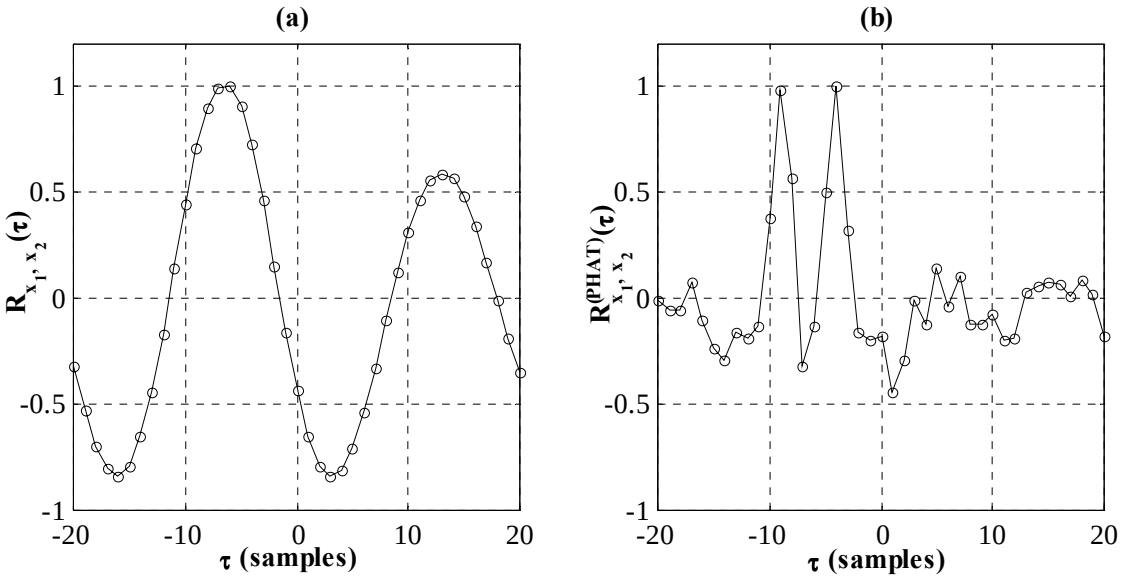

Figure 4.2 ***(a) Regular cross-correlation and (b) GCC-PHAT for two speech signals with a delay of 4 samples between them and one of the signals containing a reflection at 9 samples.***

On the other hand, the GCC-PHAT in Figure 4.2 (b) separates the two delays with two clearly separate peaks at the two correct delays. The GCC-PHAT is able to separate them because it has peaks that are not as spread as the ones for regular cross-correlation.

Now let us consider the case where the signals in both channels have one reflection each. Let the signal in channel 1 contain a reflection at a delay of 7 samples. Also let the signal in channel 2 be delayed by 4 samples and also contain a reflection at 15 samples. Figure 4.3 shows the regular cross-correlation and GCC-PHAT for this situation. Again notice that the cross-correlation fails completely to pick the relevant delays. In the GCC-PHAT, we notice 4 separate peaks. Each of these peaks corresponds to the delays between each pair of signals in the two channels. Hence there are peaks at –4 = – (4 – 0) samples, –15 = – (15 – 0) samples, 3 = – (4 – 7) samples and –8 = – (15 – 7) samples. In general, if there are n_i reflections in the i^{th} channel and n_j reflections in the j^{th} channel, then there should be $n_i n_j$ peaks in the GCC-PHAT between the two channels. Also notice that while the reflections in each channel were of the same strength as the signals themselves, the strengths of the four peaks are not the same. In fact the true delays of interest, at –4, is not the one with the maximum peak. Therefore, in the presence of multiple reflections there is a potential problem of which peak to pick for the true delay.

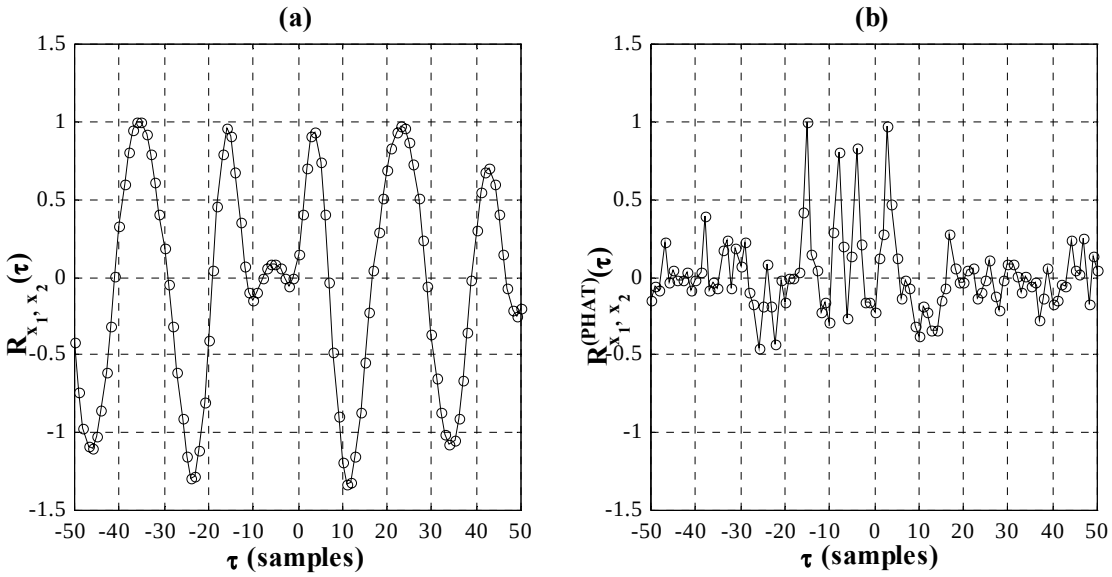

Figure 4.3 ***(a) Regular Cross-correlation and (b) GCC-PHAT with a single reflection in each channel.***

Of course in most real cases, the strengths of the reflections are typically weaker than those of the direct signals. Figure 4.4 shows the regular cross-correlation and GCC-PHAT for such a case. Here the reflection in channel 1 is of strength 0.7 times the strength of the signal and the reflection in channel 2 is of strength 0.8 times the strength of the signal. The delays at which the reflections occur are the same as for the case shown in Figure 4.3. The GCC-PHAT again shows 4 clear peaks at the correct delays. This time the peak corresponding to the delay between the direct signals is clearly stronger than those corresponding to the delay between various reflections and signals.

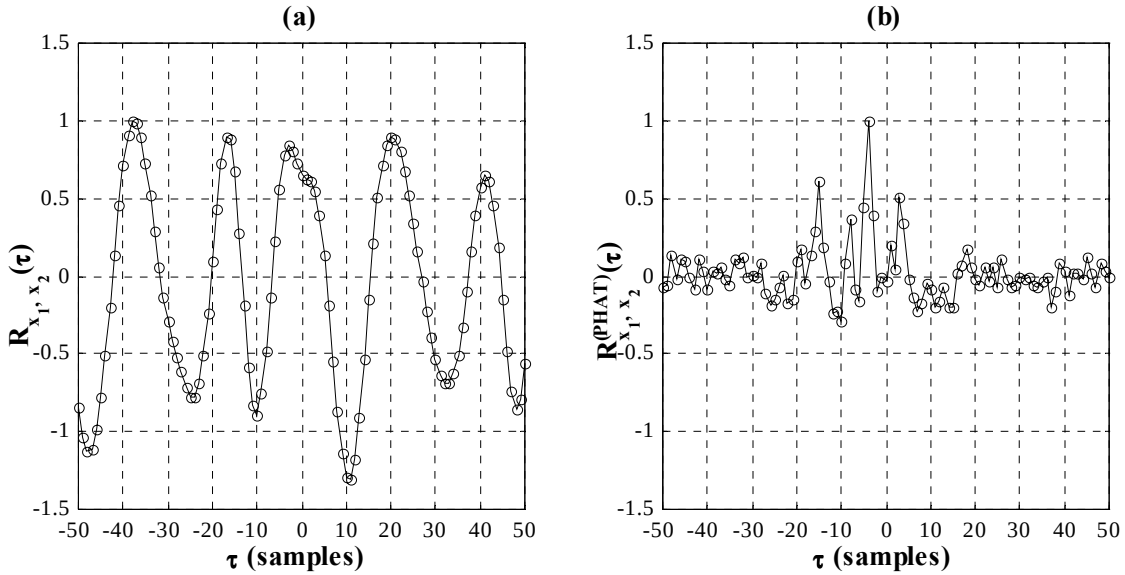

Figure 4.4 ***(a) Regular cross-correlation and (b) GCC-PHAT with strength of reflections lower than that of the signals.***

Figure 4.5 shows the frame-wise time-delay estimates obtained from a simulation. This simulation was for a speech signal that was sounded in a room with reverberation time of 100 *ms*. The microphone array consisted of 2 microphones separated by 10 *cm*. The true time delay for the simulation was 0.083 *ms*. The frames were of size 512 samples (32 *ms*). Figure 4.4 shows that the time-delay estimates from GCC-PHAT turn out to be more consistent than those from regular cross-correlation.

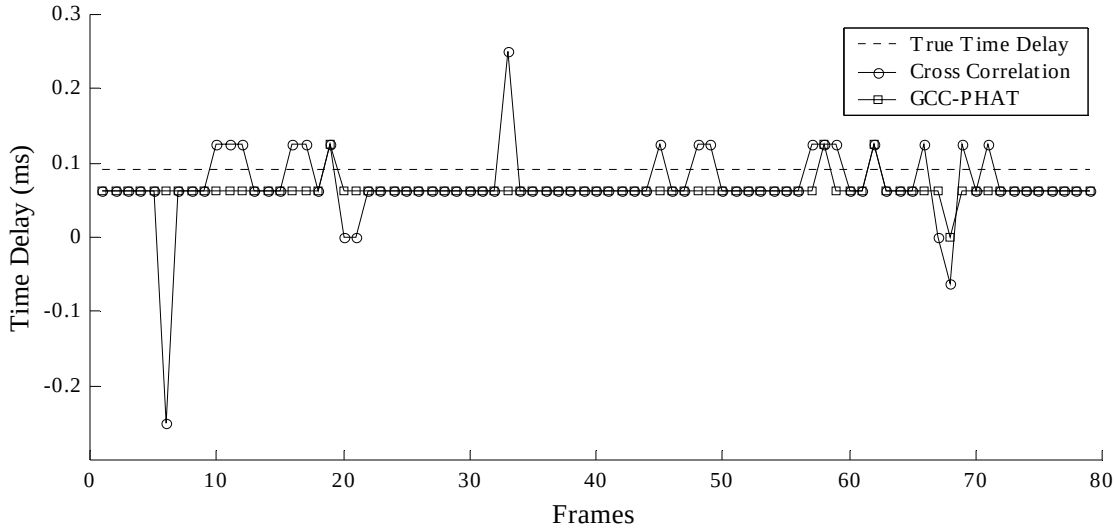

Figure 4.5 ***Frame-wise time-delay estimates showing improvement by using the phase transform.***

The better performance of PHAT-weighting is more evident in the reliability rate curves shown in Figure 4.6. The reliability rate has been plotted against percentage error in the time-delay estimates. The figure shows that PHAT-weighting improves the reliability of the time-delay estimates.

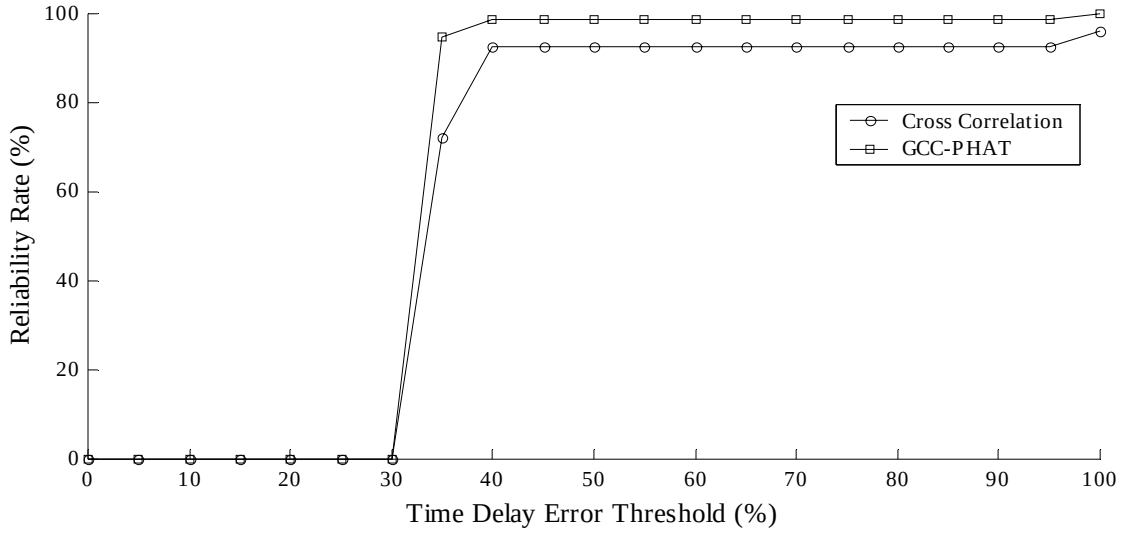

Figure 4.6 ***Reliability rate of time-delay estimates showing improvement by using the phase transform.***

4.2. Computation of Sub-sample Values of GCC-PHAT

In the discussions in Section 4.1 we have assumed that the delay between signals is an integer number of samples. This is not always the case. For small arrays with low sampling frequencies, it is conceivable that most delays between signals will be in between sample delays. For example, a ULA with a separation of 10 *cm* will have a maximum delay between microphones of approximately 0.29 *ms*. At a sampling frequency of 8 *kHz* this amounts to only about 2.32 samples of delay. Hence all angles between 0° and 90° would be mapped to sample delays between 0 and 2. The error in angle encountered when the delay of 2.32 samples is rounded to 2 samples would be as big as 30.4°. Hence we see the need to interpolate the GCC-PHAT for more accurate estimation of the DOA.

The interpolation of GCC-PHAT can be performed by frequency domain zero-padding. As an example consider a sinusoidal signal of discrete frequency 0.25 cycles per sample as shown in Figure 4.7 (a). A frame of 256 samples of this signal was generated using the

computer. Figure 4.7 (b) shows the magnitude of the DFT samples of this signal. A 256-point DFT was computed to generate this figure.

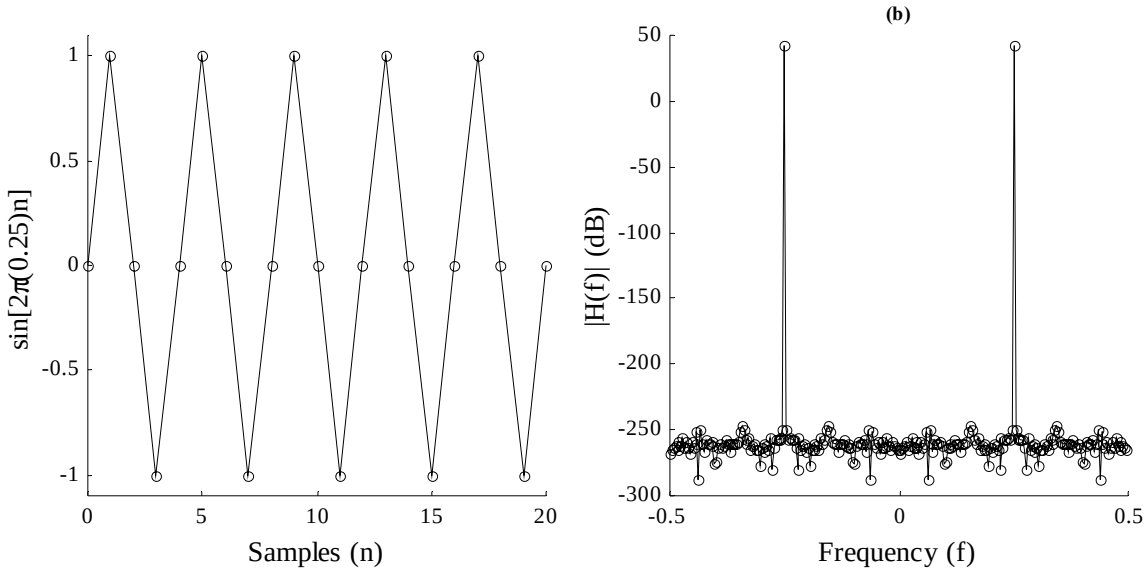

Figure 4.7 ***(a) Sinusoid of discrete frequency 0.25 cycles per sample and (b) magnitude of its DFT.***

Figure 4.8 (b) shows a zero-padded version of the DFT of the signal. The zero-padding is done in the middle of the DFT and the number of zeros padded is such that the length of the DFT is doubled. Figure 4.8 (a) shows the IDFT of this zero-padded DFT. Notice that zero-padding in the frequency domain has resulted in interpolation in the time domain. In effect, by zero-padding the DFT, we have decreased the discrete frequency by a factor of 2, thus increasing the sampling frequency by a factor of 2, which results in interpolation. Figure 4.9 shows the error in interpolation. The interpolated sinusoid was subtracted from a sinusoid that was generated at double the sampling frequency to generate the plot in Figure 4.9. The energy in the error signal in Figure 4.9 was found to be 4.1×10^{-14}.

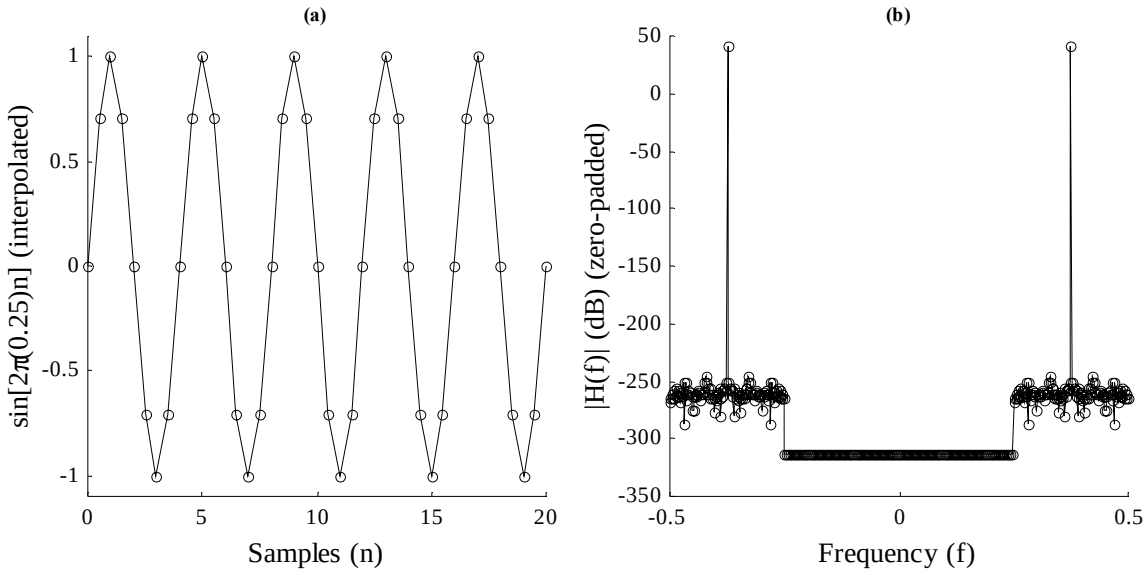

Figure 4.8 ***(a) Interpolated Sinusoid at 0.25 cycles per sample and (b) magnitude of zero-padded DFT.***

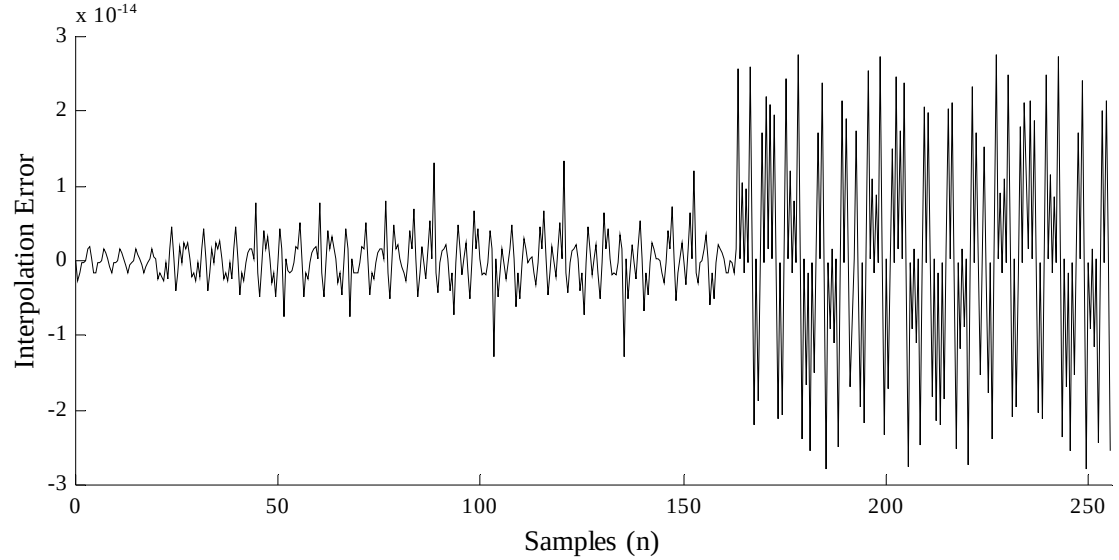

Figure 4.9 ***Error in Interpolation for a sinusoid at 0.25 cycles per sample.***

The interpolation-error that was computed in the preceding example was for a sinusoid that had exactly 4 samples in each time period. Also these 4 samples were of exactly the same value in each time period. The length of the DFT that was computed for this signal was of the same length as that of the signal and this resulted in one of the DFT samples being at exactly the frequency of the sinusoid. This DFT sample has a high magnitude and all other DFT samples have very low magnitudes. Therefore the DFT was a very accurate representation of the sinusoid and the effects of the finite frame size of the time-domain signal are not evident. This resulted in

the interpolated result also being very accurate. The same is not true if we compute the DFT at a finer level, say, at twice the frequency resolution by zero-padding the time-domain signal to make it twice the length. This is usually done while computing cross-correlations from the XPSD samples because such cross-correlations are in fact circular convolutions. Therefore it is prudent to zero-pad the signals at the end to increase their lengths before performing the cross-correlation to counter the edge effects of circular convolution.

The magnitudes of the DFT samples obtained by zero-padding the time-domain signal are shown in Figure 4.10. The effect of the finite length of the signal, which was not visible in Figure 4.7 (b) are now visible in Figure 4.10. We no longer have only one DFT sample having a large magnitude and all the other samples having small magnitudes.

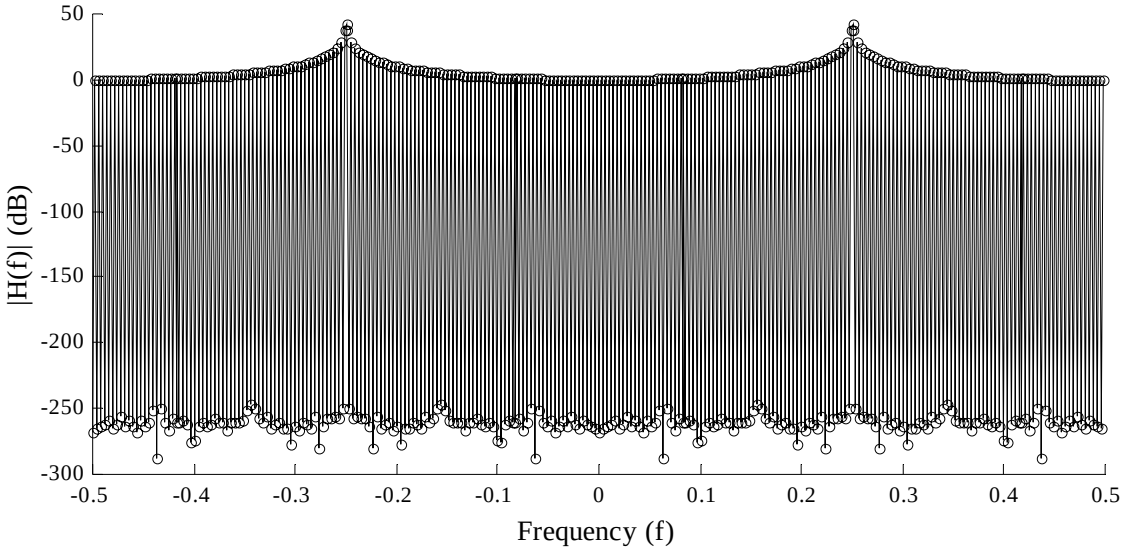

Figure 4.10 ***Magnitude of the DFT samples of a sinusoid at f = 0.25 cycles per sample computed with 256 samples of signal and 256 samples of zero padded at the end.***

The signal was interpolated using these DFT samples and the error in interpolation for this case is shown in Figure 4.11. The energy in this error signal was found to be 0.0484. We observe that because we have a less accurate frequency domain representation of the signal, we have less accurate interpolation.

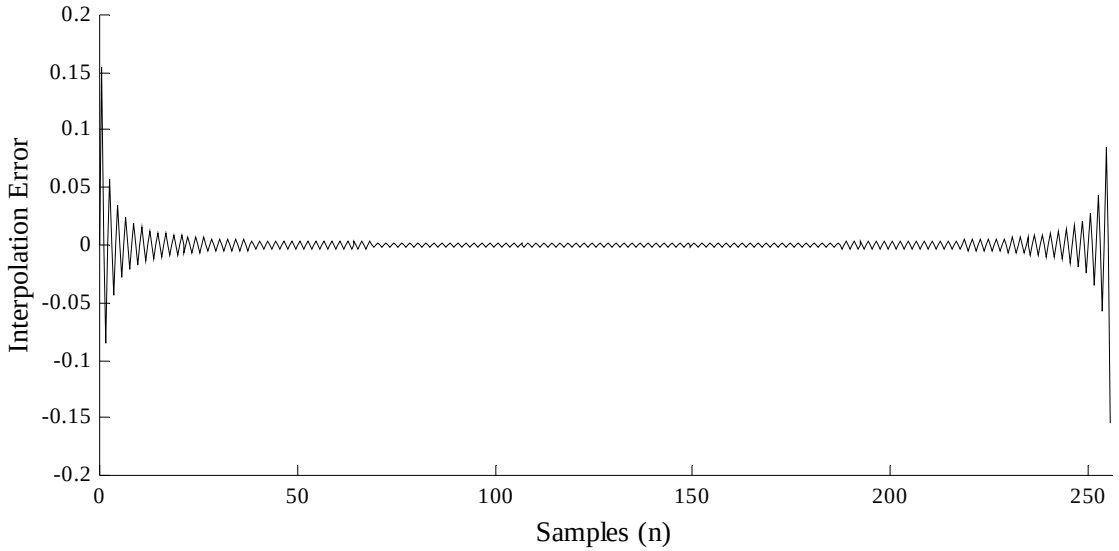

Figure 4.11 ***Interpolation error for a sinusoid of frequency 0.25 cycles per sample when length of the DFT was twice the length of the signal frame.***

The same effect is observed when the frequency of the sinusoid does not exactly fall on one of the DFT samples. For example, the interpolation was performed for a sinusoid of frequency 0.2512 cycles per sample. Figure 4.12 shows the magnitude of the DFT samples and interpolation error for the case when the length of the DFT was equal to the length of the signal. Figure 4.13 shows the same quantities for the case when the length of the DFT was twice that of the signal. We observe that in both cases here the errors in interpolation are much larger than those observed in Figure 4.9. The energy in the error signal for the first case was 0.8245 and the same in the second case was 0.4272. In this case we observe that there is less error when we compute the DFT at twice the length of the signal.

Therefore we need to study the energies in the error signal for the range of frequencies of interest and determine whether we need to compute the DFTs at twice the signal length or not. Assuming a sampling frequency of 8 *kHz* and assuming that most of the energy in human speech will be at frequencies less than 1.5 *kHz*, we have a range of fractional frequencies from 0 to 0.1875 cycles per sample. Signal frames of 256 samples were generated for 1000 frequencies distributed evenly in this range and the energies in the error signals were computed for each of these frequencies for both cases, one where the DFT length was equal to the signal length, and the other where the DFT length was twice that of the signal.

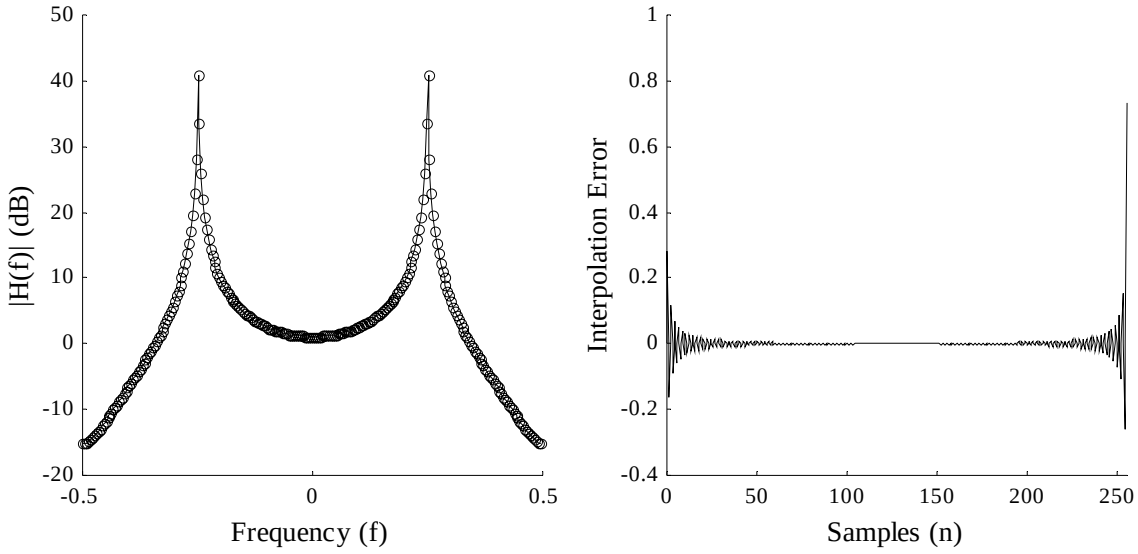

Figure 4.12 ***Magnitude of DFT samples and interpolation error for a sinusoid at 0.2512 cycles per sample with DFT length equal to signal length.***

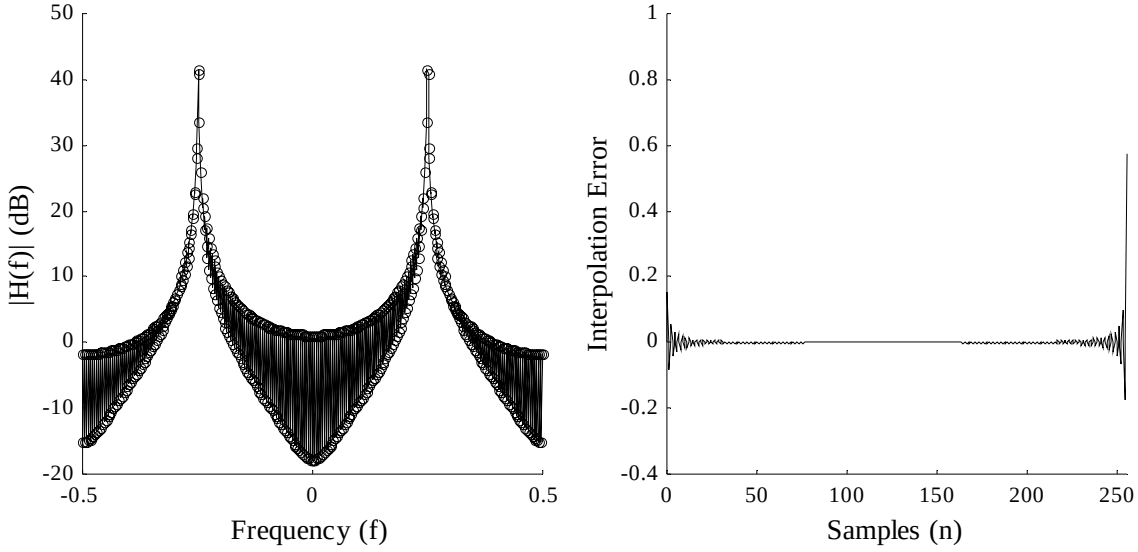

Figure 4.13 ***Magnitude of DFT samples and interpolation error for a sinusoid at 0.2512 cycles per sample when DFT length is twice that of signal length.***

The energies obtained for different frequencies for the two cases are shown in Figure 4.14. It is interesting to note the periodic behavior in the energy. As the frequency gets closer to one of the DFT samples, the energy keeps decreasing and reaches a minimum. Similarly as the frequency gets farther away from one of the DFT samples, the energy keeps increasing and reaches a maximum.

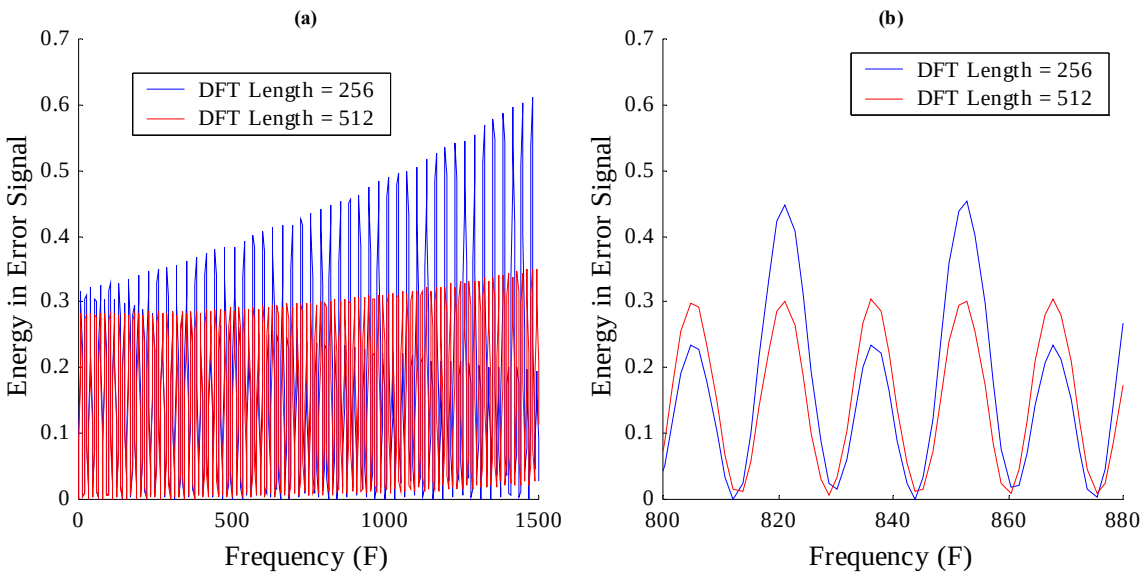

Figure 4.14 ***Energies in interpolation-error signals against frequency of sinusoid for both cases, one where DFT length is equal to signal length and the other where DFT length is twice the signal length.***

We also observe that if at one maxima, the case where the DFT length is equal to the signal length has more energy, at the next maxima, the other case has more energy. The energies where summed across all frequencies and it was observed that the case where the DFT length was equal to the signal length had a total energy of 181.2 and the other case had a total energy of 156.7. Thus we observe that – on average – computing the DFT at twice the frequency resolution gives us better results.

In practice, the process of zero-padding the DFT in the frequency domain and computing the IDFT to get the interpolated signal can be performed using the Goertzel algorithm [19]. Previously a real linear filter was used to compute the DFT of a real signal at the required frequency sample. Here a complex linear filter is used to compute the IDFT of a complex signal at the required time-sample. With a proper rotational factor the Goertzel algorithm can be used to compute the IDFT at any arbitrary time between samples.

Now we can redo the simulation that was performed in Section 4.1.1 with interpolation of the GCC-PHAT using the Goertzel algorithm. The results for this new simulation are shown in Figure 4.15 and Figure 4.16. For comparison the results for the non-interpolated case have also been plotted. The interpolation was done in such a way such that the difference between adjacent time delays at which the cross-correlations and GCC-PHAT were computed was equivalent to 1° separation in bearing angle for each pair of microphones. The reliability rate

curve of the interpolated case shows a definite improvement over the non-interpolated case in the low error threshold region between 0 % to 20 % error. The sudden jump in the non-interpolated curves at 25 % causes them to perform better than the non-interpolated case in the 25 % to 50 % error regions. The reason for this jump is that the time-delay estimates appear to be biased estimates and the non-interpolated estimates hold a steady biased value whereas the interpolated estimates take values on either side of the bias.

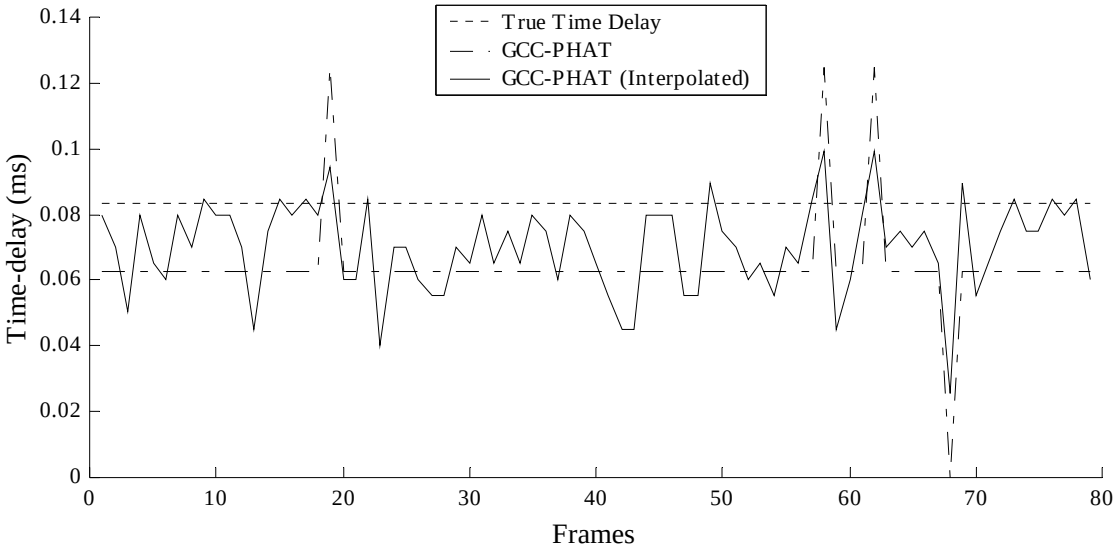

Figure 4.15 ***Frame-wise time –delay estimates with and without interpolation.***

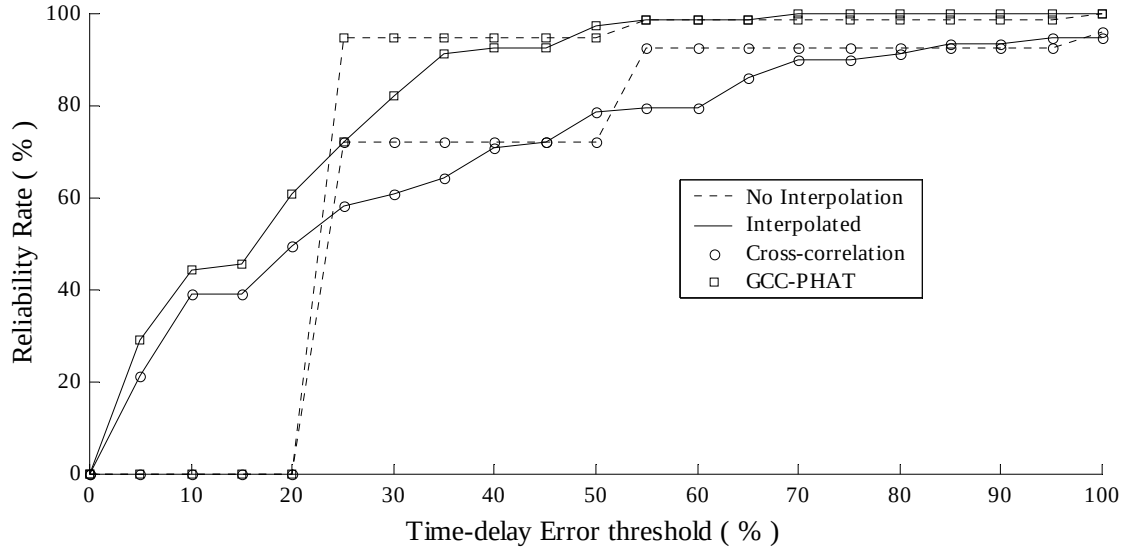

Figure 4.16 ***Reliability rate of time-delay estimate with and without interpolation.***

We also need to study whether computing the DFT at twice the frequency resolution does indeed give us better results in time-delay estimation. Figure 4.17 shows the reliability-rate curves for such a simulation. The simulation was performed using a randomly chosen speech signal as the source impinging on the array from different angles, ranging from –90° to +90°, with a separation of 10°. The array used was linear and consisted of 4 elements separated by a distance of 10 *cm*. The array was placed in the center of a room of dimensions $5\ m \times 5\ m \times 5\ m$ having a reverberation time of 100 *ms.* A signal to noise ratio of 30 *dB* was used for this simulation. DOA estimation was performed using GCC-PHAT. In one case the DFT used was of the same length as the length of the signal. In the second case the DFT length was twice that of the signal length. Figure 4.17 shows that we get a slight improvement in performance by making the DFT lengths twice that of the signal length. This result corroborates our earlier result where we got less energy in the interpolation signal-error when the DFT length was twice the signal length.

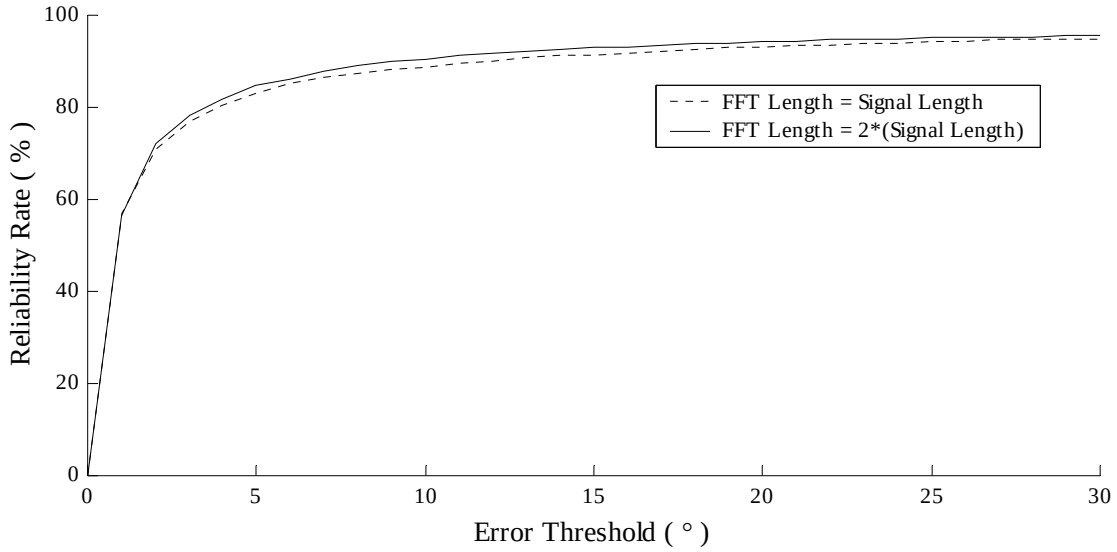

Figure 4.17 ***Reliability-rate with and without time-domain zero-padding.***

4.3. Formulation for Three Dimensional Array

In this section we develop the least squares solution using the GCC-PHAT for an arbitrary 3-dimensional (3D) array. Consider an arbitrary 3-dimensional microphone array consisting of N microphones. The location of the array can be represented using a $3 \times N$ matrix, $\mathbf{M}$, each column of which represents the position of a microphone in 3-dimensional Euclidean

space. The first column in this matrix represents the reference microphone and thus can be considered to be the origin of the 3D space. The DOA for the 3D case has two components, azimuth (θ) and elevation (ϕ). Figure 4.18 shows the reference microphone with a signal impinging on it from an arbitrary direction in 3D space. This DOA can be represented as a unit vector, $\mathbf{s}$, pointing towards the source. Note that the direction of $\mathbf{s}$ is the negative of the actual DOA. We define the elevation, ϕ, of the DOA as the angle between $\mathbf{s}$ and the *x*-*y* plane, measured from the *x*-*y* plane towards the positive *z*-axis. Next we define the azimuth, θ, of the DOA as the angle between the *y*-axis and the projection of $\mathbf{s}$ onto the *x*-*y* plane, measured from the positive *y*-axis towards the positive *x*-axis (clockwise). The azimuth can take on values between 0° and 360° and the elevation can take on values between – 90° and 90°.

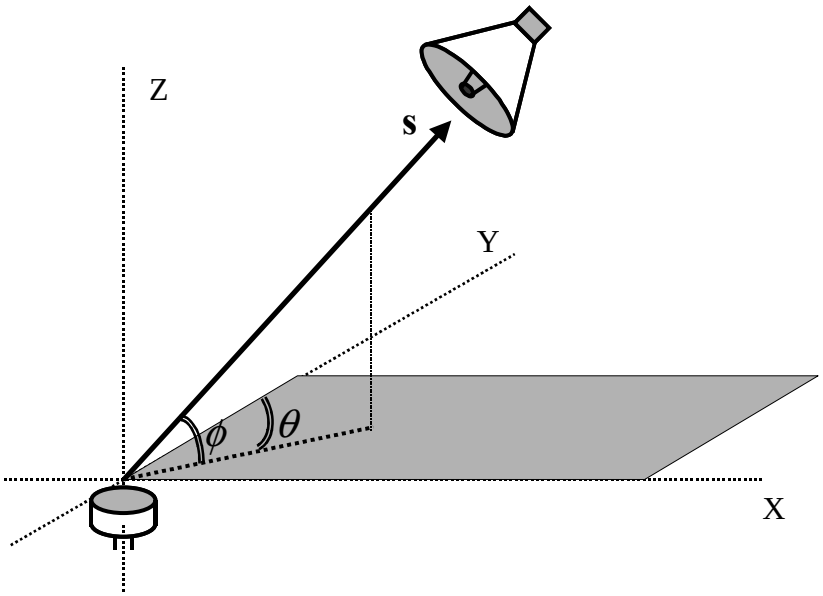

Figure 4.18 ***Microphone in 3D space showing azimuth and elevation of the DOA.***

The unit vector can be expressed in terms of azimuth and elevation. The length of the projection of $\mathbf{s}$ onto the *x*-*y* plane is $\cos\phi$. When this length is projected on the *x*-axis, it becomes $\cos\phi \cdot \sin\theta$. Similarly when it is projected on the *y*-axis it becomes $\cos\phi \cdot \cos\theta$. Also the projection of $\mathbf{s}$ on the *z*-axis is $\sin\phi$. Thus the unit vector, $\mathbf{s}$, can be written, in rectangular coordinates, as $\mathbf{s} = [\cos\phi \cdot \sin\theta \quad \cos\phi \cdot \cos\theta \quad \sin\phi]^T$. Consequently the unit vector in the direction of the DOA can be written as $-[\cos\phi \cdot \sin\theta \quad \cos\phi \cdot \cos\theta \quad \sin\phi]^T$.

Next we derive the relationship between the DOA and the time delay between any pair of microphones in 3D space. We derive it first in 2-dimensional (2D) space and then extrapolate it to 3D space. Consider the pair of microphones as shown in Figure 4.19 for 2D space. The position of microphone m_1 can be represented with the column vector $\mathbf{m}_1$ and the position of microphone m_2 can be represented with the column vector $\mathbf{m}_2$. The vector joining the two microphones can be written as $\mathbf{m}_2 - \mathbf{m}_1$. The unit vector $\mathbf{d}$ represents the DOA. The extra distance that the signal has to travel to reach m_2 over that to reach m_1 is called the range difference and is represented by d_{21}.

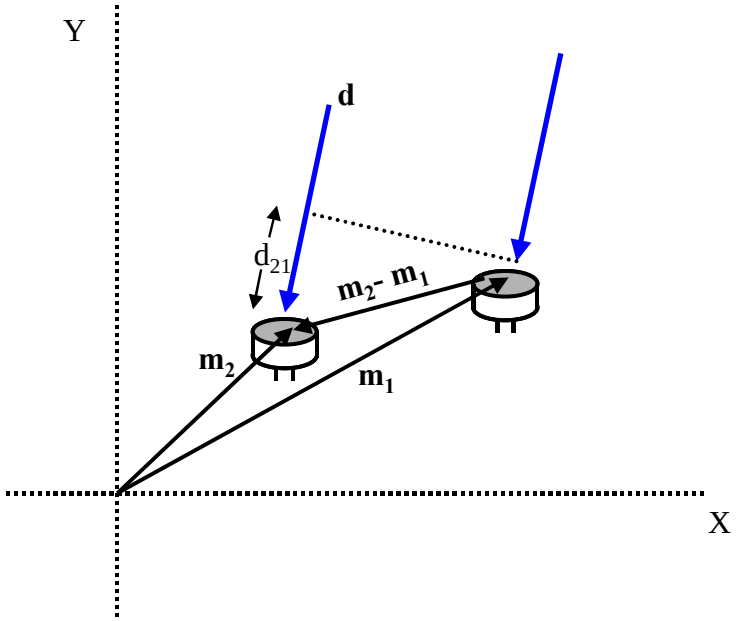

Figure 4.19 ***Range difference as a projection of the vector joining two microphones on the DOA.***

It is this range difference that results in a time delay between signals arriving at m_1 and m_2. As can be seen from Figure 4.19, the range difference is really the projection of the vector joining the two microphones onto the DOA. In other words the range difference is the inner product of $(\mathbf{m}_2 - \mathbf{m}_1)$ and $\mathbf{d}$. Thus we have

$$d_{21} = (\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{d} \tag{4.18}$$

Without loss of generality, (4.18) is valid for the 3D case as well. The time delay of arrival between the signals at m_2 and m_1 is thus given by

$$\tau_{21} = \left(\frac{1}{v}\right)\left[(\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{d}\right] \tag{4.19}$$

where v is the velocity of sound.

We can construct a matrix, $\mathbf{A}$, each column of which is a vector joining a pair of microphones and will thus be of the form $\mathbf{m}_i - \mathbf{m}_j$. Thus, if we have N microphones in the array, $\mathbf{A}$ will be a matrix of size $3 \times M$ where $M = \binom{N}{2}$. Here M is the number of microphone pairs in the array. From this we can construct a vector of M time delays for each pair of microphones in the array.

$$\boldsymbol{\tau} = \left(\frac{1}{v}\right)\mathbf{A}^T \mathbf{d} \tag{4.20}$$

For a fixed microphone array, the time delays are a function of only the DOA, or, in other words, the time delays are a function of azimuth and elevation. Thus (4.20) can be re-written as

$$v\,\boldsymbol{\tau}(\theta,\phi) = \mathbf{A}^T \mathbf{d}(\theta,\phi) \tag{4.21}$$

where $\mathbf{d}(\theta,\phi) = -[\cos\phi \cdot \sin\theta \quad \cos\phi \cdot \cos\theta \quad \sin\phi]^T$ and v has been taken to the other side of the equation. Equation (4.21) can be solved for $\mathbf{d}(\theta,\phi)$ in the least squares sense [13] as

$$\hat{\mathbf{d}}_{LS}(\theta,\phi) = \left[\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\right] v\,\hat{\boldsymbol{\tau}}(\theta,\phi) \tag{4.22}$$

where $\hat{\mathbf{d}}_{LS}(\theta,\phi)$ is the least squares (LS) estimate of the unit vector and $\hat{\boldsymbol{\tau}}(\theta,\phi)$ is the vector of estimated pair-wise time-delays.

4.4. Steered Response Power with Phase Transform (SRP-PHAT)

In this section we look at a different method [1] to estimate the DOA of acoustic signals in a reverberant environment. This method is very similar to the beamformer-based methods where we compute the power of the array output signal by forming beams in each of the possible DOAs. The direction that gives the highest power is assumed to be the estimated DOA. In the

SRP-PHAT algorithm the metric that is computed for each direction is the cumulative GCC-PHAT value across all pairs of microphones at the theoretical time-delays associated with the chosen direction. Consider the microphones m_i and m_j. The time-delay between the two microphones for a signal coming from an azimuth of θ and an elevation of ϕ can be labeled as $\tau_{ij}(\theta,\phi)$. Then the estimated GCC-PHAT value for these two microphone-signals at that delay can be written as the inverse Fourier transform of the generalized cross power spectral density.

$$\hat{R}_{x_i x_j}^{PHAT}\left(\tau_{ij}(\theta,\phi)\right) = \int_{-\pi}^{\pi} \Psi_{i,j}(\omega)\hat{X}_i(\omega)\hat{X}_j^*(\omega)e^{j\omega\left(\tau_{ij}(\theta,\phi)\right)}d\omega \tag{4.23}$$

The SRP-PHAT metric, $P(\theta,\phi)$ can be written as the sum of $\hat{R}_{x_i x_j}^{PHAT}\left(\tau_{ij}(\theta,\phi)\right)$ over all microphone pairs.

$$\begin{aligned} P(\theta,\phi) &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \hat{R}_{x_i x_j}^{PHAT}\left(\tau_{ij}(\theta,\phi)\right) \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\int_{-\pi}^{\pi} \Psi_{i,j}(\omega)\hat{X}_i(\omega)\hat{X}_j^*(\omega)e^{j\omega\left(\tau_{ij}(\theta,\phi)\right)}d\omega \\ &= \int_{-\pi}^{\pi}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \Psi_{i,j}(\omega)\hat{X}_i(\omega)\hat{X}_j^*(\omega)e^{j\omega\left(\tau_{ij}(\theta,\phi)\right)}d\omega \end{aligned} \tag{4.24}$$

where N is the number of microphones in the array. Note that this metric contains, among the cross-correlations, N autocorrelations at delay 0. These correspond to the cases in the summation where i is equal to j. These autocorrelations are constant across all DOAs and thus only add a bias to the SRP-PHAT metric. $P(\theta,\phi)$ can be computed for all possible values of θ and ϕ. In order to compute $P(\theta,\phi)$ one needs to first compute the theoretical time delay between all pairs of microphones for the given DOA. The direction that maximizes $P(\theta,\phi)$ is the estimated DOA.

Simulations were performed using a 4-element linear array with a separation of 10 *cm* placed in a room of dimensions 5 *m* × 5 *m* × 5 *m* and with a reverberation time of 100 *ms*. Random speech signals were played from 19 different directions ranging from –90° to +90° separated by 10°. Frames of size 512 samples (32 *ms*) were used to perform DOA estimation

using both methods, namely, least squares solution from GCC-PHAT based TDE estimates and the SRP-PHAT algorithm. Figure 4.20 shows a sample SRP-PHAT computed from one of the frames of the signal incident from an angle of 30°. It displays a strong sharp peak at the correct DOA.

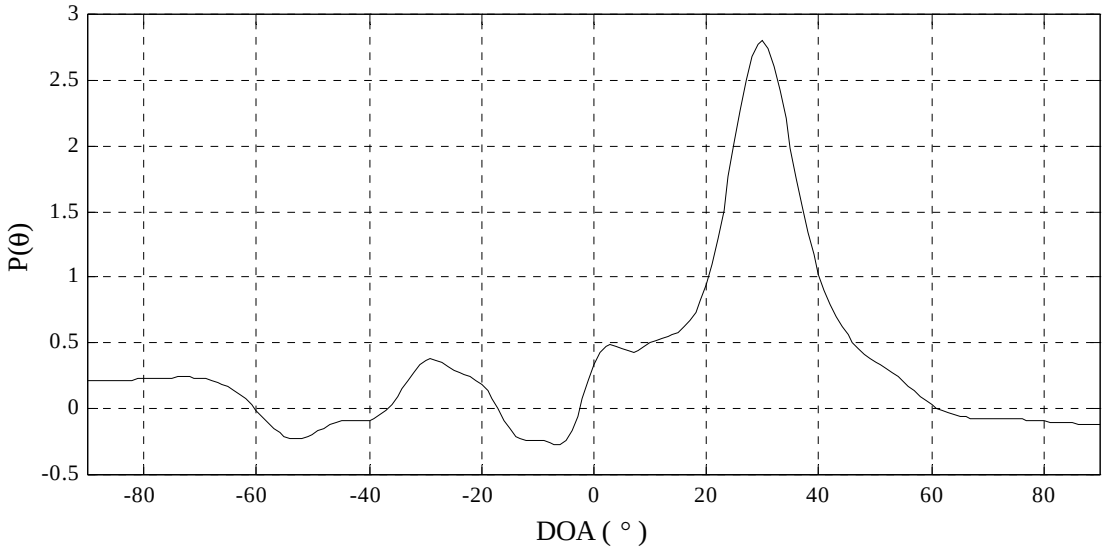

Figure 4.20 ***Sample SRP-PHAT for a true DOA of 30°.***

Figure 4.21 shows the reliability rate curves for the two methods. The SRP-PHAT method shows a distinct performance benefit over the GCC-PHAT based method. This result corroborates the claim made in the literature [3].

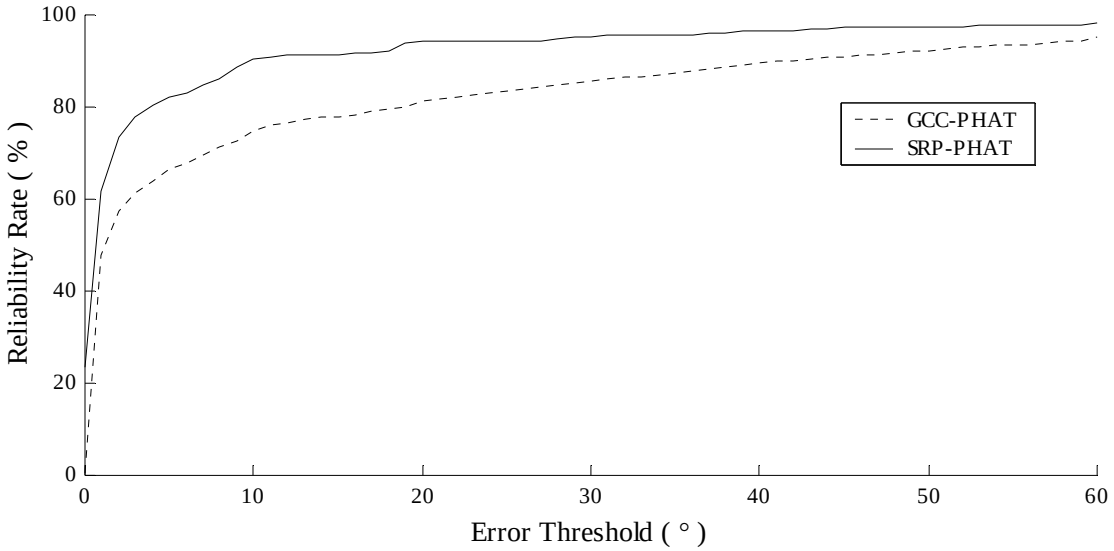

Figure 4.21 ***Reliability-rates for GCC-PHAT and SRP-PHAT methods.***

When we compare the computational requirements for the two methods we can see a distinct advantage in using the time-delay estimate based method. First let us assume a linear array with 4 microphones. This gives us a total of 6 pairs of microphones in the array. Assuming that we want a resolution of 1°, this gives us 181 directions ranging from -90° to +90°. We are required to compute 6 different GCC-PHAT values (one for each pair) for each of the directions giving a total of $6\times181=1086$ computations of GCC-PHAT. Again, assume that we need to estimate time delays at a resolution of 1°. This means that we need to compute 181 GCC-PHAT values for each microphone pair, from which we pick the delay that maximizes the GCC-PHAT. Again we have 6 pairs, giving a total of $6\times181=1086$ computations of GCC-PHAT. Thus we see no apparent computational disadvantage in the SRP-PHAT method for a linear array. However if we use a search based method [8] to reach the peak of the GCC-PHAT for each pair of microphones, then we need to compute far fewer GCC-PHAT evaluations than 1086.

When we consider a 3D array, we can see a distinct computational advantage in the time-delay based method. In this case we have 360 different azimuth angles between 0° and 359° and 181 different elevation angles between -90° and +90°. At 1° resolution, and for 6 pairs of microphones, thus requires $359\times181\times6=389874$ GCC-PHAT evaluations. Even if we perform a two-step search where we first compute the SRP-PHAT at 5° resolution and then refine the

search to 1° resolution, we still have 16710 evaluations of GCC-PHAT. Compared to this in the time-delay based method we still only have to compute 1086 values of GCC-PHAT.

Thus we see that in spite of the superior performance of the SRP-PHAT method, the TDE based methods are still an attractive option in terms of computational requirements. In this thesis we try to come up with methods that improve the performance of the TDE based methods to make it comparable to the performance of SRP-PHAT.

4.5. Implementation of the Phase Transform

Equation (4.16) describes the computation of the PHAT weighted GXPSD samples as a division by the magnitude of the XPSD sample. This involved two steps. The first step is the computation of the magnitude of the XPSD sample, which involves the computation of a square root because the XPSD samples are complex numbers. The second step is division of a complex number by a real number, which, in essence involves two real divisions. Both of these operations viz., computations of square roots and divisions are expensive operations in a digital signal processor (DSP). This is because DSPs are tuned towards optimizing multiply-accumulate and shifting operations. Therefore operations such as computation of square roots and divisions have to be manually programmed using standard algorithms available in the computer engineering literature. These algorithms add to the cost of the DOA estimation task because they take up multiple clock cycles for each square-rooting or division operation. Therefore it is important to study alternate methods to easily implement the phase transform.

Consider a complex number, c, which can be written as

$$c = a + jb \tag{4.25}$$

This can also be expressed as

$$c = Ae^{j\phi} \tag{4.26}$$

where

$$A = \sqrt{a^2 + b^2} \tag{4.27}$$

and

$$\phi = \tan^{-1}\left(\frac{b}{a}\right) \tag{4.28}$$

The computation of each PHAT weighted GXPSD sample involves the operation

$$c' = \frac{c}{|c|} \tag{4.29}$$

The real part of c' can be expressed as

$$\begin{aligned} \mathrm{Re}\{c'\} &= \frac{a}{\sqrt{a^2 + b^2}} \\ &= \frac{A\cos\phi}{\sqrt{A^2\cos^2\phi + A^2\sin^2\phi}} \\ &= \frac{A\cos\phi}{A\sqrt{\cos^2\phi + \sin^2\phi}} \\ &= \cos\phi \end{aligned} \tag{4.30}$$

The imaginary part of c' can be computed in a similar manner. Thus we have

$$c' = \cos\phi + j\sin\phi \tag{4.31}$$

Therefore we observe that the phase transform merely extracts the cosine and sine values of the phase of the XPSD sample. We can use the computationally efficient Co-Ordinate Rotation Digital Computer (CORDIC) algorithm to compute the cosine and sine values of the phase of a complex number. We use a two step procedure to compute the PHAT weighted GXPSD. First we use the CORDIC algorithm to compute the phase of the XPSD sample. Next we use the CORDIC algorithm to compute the cosine and sine values of the phase.

4.5.1. CORDIC-Based Computation of the Phase

The CORDIC algorithm was originally introduced as a simple fixed-point technique to efficiently compute trigonometric values. Using simple operations such as shifts and additions and using a look-up table, the CORDIC algorithm computes trigonometric values that increase in precision at the rate of one bit per iteration. The procedure to compute the phase of a complex number is as follows.

The complex number c given in (4.25) can be expressed as a two dimensional vector of the form

$$\mathbf{c} = \begin{bmatrix} a & b \end{bmatrix} \tag{4.32}$$

In order to rotate this vector by an angle θ_1, as shown in Figure 4.22, we can multiply the vector with a matrix as follows:

$$\mathbf{c}' = \begin{bmatrix} a' & b' \end{bmatrix} = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{4.33}$$

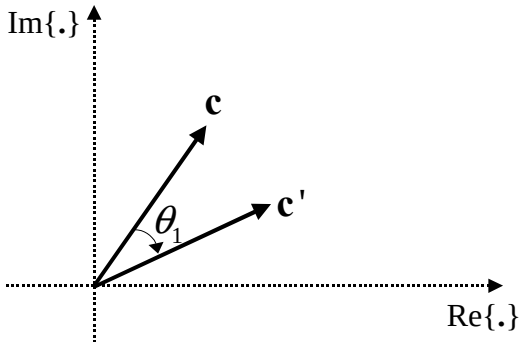

Figure 4.22 ***A complex number represented as a two dimensional vector and another complex number generated by rotating it.***

The matrix that performs the rotation is called a co-ordinate rotation matrix. If we need to rotate $\mathbf{c}'$ further by an angle θ_2 then we need to multiply it again by a matrix of the form $\begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{bmatrix}$. Suppose now we have a set of N angles, θ_i, such that

$$\theta = \sum_{i=1}^{N} \theta_i \tag{4.34}$$

The sequence of rotations of $\mathbf{c}$ by these N angles can be computed as a sequence of multiplications of $\mathbf{c}$ by co-ordinate rotation matrices as shown below.

$$\mathbf{c}' = \begin{bmatrix} a & b \end{bmatrix} \left\{ \prod_{i=1}^{N} \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \right\} \tag{4.35}$$

The scalar, $\cos\theta_i$, can be factored out of the matrix to give

$$\begin{aligned}\mathbf{c}' &= \begin{bmatrix} a & b \end{bmatrix} \left\{ \prod_{i=1}^{N} \cos\theta_i \begin{bmatrix} 1 & \tan\theta_i \\ -\tan\theta_i & 1 \end{bmatrix} \right\} \\ &= \left\{ \prod_{j=1}^{N} \cos\theta_j \right\} \begin{bmatrix} a & b \end{bmatrix} \left\{ \prod_{i=1}^{N} \begin{bmatrix} 1 & \tan\theta_i \\ -\tan\theta_i & 1 \end{bmatrix} \right\} \\ &= K \begin{bmatrix} a & b \end{bmatrix} \left\{ \prod_{i=1}^{N} \begin{bmatrix} 1 & \tan\theta_i \\ -\tan\theta_i & 1 \end{bmatrix} \right\}\end{aligned} \tag{4.36}$$

where

$$K = \prod_{j=1}^{N} \cos\theta_j \tag{4.37}$$

For a given set of angles, the value of K can be computed in advance. We now constrain the angles such that

$$\tan\theta_i = (\pm)_i \, 2^{-i} \tag{4.38}$$

The constraint on the angles given in (4.38) means that the only operations involved in the matrix multiplication, and consequently in the co-ordinate rotations, are shifts and adds. This is a very desirable feature for a fixed-point implementation. Notice that the sign of the tangent in (4.38) is dependent on the iteration. This means that at each iteration, the sign of the tangent, and thus the direction of rotation, is chosen such that the new complex number gets closer to the x-axis. This can be detected by choosing a sign that decreases the absolute value of b', the imaginary part of the rotated complex number. Also, the fact that at each iteration the value of the tangent decreases by a factor of two means that the angle of rotation at each iteration becomes smaller and smaller. Thus we can choose a set of N rotations of increasing precision that ends up aligning the rotated complex number along the x-axis. The sum of all these rotations is the total angle by which the complex number was rotated in order to make it align with the x-axis. The negative of this sum thus gives the phase of the complex number.

$$\angle \mathbf{c} = -\sum_{i=1}^{N} \theta_i \tag{4.39}$$

4.5.2. CORDIC-Based Computation of Cosines and Sines

We can use a procedure very similar to the one used in Section 4.5.1 to compute the cosine and sine values of a given phase angle. In this case, we start off with a complex number, of unit magnitude, that is aligned along the *x*-axis. This is represented by the vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$. We rotate this complex number with progressively increasing precision, till it is rotated by the given phase angle. The final values of the *x* and *y* co-ordinates of the rotated vector are the cosine and sine values respectively of the phase angle. Again the rotations are performed using only shifts and adds, and the rotation performed in each iteration is a progressively smaller angle thus increasing the precision with each iteration. Also the sign of the rotation in each iteration is so chosen that the total rotation comes closer to the given phase angle.

4.5.3. Results from Implementation

The CORDIC-based GCC-PHAT computation was implemented on the ADSP-21065L DSP from Analog Devices Inc. The ADSP-21065L is a 32-bit floating point DSP. Real data was recorded using a 4-element 3-dimensional microphone array with white noise as the source signal. A frame of 2048 samples of the recorded data was used to compute the GCC-PHAT functions for all six pairs of microphones. First a 2048-point DFT samples corresponding to each channel data was computed using the FFT algorithm. Six sets of XPSD samples were computed using complex multiplication. Next the CORDIC based PHAT weighting was implemented to generate six sets of GXPSD samples. The CORDIC algorithms that were implemented were for a total of 24 iterations. Next the GCC-PHAT was computed from the GXPSD for 181 different time delays. These time delays corresponded to each degree of bearing angle from –90° to +90° for a pair of microphones. A sample GCC-PHAT function between microphones 3 and 4 that was computed using the implementation is shown in Figure 4.23. Figure 4.24 shows the difference between the two signals in Figure 4.23.

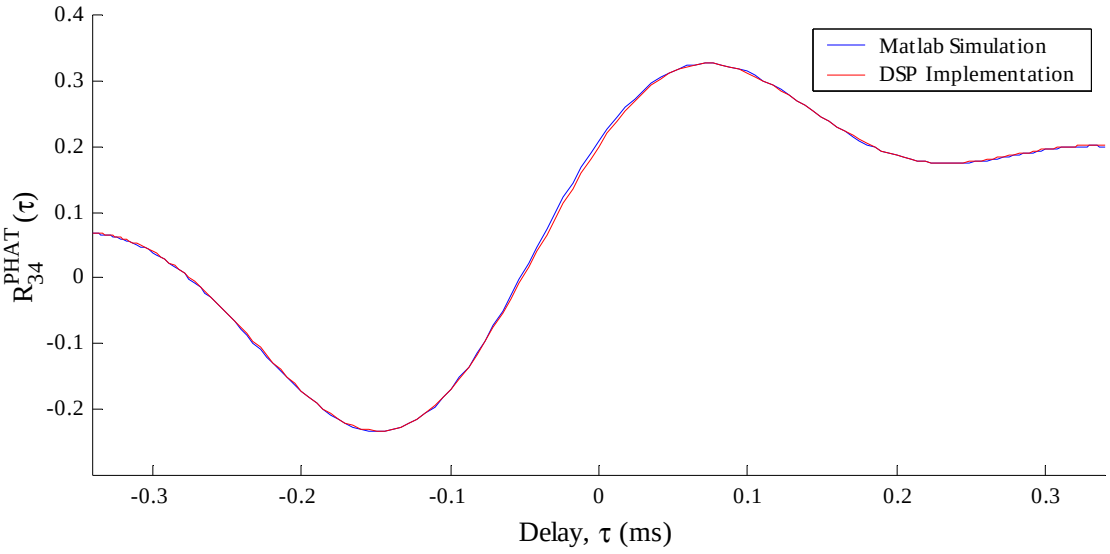

Figure 4.23 ***Sample GCC-PHAT obtained from the CORDIC-based DSP implementation of the phase transform.***

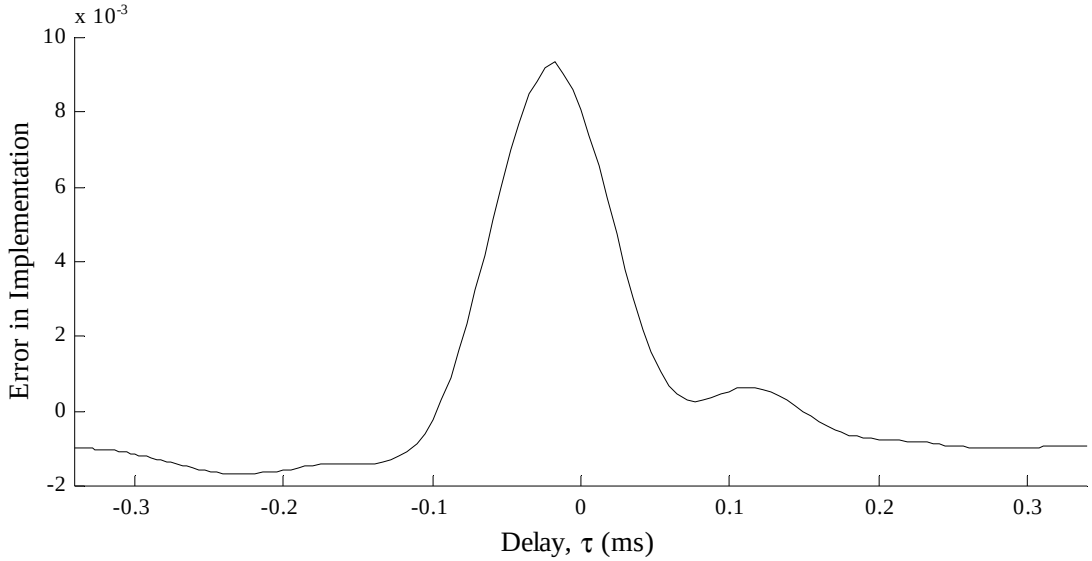

Figure 4.24 ***Error in implementation obtained by subtracting the GCC-PHAT obtained from the DSP implementation from that obtained from simulation.***

For comparison purposes the result of a Matlab simulation is also shown in Figure 4.23. The DSP implementation appears to be very accurate. The difference is observed to be very small.

5. The Time Delay Selection (TIDES) Algorithm

5.1. Data Acquisition Hardware

For this research we used the Analog Devices ADSP-2181 DSP evaluation board and the AD-7874 A/D converter to perform data acquisition from the microphone array. A schematic of the wiring between the AD-7874 and the ADSP-2181 is shown in Figure 5.1. The sensors are the powered microphones. These microphones were powered using batteries fed through a voltage divider network. The signal conditioning basically involves a two-stage active band-pass filter as shown in Figure 5.2. The input series capacitor acts as high-pass filter blocking DC and some low frequencies such as the power supply hum. The parallel feedback capacitor acts as a low-pass filter taking out all the high frequency noise. The gain of each stage of the filter is approximately 47.

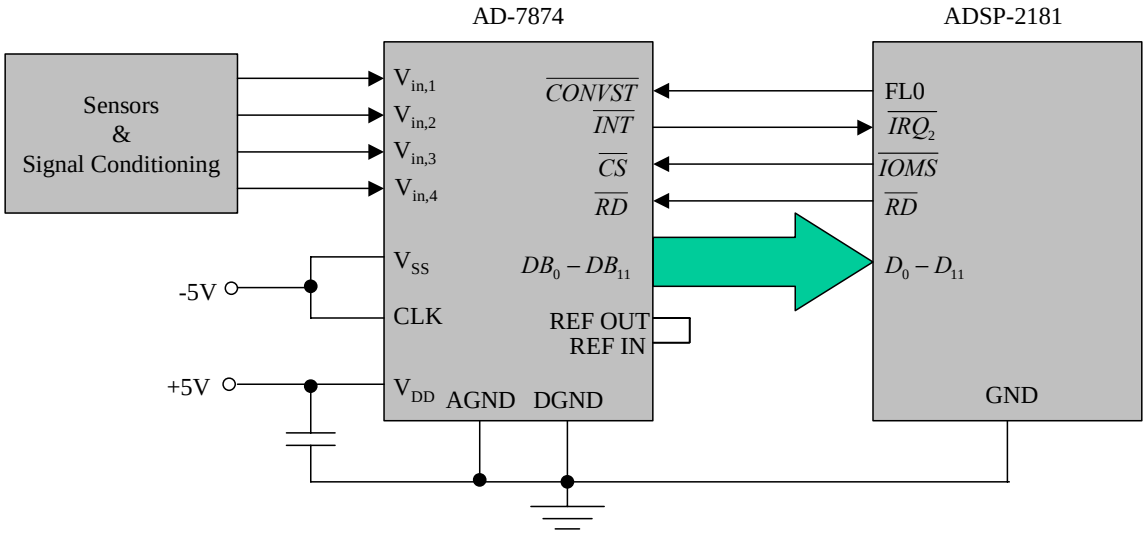

Figure 5.1 ***Schematic of interface between the A/D and the DSP for data acquisition.***

The AD-7874 is a 4-channel, 12-bit A/D. It is capable of sampling all four channels at the same time using a single hardware command-signal with arbitrarily small phase errors between the channels. This is also true across multiple AD-7874 chips. The chip works from two power supplies of ±5 V and accepts 4 input analog channels in the range ±10 V. These signals are converted to 12-bit 2's complement numbers, which become available on 12 output

pins as parallel outputs. Any signal out of the input voltage range will be clipped to the maximum or minimum value of the 12-bit output word.

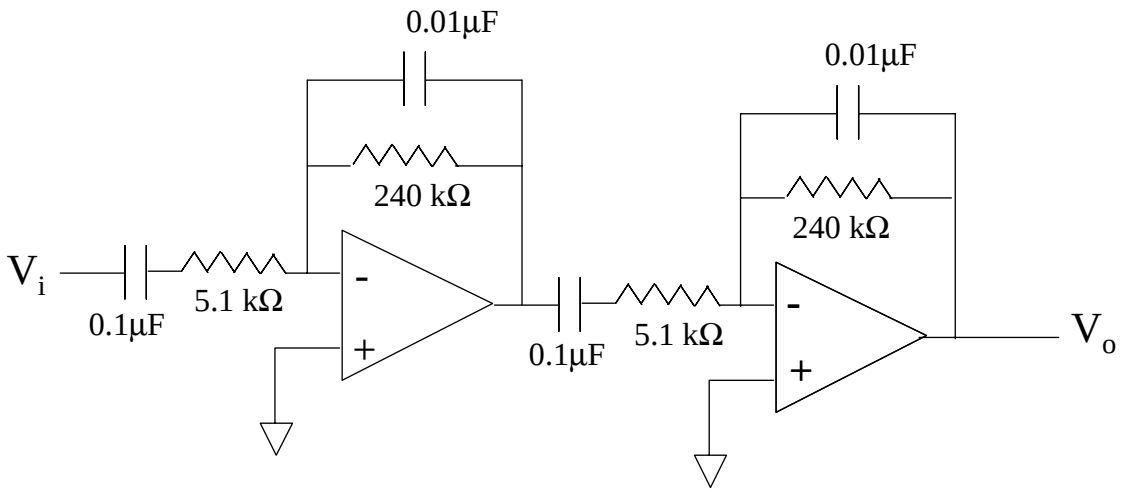

Figure 5.2 ***Two stage active band-pass filter used to condition the microphone signal.***

The sequence of events for each conversion process is as follows. Conversion is initiated by the DSP by asserting the $\overline{CONVST}$ signal. Once the $\overline{CONVST}$ signal is asserted, the A/D samples all four channels simultaneously and converts the instantaneous voltage values to 2's complement numbers using the internal clock. These 4 numbers are then transferred to 4 output data registers. On completing this operation the A/D sends out an interrupt $\left(\overline{INT}\right)$ signal to the DSP. On receipt of the $\overline{INT}$ signal from the A/D the DSP has to perform 4 successive read operations from some address location. This address location has to be properly decoded to the $\overline{CS}$ signal of the A/D. So each read by the DSP asserts both the $\overline{CS}$ and the $\overline{RD}$ signals of the A/D and, consequently, each of the converted 12-bit words become available at the output pins in succession. These output pins can be connected to the data pins of the DSP to get the words into the DSP core. This whole process is repeated for each sampling operation.

The CLK pin of the A/D is tied to the negative supply voltage. This forces the A/D to use its internal clock for the conversion. Also the REF-IN pin is directly tied to the REF-OUT pin. The timer on the DSP has been programmed to generate interrupts at regular intervals equal to the required sampling period. Within the service routines of these interrupts the flag FL0 is reset to 0 and then set back to 1 after some time. The FL0 is connected to the $\overline{CONVST}$. Thus the A/D receives $\overline{CONVST}$ signals at regular intervals.

The $\overline{INT}$ pin of the A/D is connected to the $\overline{IRQ_2}$ pin of the DSP. Therefore whenever the A/D finishes its conversion an $\overline{IRQ_2}$ interrupt is generated in the DSP. Within the service routine of $\overline{IRQ_2}$, we read from the A/D four successive times, once for each channel. Also the $\overline{CS}$ pin of the A/D has been connected to the $\overline{IOMS}$ pin of the DSP. This means that any address in the IO memory-bank of the DSP selects the A/D for read operations.

5.2. Effect of the Phase Transform

As an initial experiment we sought to study the effect of the phase transform on the accuracy of the time delay estimates and therefore on the accuracy of the DOA estimates. First a 4-element linear array was constructed with a spacing of 20 *cm* between adjacent elements. Signals were recorded from five different directions namely –30°, 0°, 30°, 60° and 90°. DOA estimation was performed using three methods — time-delay estimation using cross-correlations, time-delay estimation using GCC-PHAT and direct DOA estimation using SRP-PHAT. Figure 5.3 shows the framewise DOA estimates for an actual angle of 30°. It clearly shows that using the PHAT has reduced the impulsive errors that are present in the results obtained from regular cross-correlation. Figure 5.4 shows clearly that the reliability of the estimates has improved because of the phase transform. These reliability rates were computed cumulatively for all five angles for which we had experimental data.

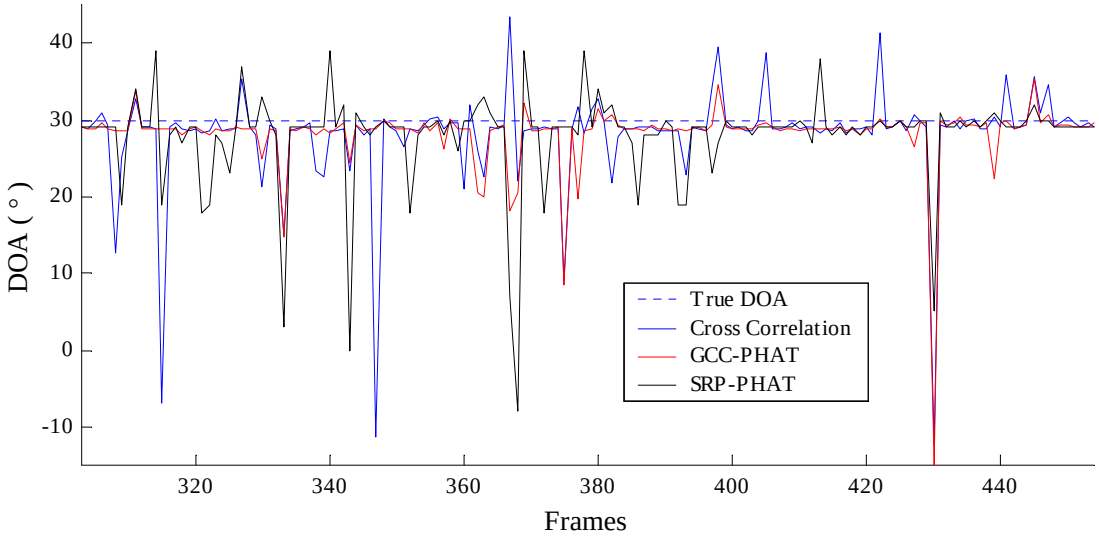

Figure 5.3 ***Framewise DOA estimates for linear array with true DOA = 30°.***

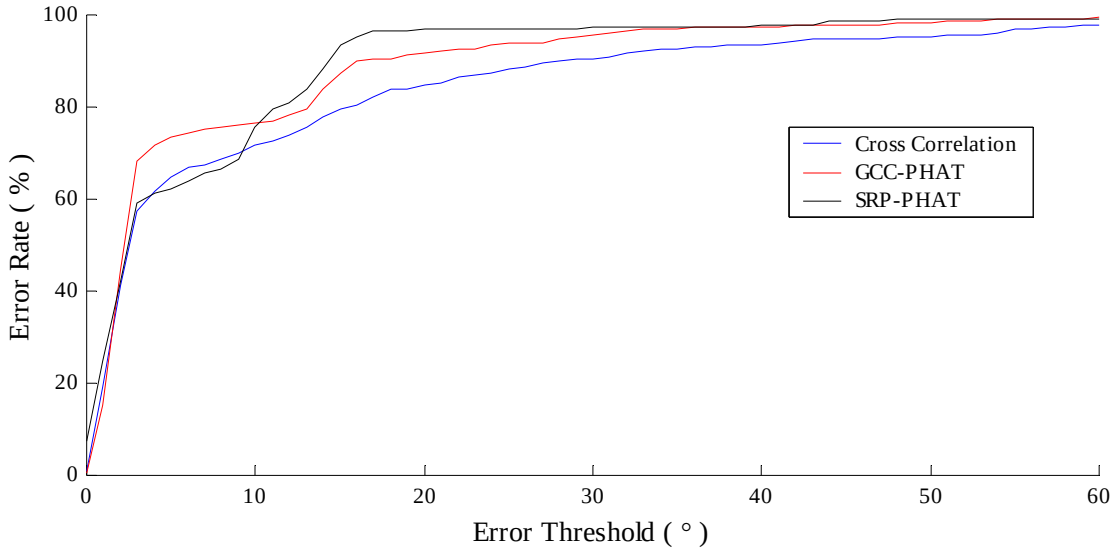

Figure 5.4 ***Reliability rates for the estimates shown in*** *Figure 5.3* ***showing improvement with PHAT.***

Next a 3-dimensional array containing 7 elements was constructed and speech signals were sounded from an azimuth of 65° and an elevation of 20°. The locations of the microphones of this array in Euclidean space are represented by

$$\mathbf{M} = \begin{bmatrix} -0.1 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & -0.1 \end{bmatrix} \tag{5.1}$$

Each column of **M** is the position of a microphone. Geometrically this array has three microphone elements along each of the three axes with the middle element along each axis being a common element. The actual hardware available to us could only support a maximum of 4 microphones. To overcome this problem the recording was done separately for each dimension of the array using three microphones at a time. The middle microphone was not moved and the other two microphones were moved around to record signals along other dimensions. As long as time delays are estimated between microphones within one dimension using signals from the same recording session, the results would be correct. This gives us 3 time delays for each dimension, giving a total of 9 time delays. The results from this recording are shown in Figure 5.5 and Figure 5.6. Again we can observe that PHAT improves the reliability of the estimates.

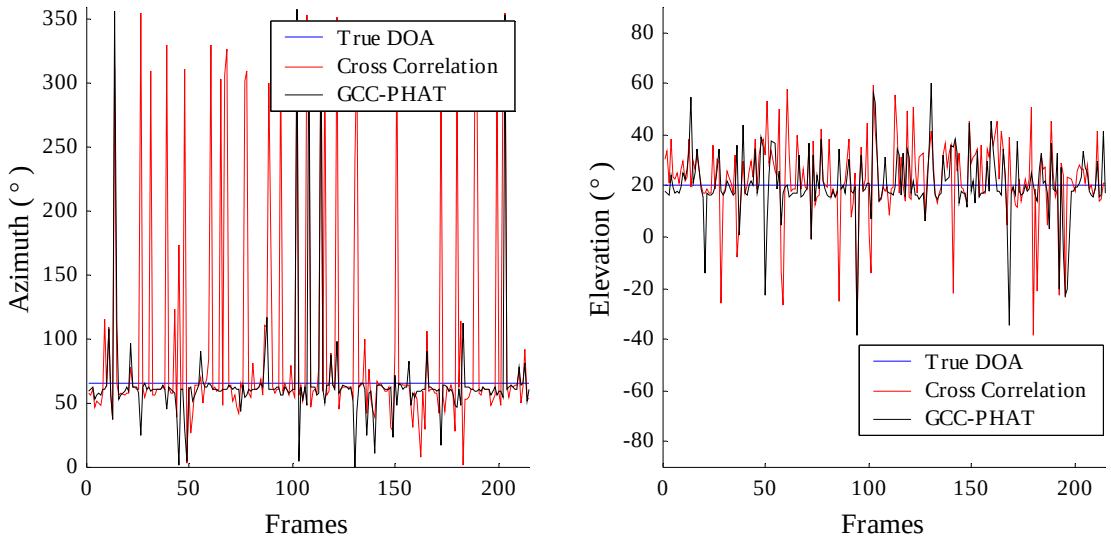

Figure 5.5 ***Framewise azimuth and elevation estimates with and without phase transform.***

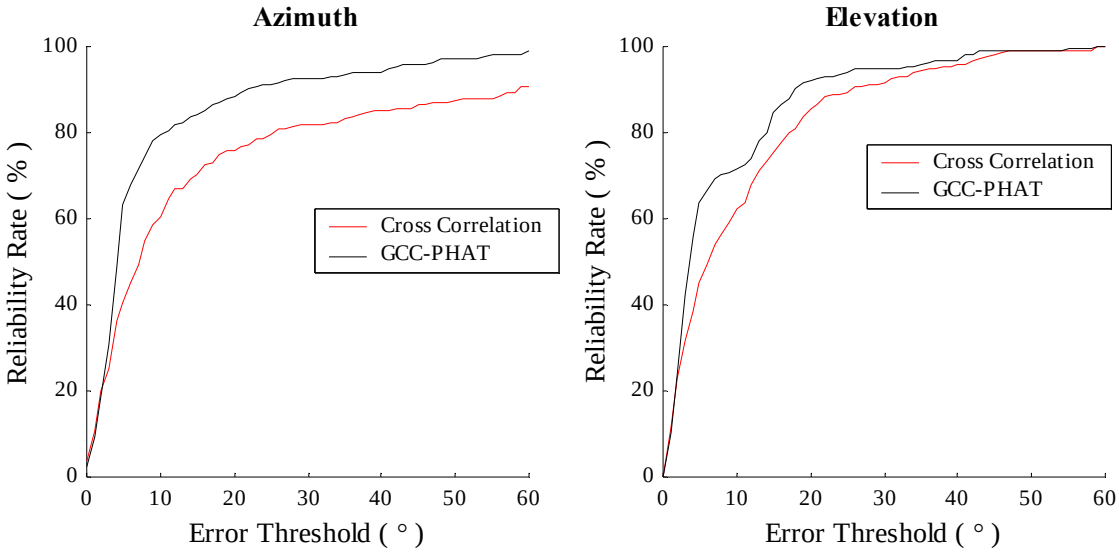

Figure 5.6 ***Reliability rates of both azimuth and elevation showing improvement with PHAT.***

5.3. Bias in Estimates

When DOA estimation was performed on all the data that we had recorded with the ULA with a separation of 20 *cm*, we noticed that the estimates were biased. Figure 5.7 shows the results obtained using both GCC-PHAT and SRP-PHAT for six different DOAs. The results show a bias towards zero, the value of which keeps increasing as we move away from the broadside.

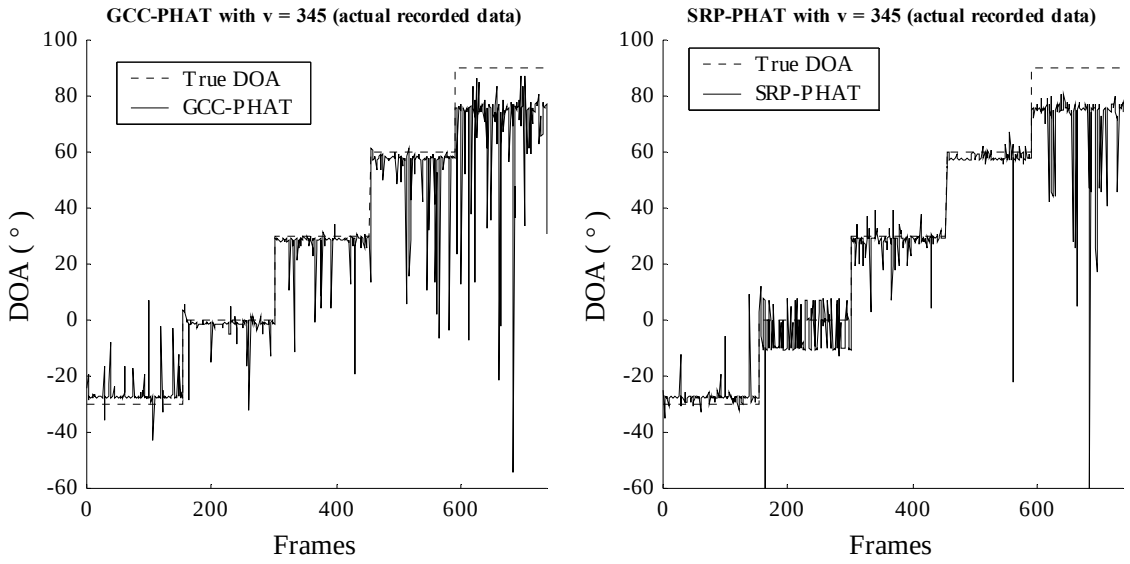

Figure 5.7 ***DOA estimation results for actual recorded data with both GCC-PHAT and SRP-PHAT using v = 345 m/s showing increasing bias with increasing angular separation from the broadside.***

Figure 5.8 shows similar results for simulated data. The latter results do not show any bias in the DOA estimates. There appears to be some kind of a bias at 90° for the GCC-PHAT based method, but the results do not show a trend of increasing bias as we move away from the broadside.

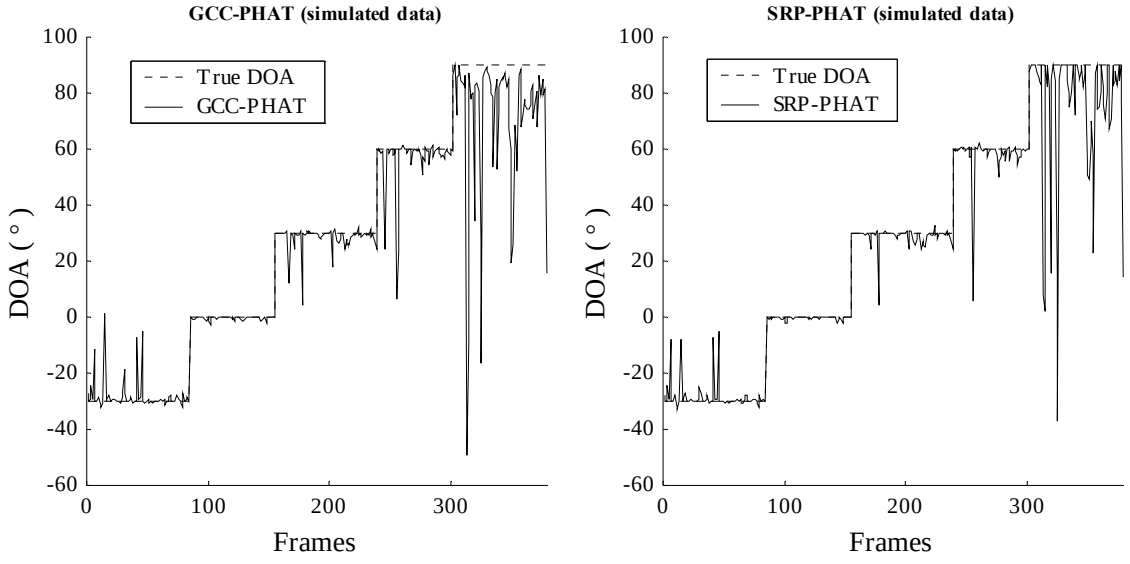

Figure 5.8 ***DOA estimation results for simulated data with both GCC-PHAT and SRP-PHAT does not show any biasing.***

This suggests that we might be using an incorrect value of the velocity of sound to compute the DOA estimates. The value that was used in the computations was v = 345 *m/s*. Since the bias is towards zero, we might have under-estimated the actual velocity of sound. If the velocity used in the computation is smaller than the actual velocity, then the same time-delays result in smaller range differences and thus DOA estimated closer to zero. Figure 5.9 shows the results from the actual recorded data, this time re-computed with v = 355 *m/s*. Now we can observe that the bias has more or less disappeared. There is still a small bias in the GCC-PHAT algorithm at DOA = 90°, but it has decreased from approximately 15° in Figure 5.7 to approximately 5° to 6° in Figure 5.9. Also this bias is absent in the SRP-PHAT method, which also agrees with the simulation results. Champagne et. al. [20] have studied the effect of room reverberation on the performance of time-delay estimation. One of the results presented by them is that the bias in the delay estimates keeps increasing with reverberation. This bias is thought to originate from strong initial echoes whose time differences of arrival are close to that of the direct signals. The actual value of this bias is difficult to predict and is dependent of factors like the auto-correlation of the source signal, bearing-angle of the source with respect to the line joining the two microphones and also the actual nature of reflection in the room.

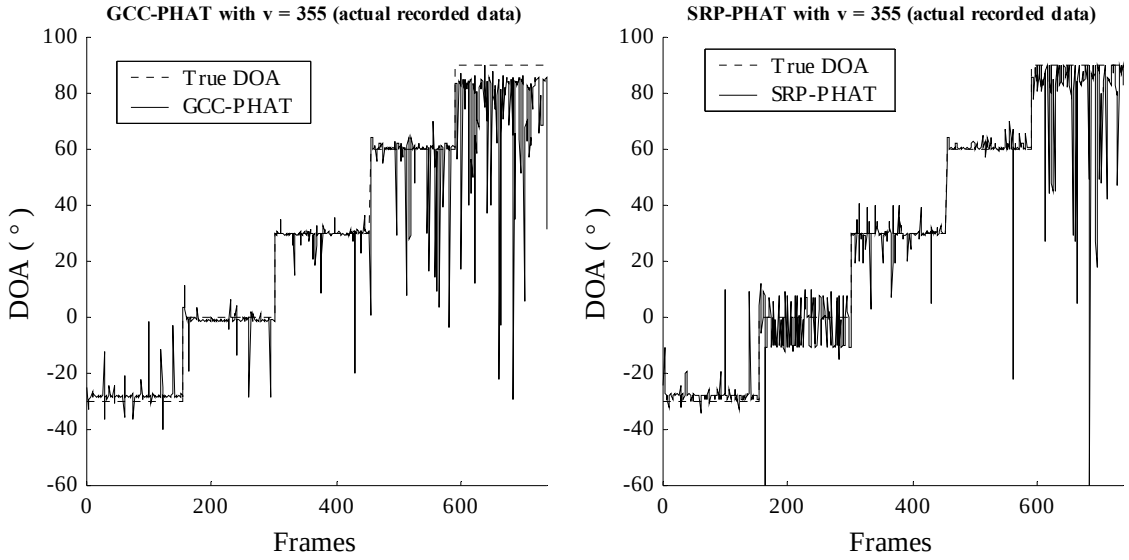

Figure 5.9 ***DOA estimation results for actual recorded data with both GCC-PHAT and SRP-PHAT using v = 355 m/s showing no bias.***

One other issue in these results is the bias that is observed at DOA = –30° which seems to be present with both methods. The explanation for this could be that when we manually placed the loudspeaker, the actual DOA was not –30°, but slightly closer to the broadside.

5.4. SNR Based Thresholding of the GXPSD

In this section we look at a method to improve the reliability of the DOA estimates by thresholding the GXPSD samples. The GXPSD is computed from the DFT of the array signals at equally spaced frequency samples. Depending on the nature of the source signal, the reflections from walls and SNR, not all of these frequencies contain signal frequencies with strong enough power to give us direction information. On the other hand many of these frequencies come from noise, which is independent of the array elements. Thus including these frequencies could actually degrade the performance of the algorithm. Thus it makes sense to exclude these low-power frequencies in computing the GXPSD.

Simulations were performed using a linear array of 4 microphones separated by 10 *cm*. Again it was placed in a room of dimensions $5m \times 5m \times 5m$ and with a reverberation time of 100 *ms*. The array signals were simulated for an SNR of 30 dB. The GXPSD was computed only at those frequencies for which the normalized powers in both array signals were above –30dB. Zeros were filled for those frequencies that did not satisfy this criterion. Figure 5.10 shows the magnitudes of the DFT samples from a sample frame of array signals. Figure 5.11 shows magnitudes of the PHAT weighted DFT samples with SNR based thresholding. Notice that the PHAT weighted DFT samples reduce to zero for those frequencies where the magnitudes of both channels are not above the threshold. Figure 5.12 clearly shows the performance improvement obtained from using the SNR based thresholding. It can be observed that the improvement obtained in the TDE-based method is greater than that in the SRP-PHAT method.

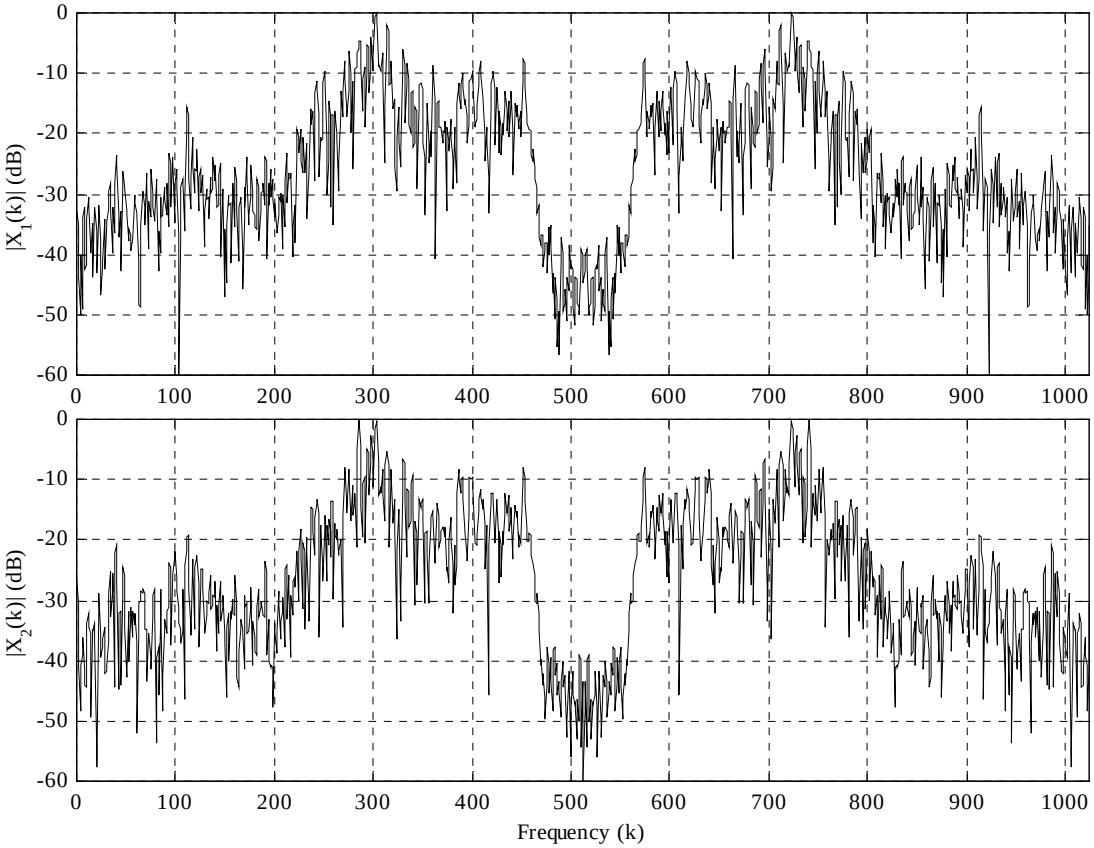

Figure 5.10 ***Frequency content of two array signals from a sample frame.***

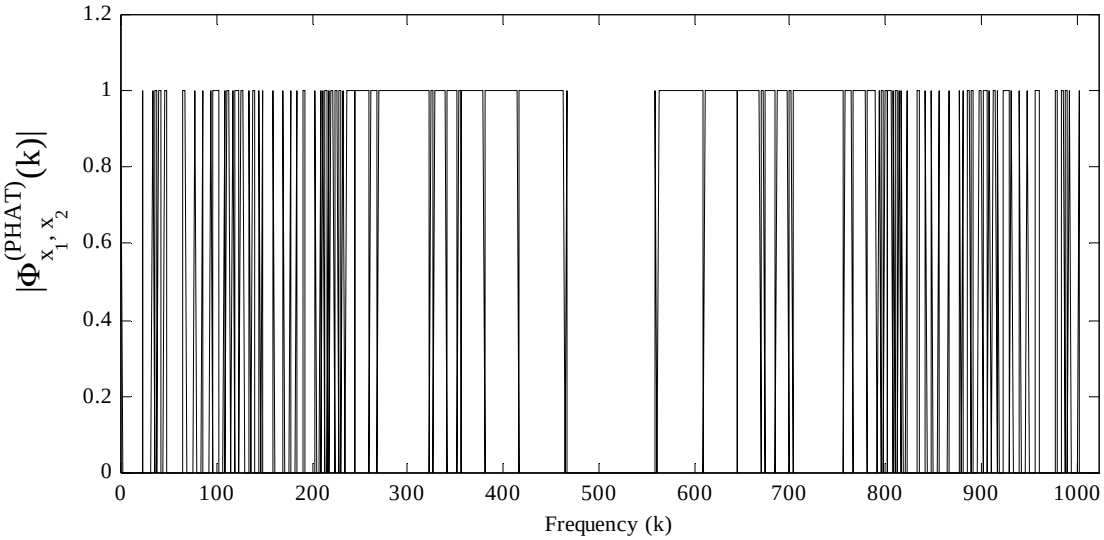

Figure 5.11 ***PHAT weighted GXPSD for the same sample frame.***

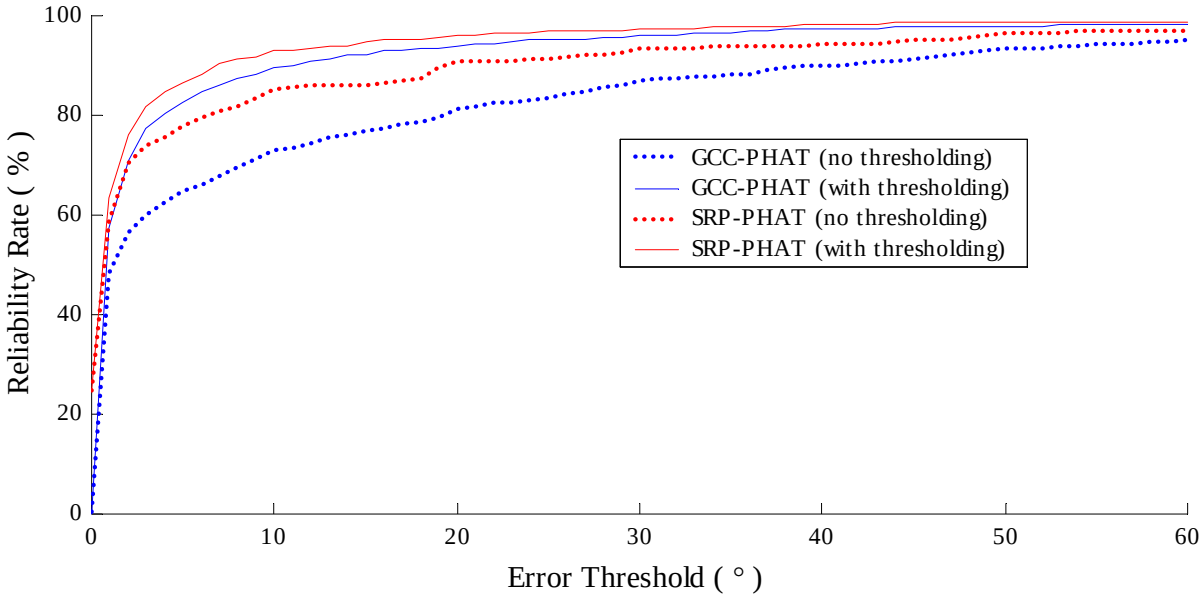

Figure 5.12 ***Performance improvement with SNR based thresholding (simulation for 30 dB SNR).***

SNR based thresholding was tried out on actual recorded data as well. Signals were recorded from five different angles for linear arrays with two different separations between adjacent microphones. The chosen angles were –60°, –30°, 0°, 30°, 60° and 90°. Framewise estimation results for the case where the source is at 30° is shown in Figure 5.13.

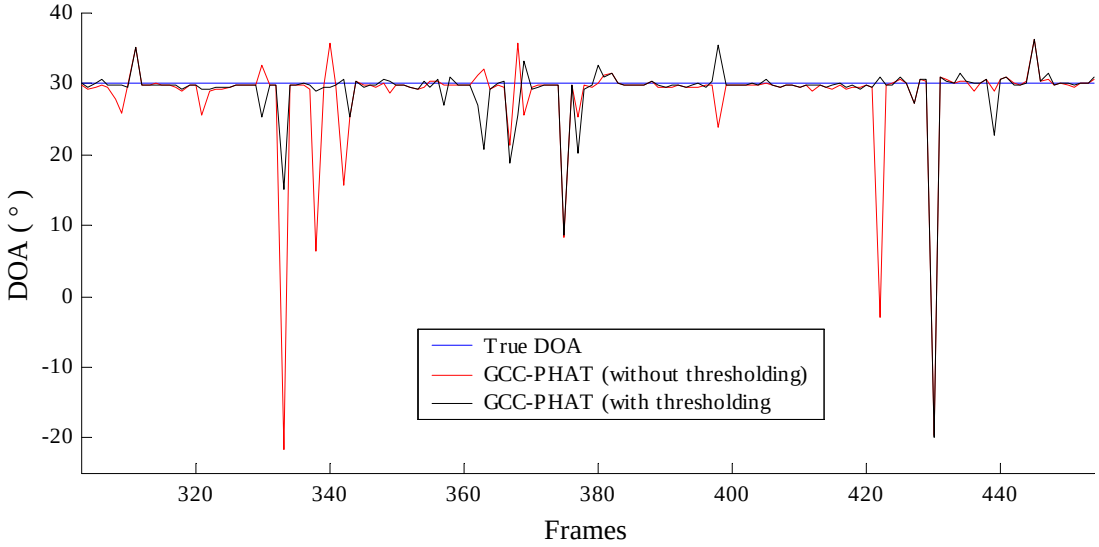

Figure 5.13 ***GCC-PHAT based frame-wise DOA estimates for linear array with and without SNR based thresholding.***

We observe in Figure 5.13 that some of the jumps that are present when SNR thresholding was not used have been eliminated because of SNR-based thresholding.

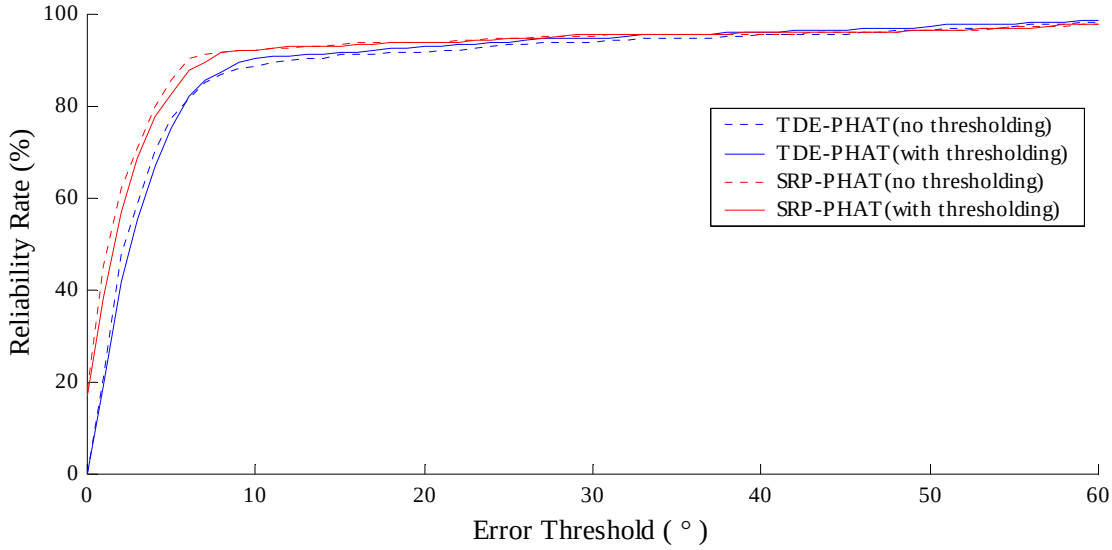

Figure 5.14 ***Reliability rates with and without thresholding for actual recorded data (linear array with separation of 5 cm).***

Two different microphone separations – 5 *cm* and 20 *cm* – were used for the recordings. Figure 5.14 shows the reliability rates obtained from the recordings with a separation of 5 *cm* and Figure 5.15 shows the reliability rates from the recordings with a separation of 20 *cm*.

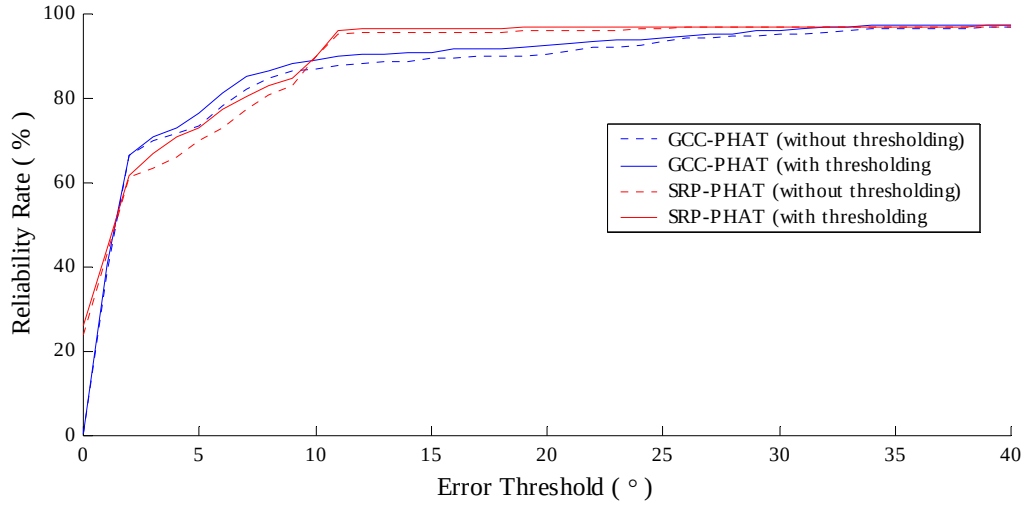

Figure 5.15 ***Reliability rates with and without thresholding for actual recorded data (linear array with separation of 20 cm).***

Again a slight improvement in performance can be observed especially for the time delay estimate based algorithm. The SRP-PHAT algorithm does not show any significant improvement.

5.5. Symmetric Extension of Frame Data

Another technique that improves the results is the symmetric extension of the frame signals. Consider a linear microphone array of 4 elements with a separation of 20 *cm.* The array signals were recorded at 8 *kHz.* Each frame consists of 512 samples of signal, which amounts to 64 *ms* of signal time. The signal in each channel is symmetrically extended on both sides. This is done by taking the first half of the signal, flipping it and attaching it in front of the signal and then taking the second half of the signal, flipping it and attaching to the end of the signal. The extended signal is thus twice the size of the original signal. This kind of extension does not add any new information, but it re-emphasizes the existing information in the signal and removes some edge effects. For example, consider one particular frame of the recorded data. The signal in this case was recorded from an angle of 60° and hence the time delay between Mic1 and Mic4 should be 1.506 *ms*. Figure 5.16 shows the GCC-PHAT function against delay. It shows that the curve peaks at the wrong delay of 0 *ms*. A peak does show up at approximately –1.5 *ms*, but this peak is only the second strongest peak in the function.

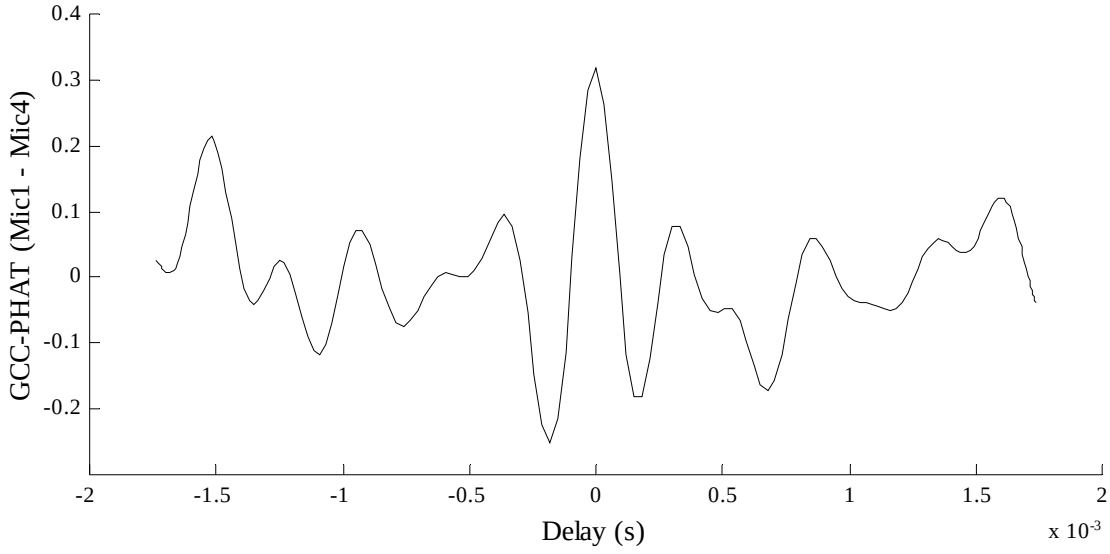

Figure 5.16 ***GCC-PHAT for Mic-pair 1-4 from frame no. 20.***

Figure 5.17 shows the GCC-PHAT function computed with symmetric extension of the frame data. The peak at the wrong delay of 0 *ms* has been suppressed and the peak at the delay of approximately –1.5 *ms* has been enhanced. In this case, since the data is circularly symmetric, the GCC-PHAT function, which is computed as a circular cross-correlation, becomes an even-function with the same values for equal positive and negative delays. Hence we get an equally strong peak at approximately 1.5 *ms.* Now we do not know whether the correct delay is the one on the positive side or the one on the negative side. Moreover, when the delay happens to be very small (for angles close to the broadside), the peaks from the positive and negative sides occur so close together that they interact to produce a single peak at delay 0 *ms.* A window can be applied to the extended frame signal to remove its circular symmetry. Figure 5.18 shows the GCC-PHAT function computed by applying a Hanning window to the extended signal frame. The spurious peak at 1.5 *ms* has been eliminated and we get one strong peak at approximately –1.5 *ms.*

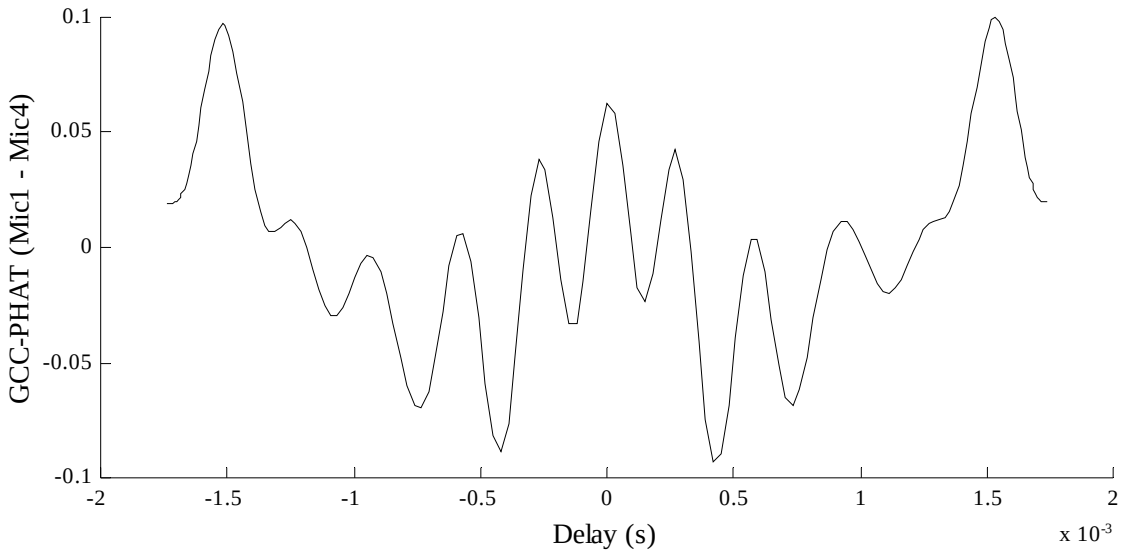

Figure 5.17 ***GCC-PHAT for Mic-pair 1-4 from frame no. 20 with symmetric extension.***

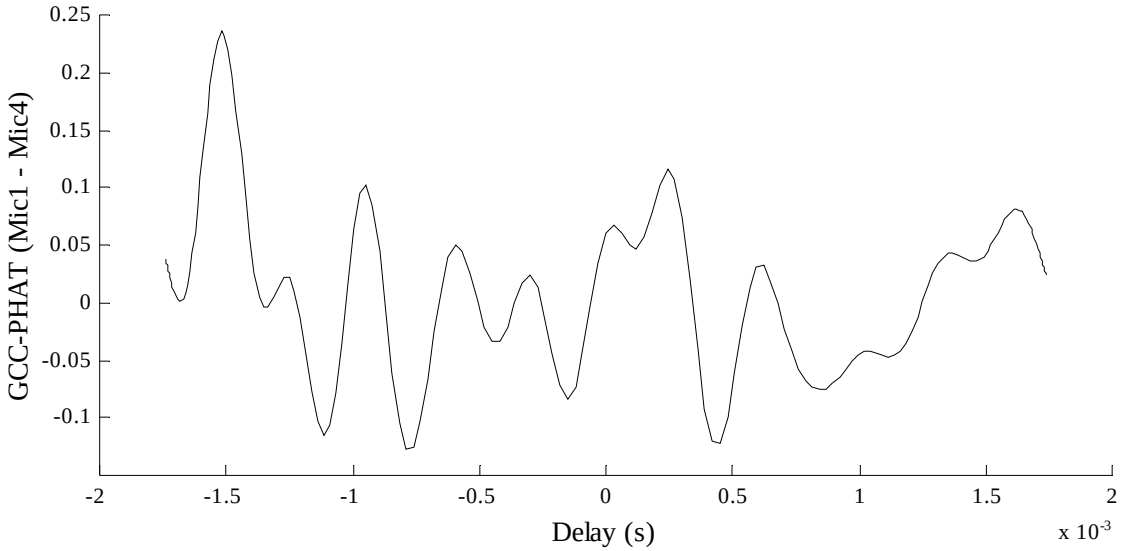

Figure 5.18 ***GCC-PHAT for Mic-pair 1-4 from frame no. 20 with symmetric extension and windowing.***

Figure 5.19 shows a portion of the frame-wise DOA estimates for the same 4-element array and an incident direction of 60°. In these simulations, the fourth power of a Hanning window was multiplied with the extended signal. In other words the windowing was done four times over and over again. In many cases, when there is an error in the DOA estimate, the use of symmetric extension and windowing is seen to reduce the error. There are a few cases when the use of symmetric extension actually increases the error in the estimates. So it makes sense to look at the reliability rate to see if symmetric extension actually improves the reliability of the estimates. Figure 5.20 shows the reliability rate for the 60° case. It shows a small improvement in reliability. On the other hand, we can see in Figure 5.21 that the reliability rate for the 0° case actually degrades marginally because of symmetric extension.

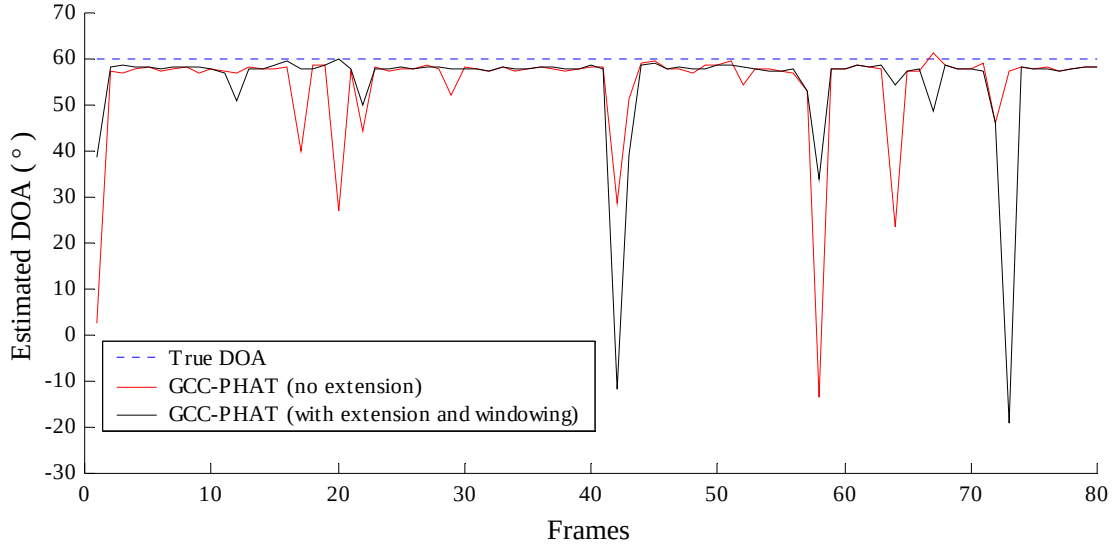

Figure 5.19 ***Frame-wise DOA estimates showing improvement with symmetric extension and windowing.***

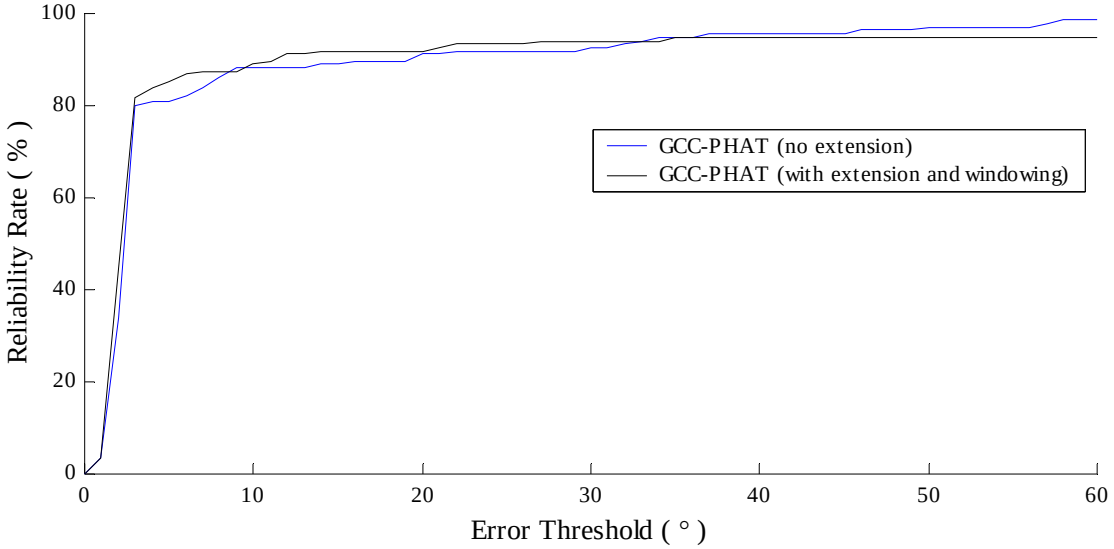

Figure 5.20 ***Reliability rates for incident DOA = 60°.***

Figure 5.22 shows that symmetric extension clearly improves results for the 90° case. In general it was observed that the larger the time delay, the larger the improvement due to symmetric extension and windowing.

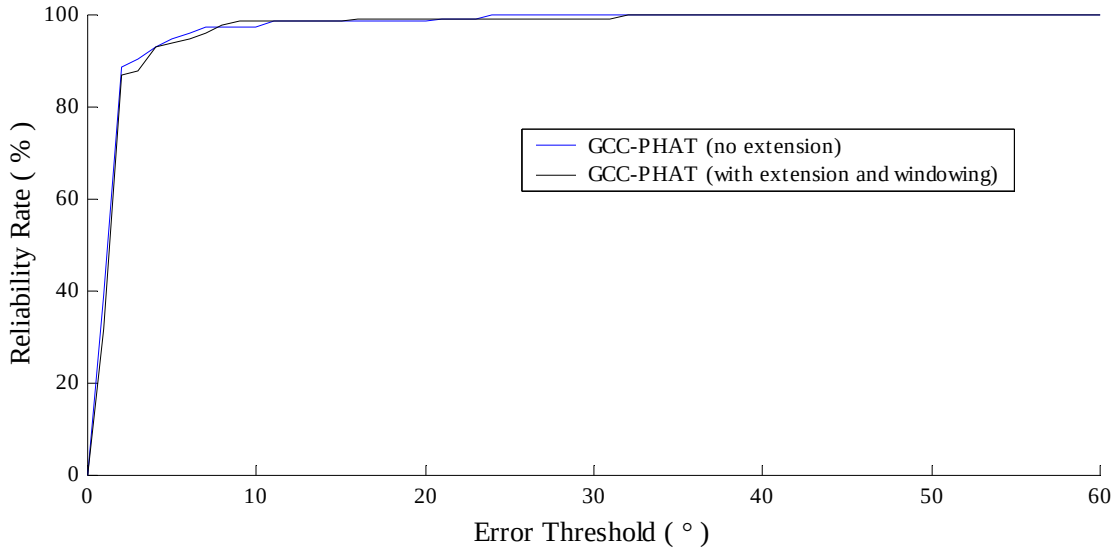

Figure 5.21 ***Reliability rates for incident DOA = 0°.***

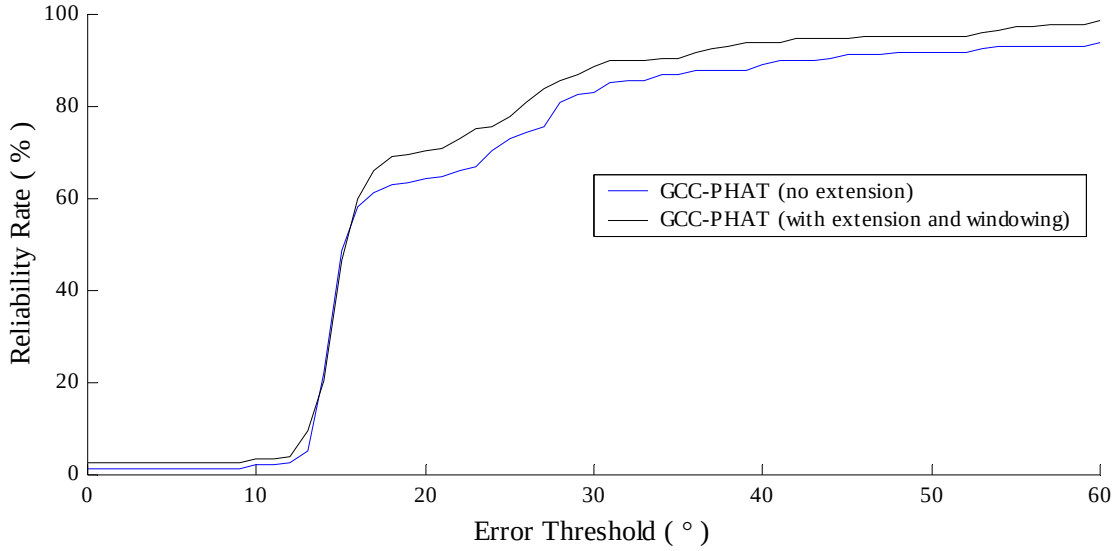

Figure 5.22 ***Reliability rates for incident DOA = 90°.***

5.6. Time-Delay Selection (TIDES) Algorithm

In all the cases that we encountered above, whether with simulation data or actual recorded data, the reason for the impulsive errors is that the algorithm picks the wrong time delays. Because of strong reflections, sometimes a delay corresponding to a reflection gives a higher peak in the GCC-PHAT, and the algorithm then picks this wrong delay. For example, Figure 5.23 shows the framewise time delay estimates between Mic-1 and Mic-2 for the 7-

element array. These estimates were obtained from real recorded data. The figure shows that in several frames the estimates have large errors in them.

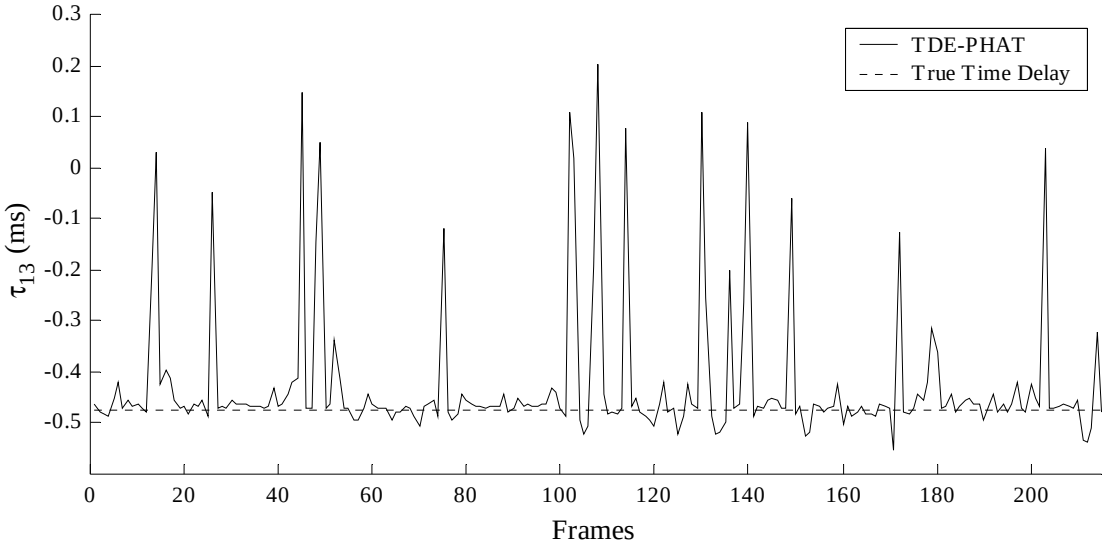

Figure 5.23 ***Time delay estimates between Mic-1 and Mic-2 from data recorded using a 7-element array.***

Figure 5.24 shows sample GCC-PHAT computed from two arbitrary frames – frame 14 and frame 45. In either case the strongest peak of the GCC-PHAT is not at the correct delay, which is approximately –0.47 *ms*.

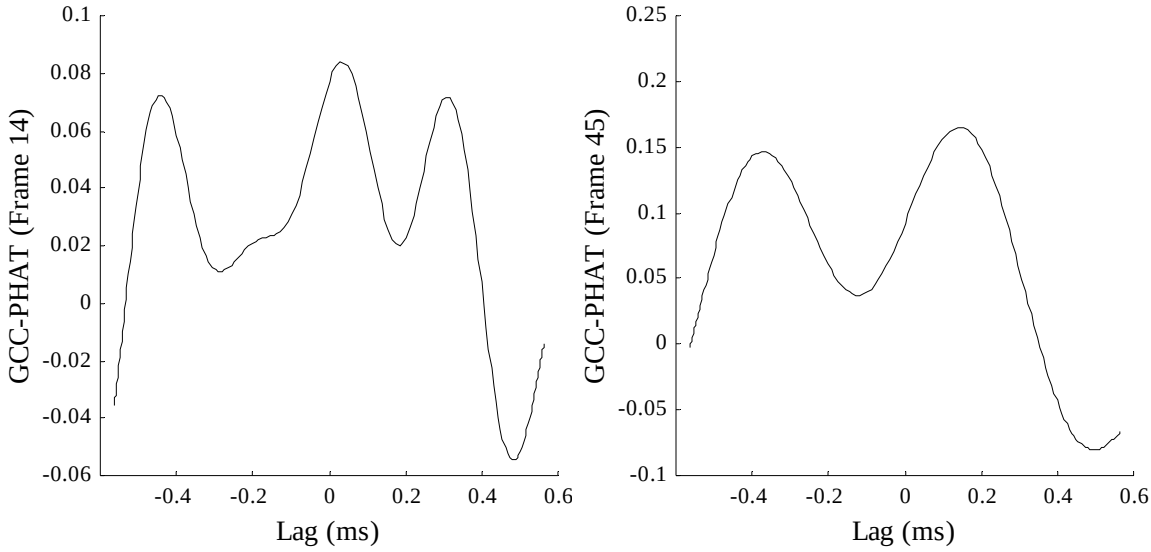

Figure 5.24 ***Sample cross-correlations that show local maxima at wrong and correct time-delays.***

However we observe in Figure 5.24 that there are peaks at the correct delays though they are not the strongest. Thus we find that in many cases, the information is available, but we do not have a method to pick the required peak. Figure 5.25 shows all candidate time delays between Mic-1 and Mic-2 by considering up to a maximum of three peaks in the order of their strength. We can see that in most cases the correct delays correspond either to the strongest or to the second-strongest peak. In some cases the correct delay corresponds to the third strongest peak. We observe that there are very few cases where there is no GCC-PHAT peak at the correct delay.

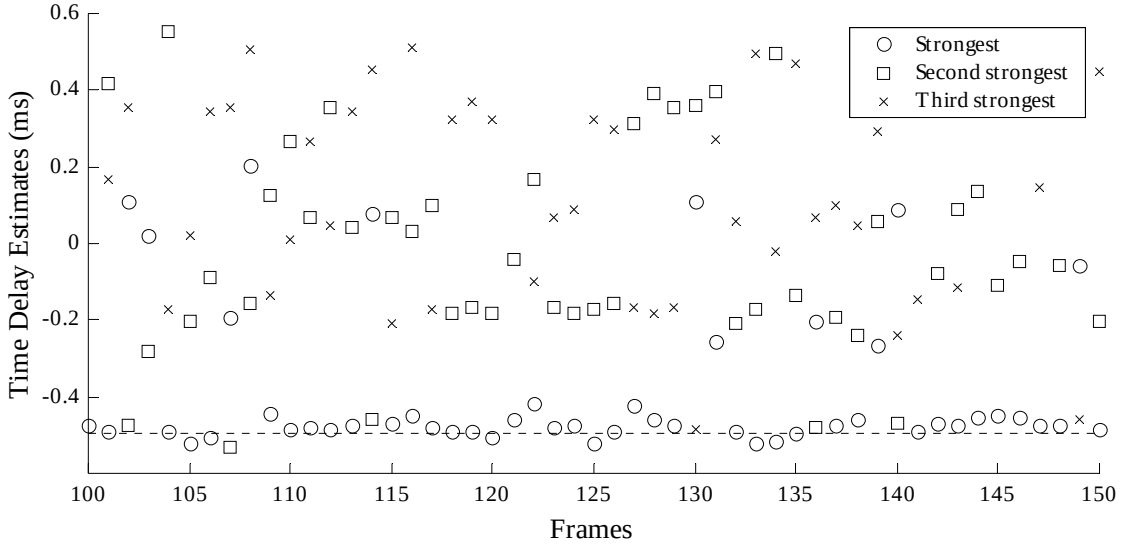

Figure 5.25 ***Framewise candidate time delays between Mic-1 and Mic-2.***

The latter observation leads us to consider all the delays at which peaks occur and then select one of these time-delay candidates based on some criterion. The DOA estimation algorithm based on considering multiple time-delays for each pair of microphones was named the Time Delay Selection (TIDES) algorithm. The steps involved in the TIDES algorithm are as follows:

1. For each pair of microphones:
    a. Compute the GCC-PHAT for the range of all possible delays.
    b. Find the delays at which the GCC-PHAT has peaks. Save these time-delay candidates and the relative strengths of their peaks. The relative strength is defined as the strength of the peak divided by the strength of the strongest peak for the microphone pair. For

example, if $\tau_{i,j}^{CAND}$ is one of the candidate delays for the microphone pair *i-j,* then the relative strength for that candidate delay, $S_{i.j}^{CAND}$ is given by

$$S_{i,j}^{CAND} = \frac{R_{i,j}^{PHAT}\left(\tau_{i,j}^{CAND}\right)}{\max\left(R_{i,j}^{PHAT}\left(\tau\right)\right)} \tag{5.2}$$

2. Construct all possible sets of time-delays from the multiple time-delay candidates for each microphone pair. Construct all corresponding possible combinations of the relative strengths.
3. From among all the possible sets, select one set of time-delays based on one of the following minimization criteria:
   a. Minimum Weighted Least Squares Error (MWLSE).
   b. Minimum Weighted Time Delay Separation (MWTDS)
4. Compute the least squares DOA estimate using the selected set of time-delays.

Step 2 in the algorithm needs an explanation. Consider an example where the array has 3 microphones, m_1, m_2 and m_3. This means that we have three pairs of microphones to consider, m_1-m_2, m_1-m_3 and m_2-m_3. Let the GCC-PHAT for the m_1-m_2 pair have two peaks at delays t_1 and t_2. Let the GCC-PHAT for the m_1-m_3 pair have three peaks at t_3, t_4 *and* t_5. Finally let the GCC-PHAT for the m_2-m_3 pair have one peak at t_6. Then each column of the following matrix represents one set of possible time delay candidates.

$$td_{comb} = \begin{bmatrix} t_1 & t_1 & t_1 & t_2 & t_2 & t_2 \\ t_3 & t_4 & t_5 & t_3 & t_4 & t_5 \\ t_6 & t_6 & t_6 & t_6 & t_6 & t_6 \end{bmatrix} \tag{5.3}$$

In step 3, the metric to be minimized is computed for each column of td_{comb}. The set of time-delays that minimizes the metric is then used to compute a least squares estimate of the DOA. Next we take a closer look at the two criteria that were studied to select one of these sets of time-delays.

5.6.1. The MWLSE Criterion

Each set of time delays forms an over-determined system of linear equations. Each of these systems of equations can be solved in the least squares sense. For example we can get six different least squares solutions for the example described in the previous section. Each of these least squares solutions has an error associated with it. For example, let the least squares solution obtained from the first set of time-delays, $\boldsymbol{\tau}_1 = \begin{bmatrix} t_1 & t_3 & t_6 \end{bmatrix}^T$, be $\theta_{LS,1}$. Then the error in each of the equations can be written in vector form as

$$\mathbf{e}_1 = \mathbf{d}\sin\left(\theta_{LS,1}\right) - \boldsymbol{\tau}_1 v \tag{5.4}$$

Here $\mathbf{e}_1$ is a vector of the form $\begin{bmatrix} e_{11} & e_{12} & e_{13} \end{bmatrix}^T$ containing the errors, $\mathbf{d}$ is a vector of the form $\begin{bmatrix} D_{12} & D_{13} & D_{23} \end{bmatrix}^T$ containing the distances between each pair of microphones and v is the velocity of sound. For a 3-dimensional array, the error vector in (5.4) is written as

$$\mathbf{e}_i = \mathbf{A}^T \mathbf{u}_{LS,i} - \boldsymbol{\tau}_i v \tag{5.5}$$

Here $\mathbf{A}$ is a matrix, each column of which is the difference between the Euclidian co-ordinates of a pair of microphones, $\mathbf{u}_{LS,i}$ is a unit vector pointing along the estimated least squares direction, and the subscript i indicates the i^{th} set of time-delays. Once the error vector is computed, the weighted norm of the least squares errors is computed as

$$e_{wn,i} = \frac{\mathbf{e}_i^T \mathbf{e}_i}{\left(\mathbf{s}_i^T \mathbf{s}_i\right)^k} \tag{5.6}$$

Here $\mathbf{e}_i$ is the error vector computed from either (5.4) or (5.5) and $\mathbf{s}_i$ is a vector containing the relative strengths of the peaks at each of the delays. The length of $\mathbf{s}_i$ is the same as that of $\mathbf{e}_i$. The numerator in (5.6) computes the squared norm of the error vector. This quantity is then weighted by the k^{th} power of the norm of the relative strengths. Thus, the stronger the peaks are for the chosen time-delays, the lower the weighted error norm will be. The amount by which the relative strengths weigh the weighted error norm depends on the

parameter k. For the simulations performed as part of this research, k was set to 2. The numerator in (5.4) is a completely independent error estimate for the particular system of equations. There is no guarantee that this error will be minimum for the correct set of equations because the least squares solution is the one that minimizes this error independently for each set of equations. By weighting the numerator with the squared norm of relative strengths, we attempt to make these errors from different candidates comparable to each other. We then pick the candidate set of time-delays that minimizes this weighted squared norm of errors as the true set of time-delays. The least squares DOA is then computed from this chosen set of time-delays. Figure 5.26 shows the framewise DOA estimation results using the TIDES-MWLSE algorithm in black. The original algorithm, which picks the strongest peak as the true time-delay and is a maximum likelihood (ML) estimator, is shown in red. This result is for an actual recording with a 4-element linear array with a spacing of 20 *cm*. The figure shows that many of the impulsive errors have been corrected.

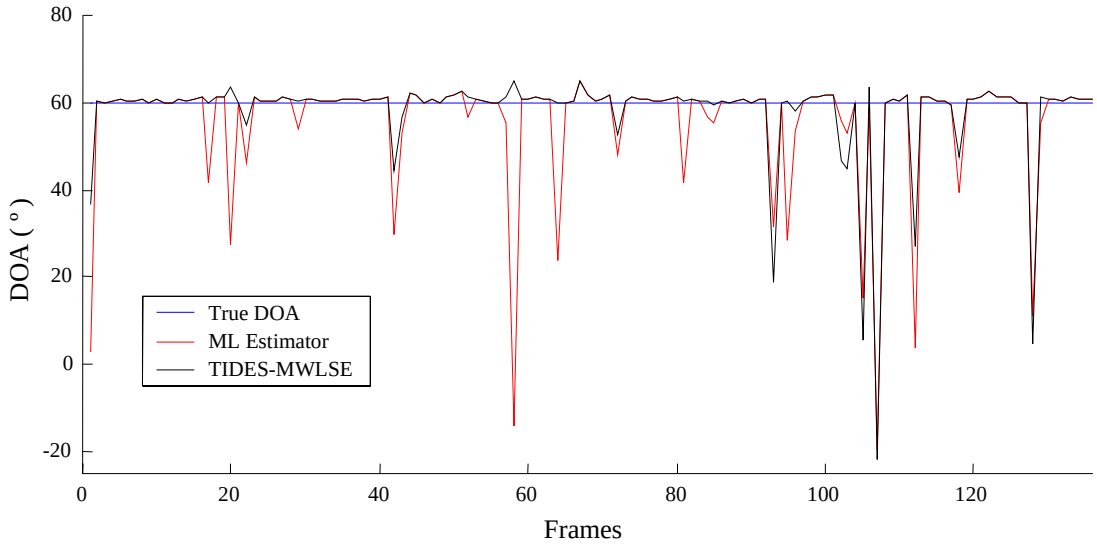

Figure 5.26 ***Framewise DOA estimates shows that the TIDES-MWLSE algorithm corrects many of the impulsive errors found in the ML estimator.***

Figure 5.27, Figure 5.28 and Figure 5.29 show the reliability rates of the two methods for incident angles of 30°, 60° and 90° respectively. The performance of the TIDES-MWLSE algorithm is good except for DOAs very far away from broadside. For example at a DOA = 90°, the performance of the TIDES-MWLSE algorithm is worse than that of the ML estimator. This

is because when the range differences between microphone pairs become very large, a larger number of reflections impinges on the microphones, thus giving more candidate TDEs. This increases the chance that a spurious TDE is picked over the correct one.

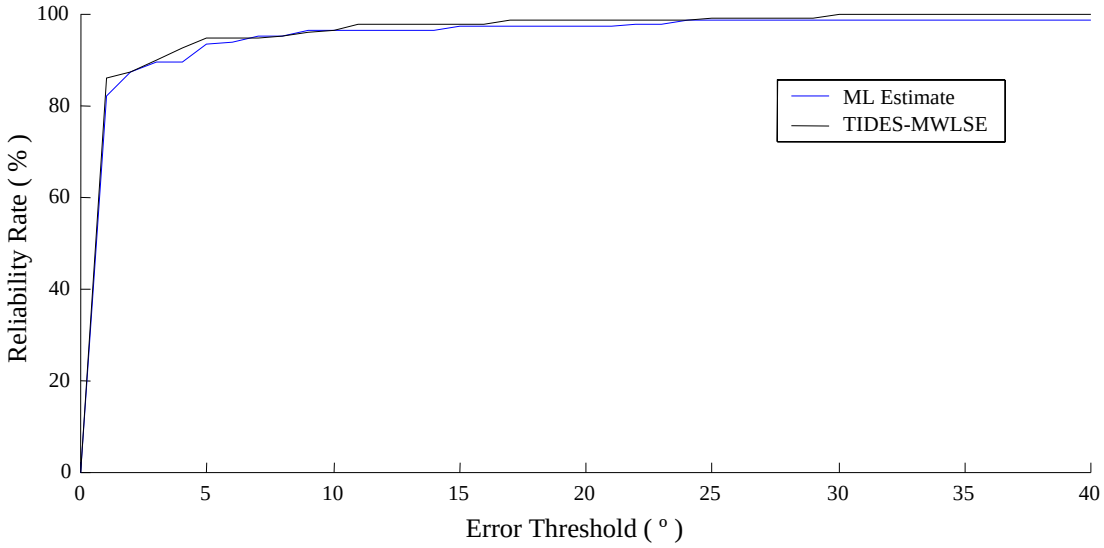

Figure 5.27 ***Reliability rates for DOA = 30 ° using TIDES-MWLSE.***

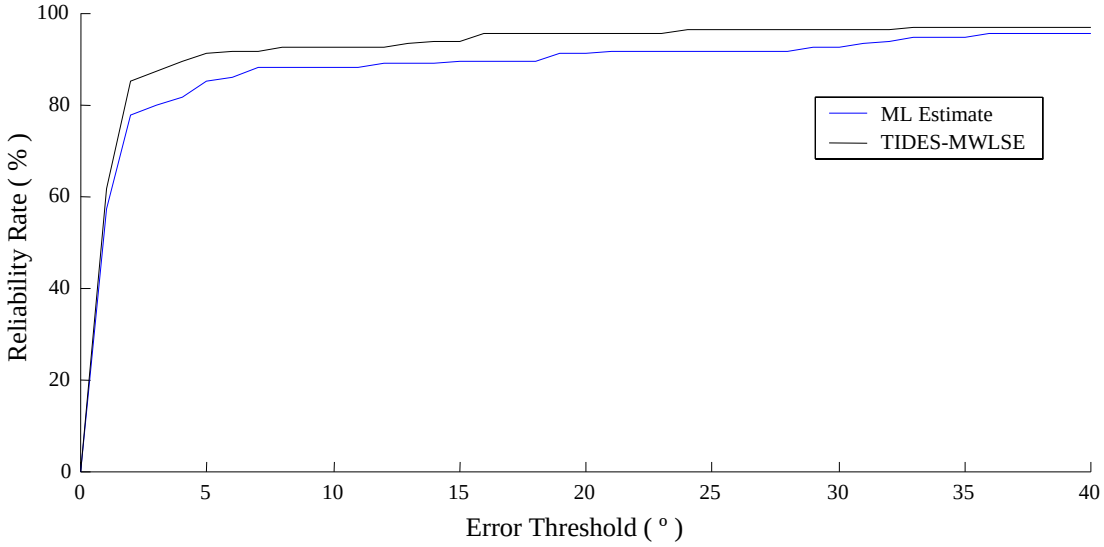

Figure 5.28 ***Reliability rates for DOA = 60 ° using TIDES-MWLSE.***

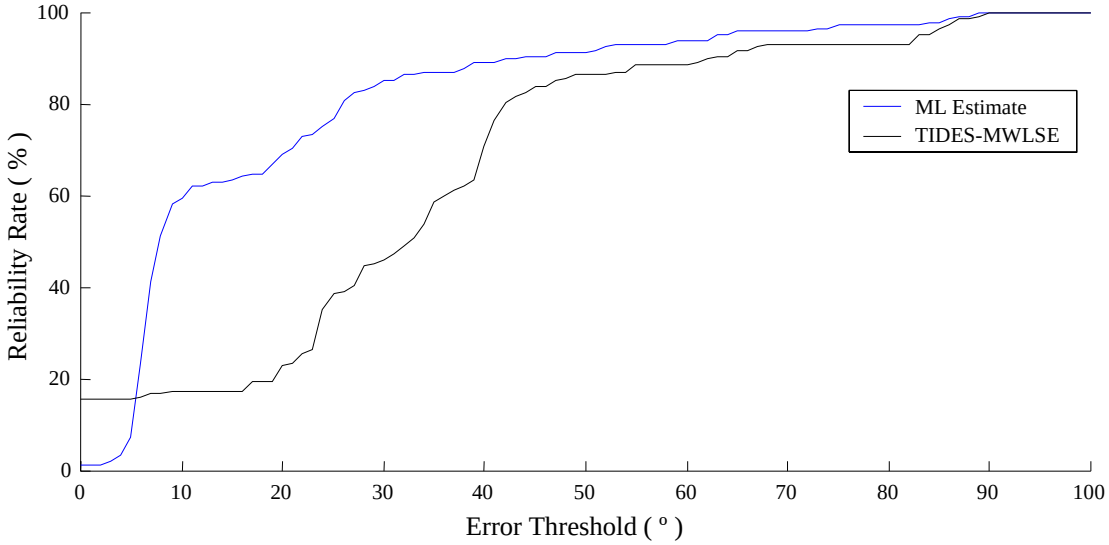

Figure 5.29 ***Reliability rates for DOA = 90 ° using TIDES-MWLSE.***

Now let us take a look at the results obtained from data recorded using our 7-element 3-dimensional array. Figure 5.30 and Figure 5.31 show the frame-wise estimates of the azimuth and elevation respectively. The reliability rates obtained for each of the methods are also shown separately for azimuth and elevation. In this case the frames were of length 256 samples, which at 8 *kHz* amounted to 32 *ms* of signal time for each frame. It can be observed that the TIDES-MWLSE algorithm does indeed give more reliable results for both the bearing estimates. The reliability rate curve for the elevation estimates shows an almost 10 % improvement in reliability for small errors (< 10°).

5.6.2. The MWTDS Criterion

In this method we try to select a set of time-delays that is closest in Euclidean distance to a statistical average of sets that were chosen for the previous frames. Let $\boldsymbol{\tau}_i$ be a candidate vector of time-delays and let $\boldsymbol{\tau}_{med,N}$ be the vector of time-delays that represents a median filtered set of time-delays from the previous N frames. Then the weighted distance between them is defined as

$$d_{w,i} = \frac{\left(\boldsymbol{\tau}_i - \boldsymbol{\tau}_{med,N}\right)^T \left(\boldsymbol{\tau}_i - \boldsymbol{\tau}_{med,N}\right)}{\left(\mathbf{s}_i^T \mathbf{s}_i\right)^k} \tag{5.7}$$

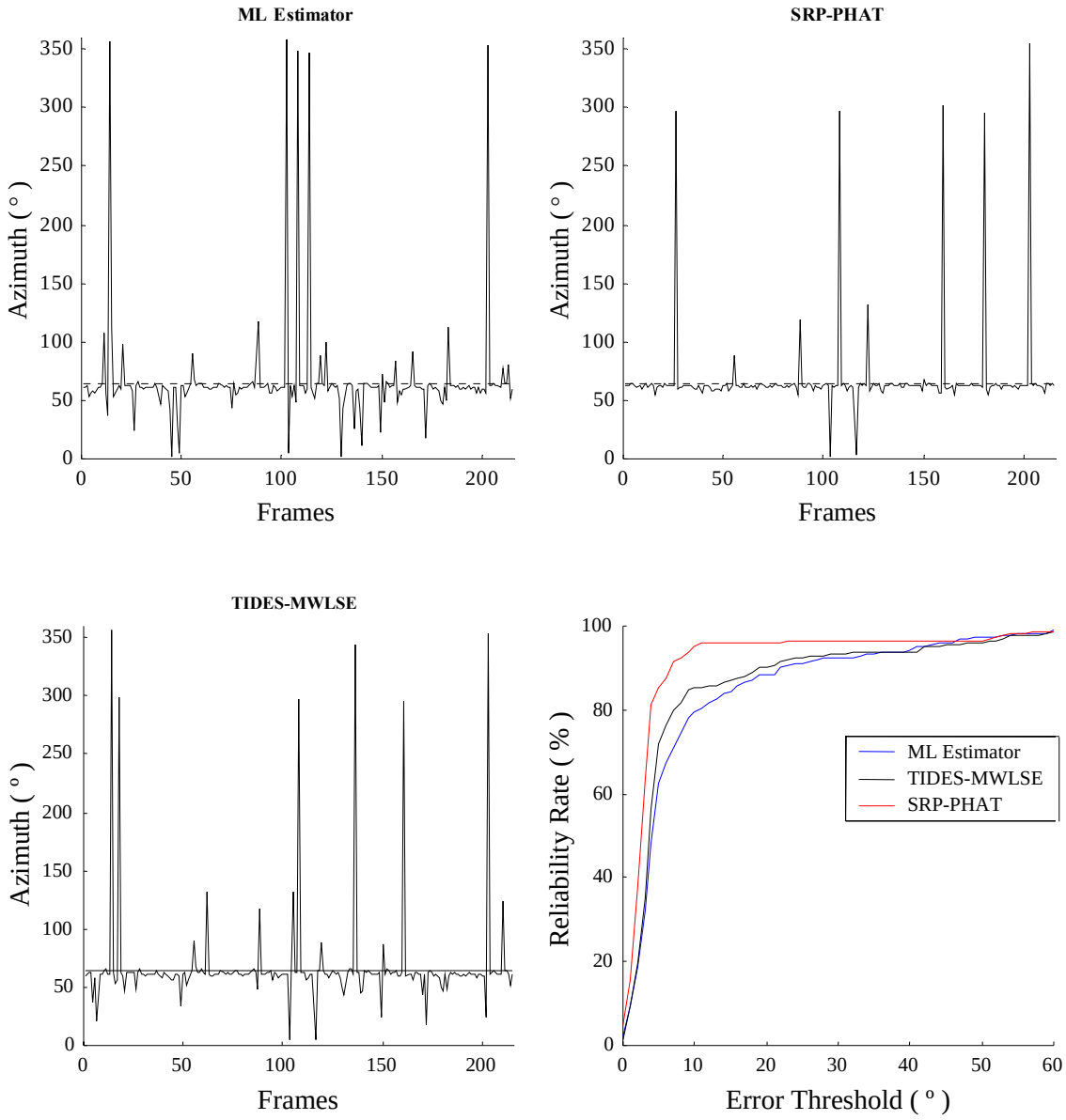

Figure 5.30 ***Frame-wise azimuth estimates and reliability-rate for TIDES-MWLSE compared with other methods.***

The numerator in (5.7) is just a squared distance between the two time-delay vectors. The denominator tries to further decrease the distance of those time-delay vectors that exhibit stronger peaks. Again the factor *k* determines how much the strength of the peaks should affect the weighted separation. The candidate set of time-delays that minimizes this weighted separation is chosen as the true time-delay and used in the computation of a least squares estimate of the direction of arrival. Figure 5.32 shows the frame-wise DOA estimates obtained from the same experimental data as before, i.e., for a 4-element linear array with a separation of

20 *cm* and the source at an angle of 60°. Again it can be seen that this technique prevents some of the impulsive errors that were present before.

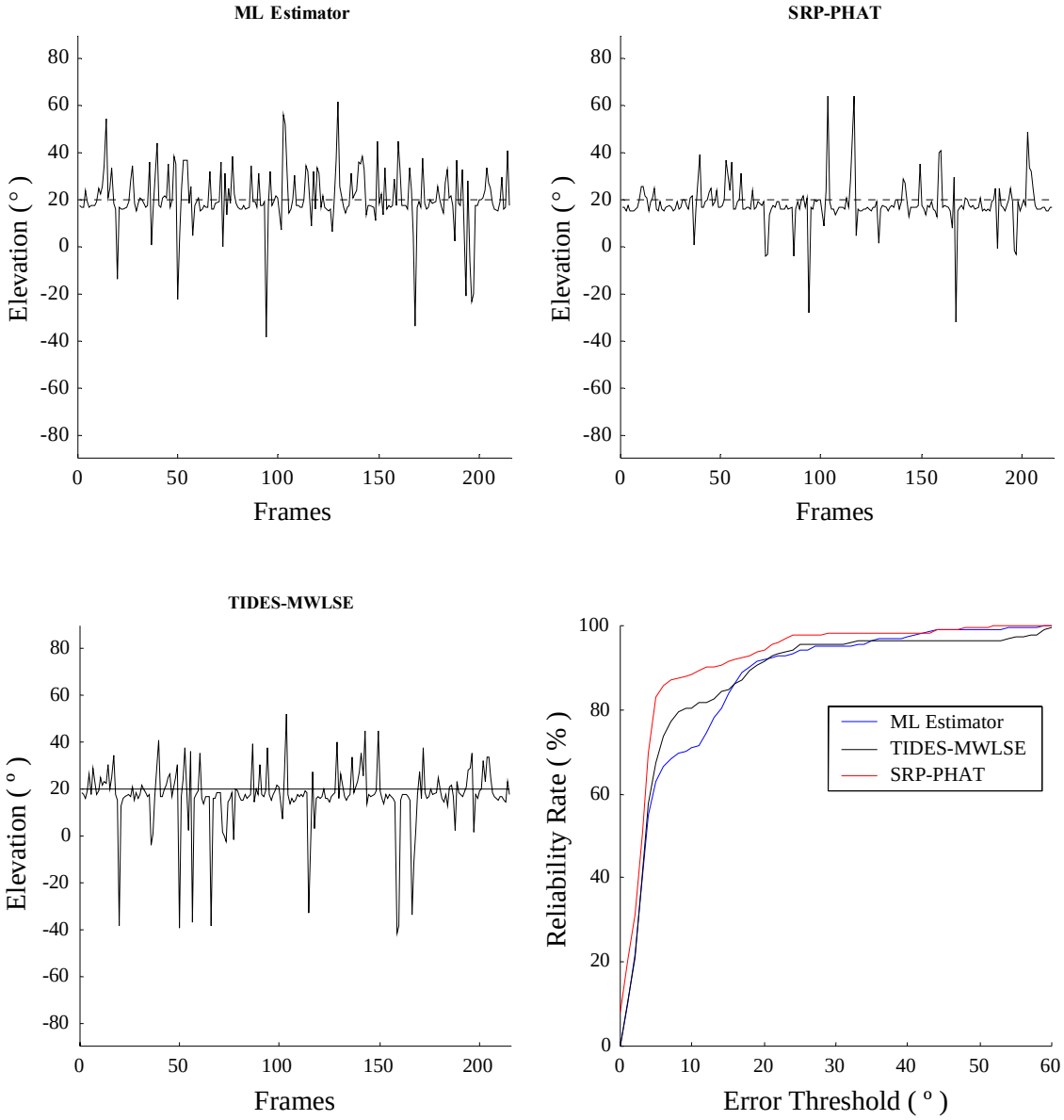

Figure 5.31 ***Framewise elevation estimates and reliability-rate for TIDES-MWLSE compared with other methods.***

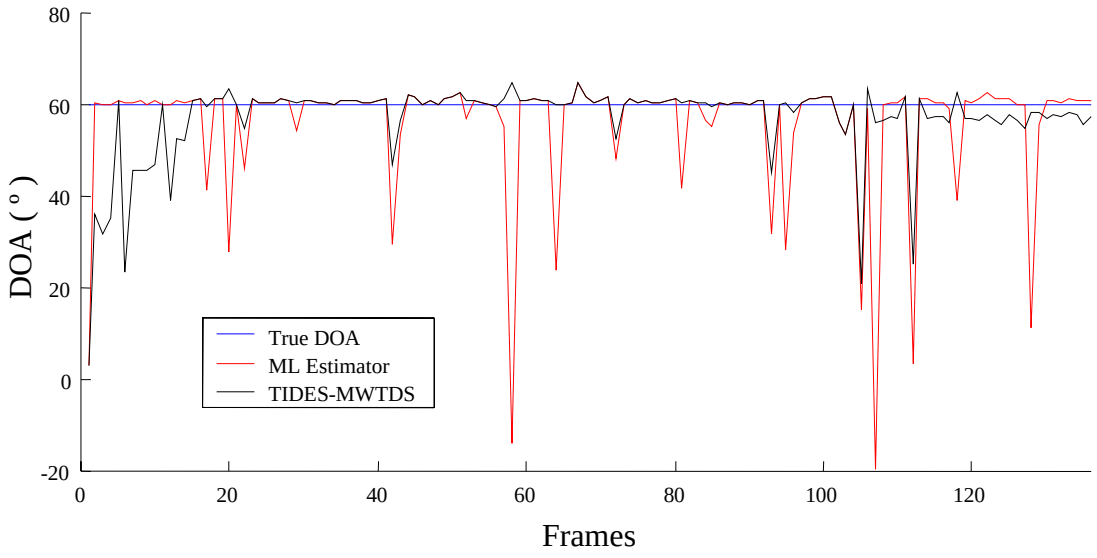

Figure 5.32 ***Framewise DOA estimates shows that the TIDES-MWTDS algorithm corrects many of the impulsive errors.***

Figure 5.33, Figure 5.34 and Figure 5.35 show the reliability rates obtained from the TIDES-MWTDS algorithm. It is observed that the algorithm improves the reliability of estimates if the DOA is not too far away from broadside.

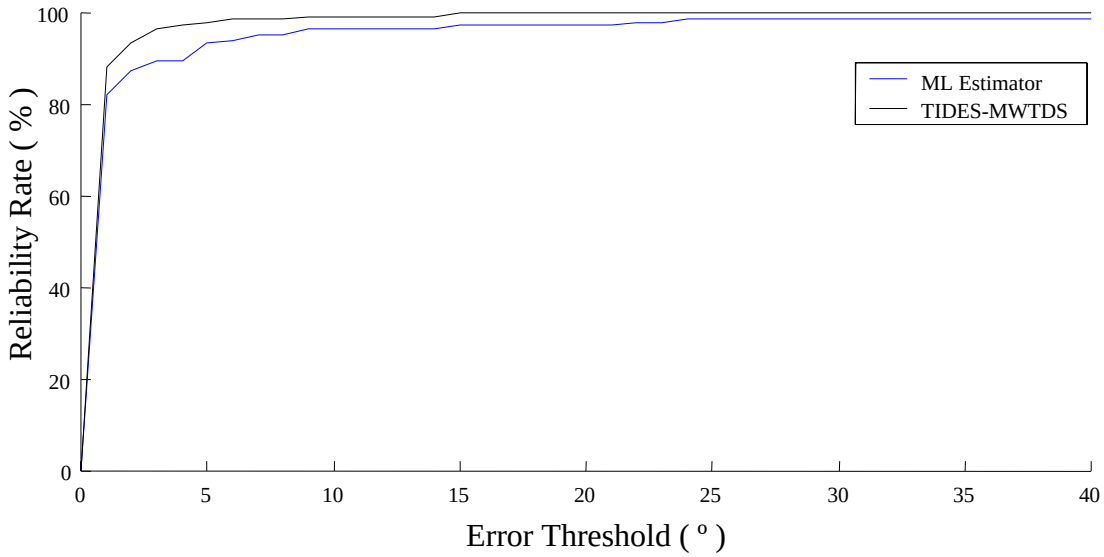

Figure 5.33 ***Reliability rates for DOA = 30° using TIDES-MWTDS.***

In the reliability rate curve for DOA = 60° we observe that the TIDES-MWTDS actually increases the probability of small errors. This relates to the portion of the curve in Figure 5.32

beyond frame number 100 where the estimates seem to have an offset. The advantage in using this algorithm is that large impulsive errors are reduced to a great extent.

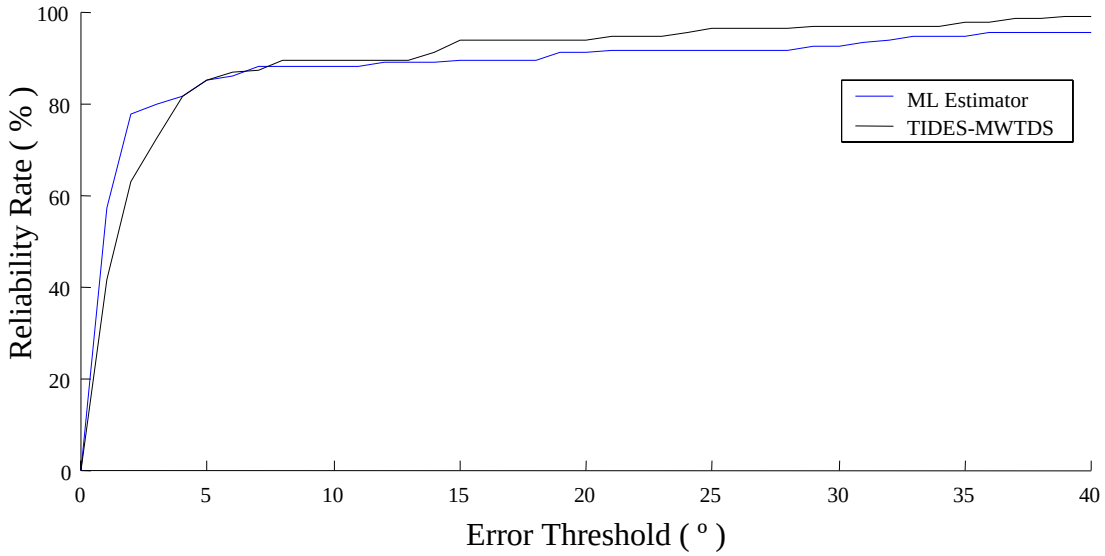

Figure 5.34 ***Reliability rates for DOA = 60° using TIDES-MWTDS.***

For the case where DOA = 90°, the algorithm actually degrades the reliability of the estimation. The source of this degradation can be seen in Figure 5.36 which shows the framewise DOA estimates for DOA = 90°. When one of the DOA estimates has an error in it, the algorithm struggles to get back to the correct estimate because it tries to find an estimate that is close to the previous one. This is especially true for angles far away from broadside because in such cases there is a higher probability of multiple strong peaks being present in the GCC function. For all these experimental cases the algorithm was made to select up to 3 time-delays for each pair of microphones to generate the candidate sets. Also the value of k used was 2 and the length of the median filter, N, was set to 5.

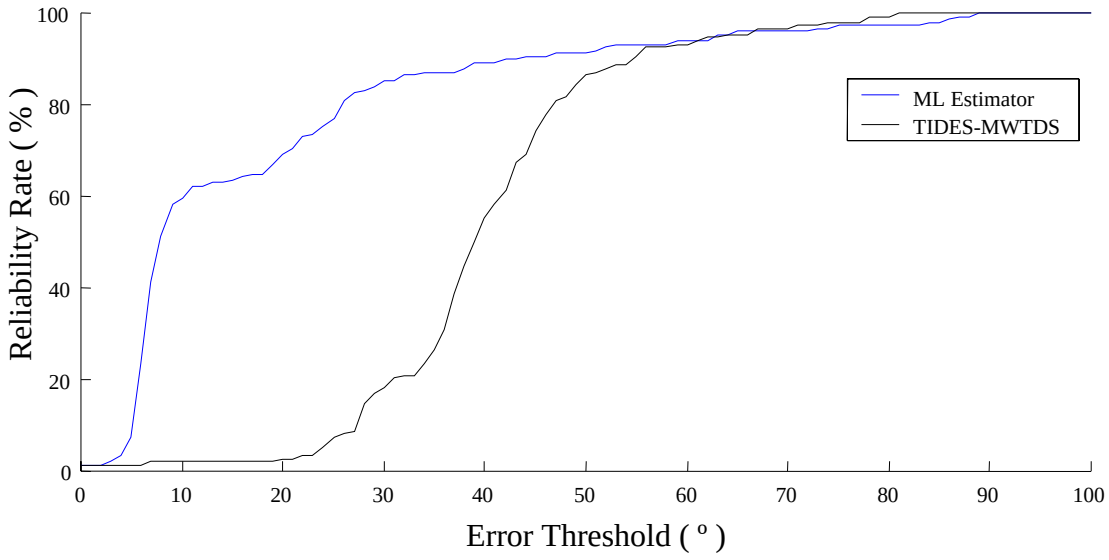

Figure 5.35 ***Reliability rates for DOA = 90° using TIDES-MWTDS.***

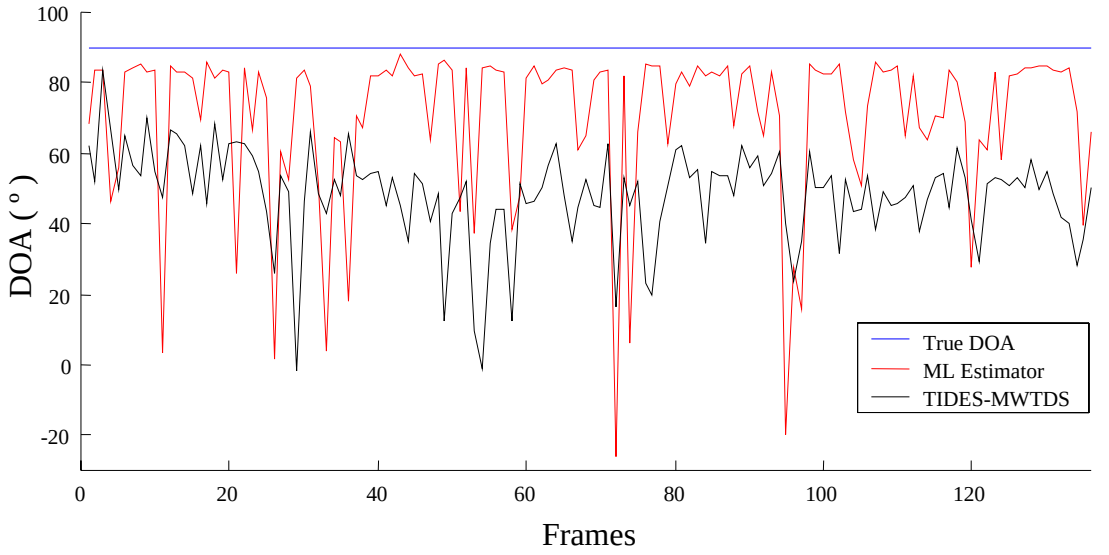

Figure 5.36 ***Framewise DOA estimates for DOA = 90°.***

Again we study the performance of the TIDES-MWTDS algorithm on data recorded using our 3-dimensional array. Figure 5.37 shows a clear improvement in the azimuth estimates. The same is the case for elevation estimates shown in Figure 5.38. The computations were done with k set to 2 and N set to 5. Also since we have 9 microphone pairs to consider, we looked for a maximum of 2 time delays for each pair so that the number of combinations would not be too large to offset the computational advantage that the algorithm has over the SRP-PHAT algorithm. The improved performance of this algorithm is also observed in the reliability rate

curves for the azimuth and elevation, which again show an approximately 10% improvement in reliability for small errors (< 10°).

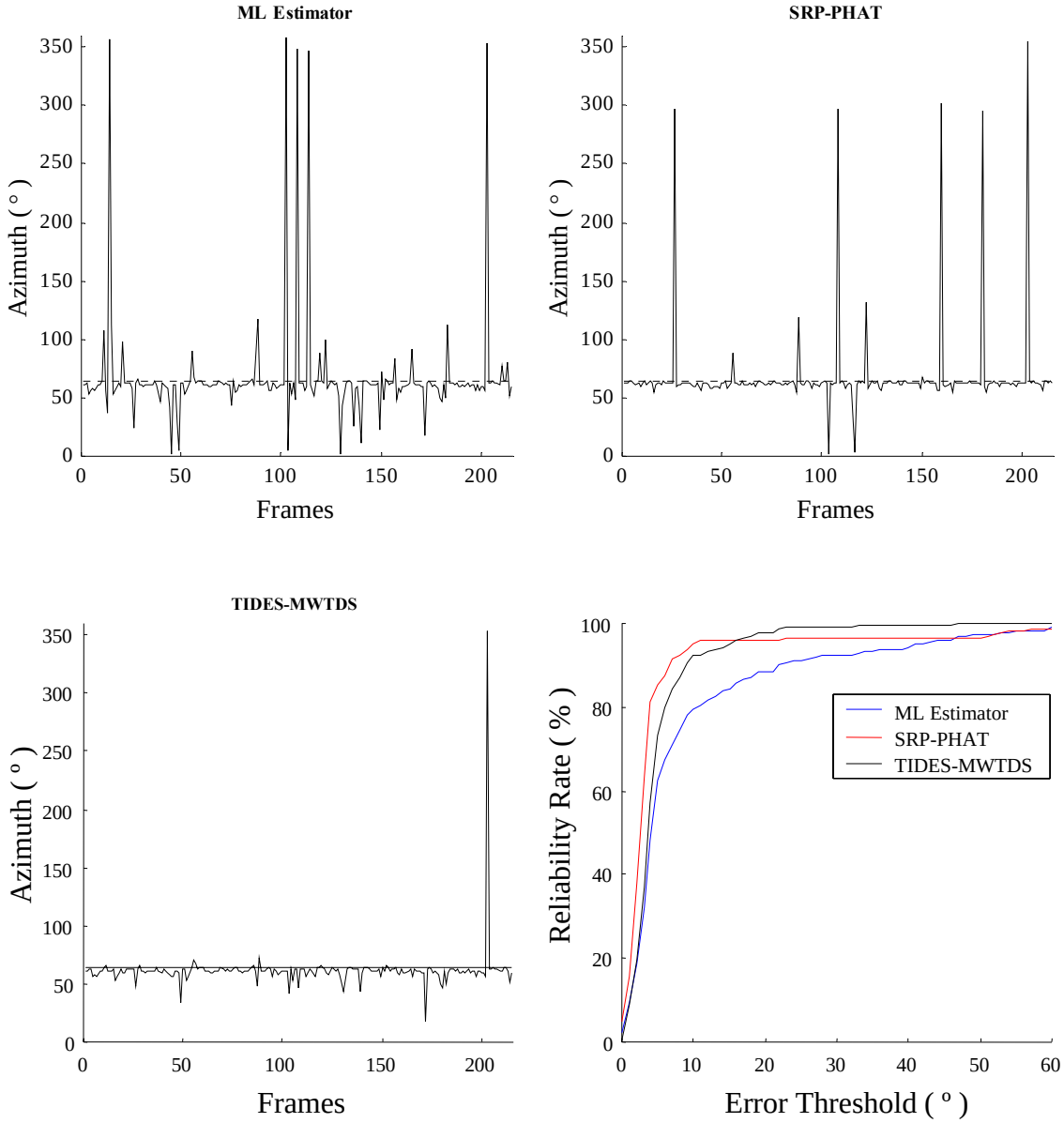

Figure 5.37 ***Framewise azimuth estimates and reliability-rate for TIDES-MWTDS compared with other methods.***

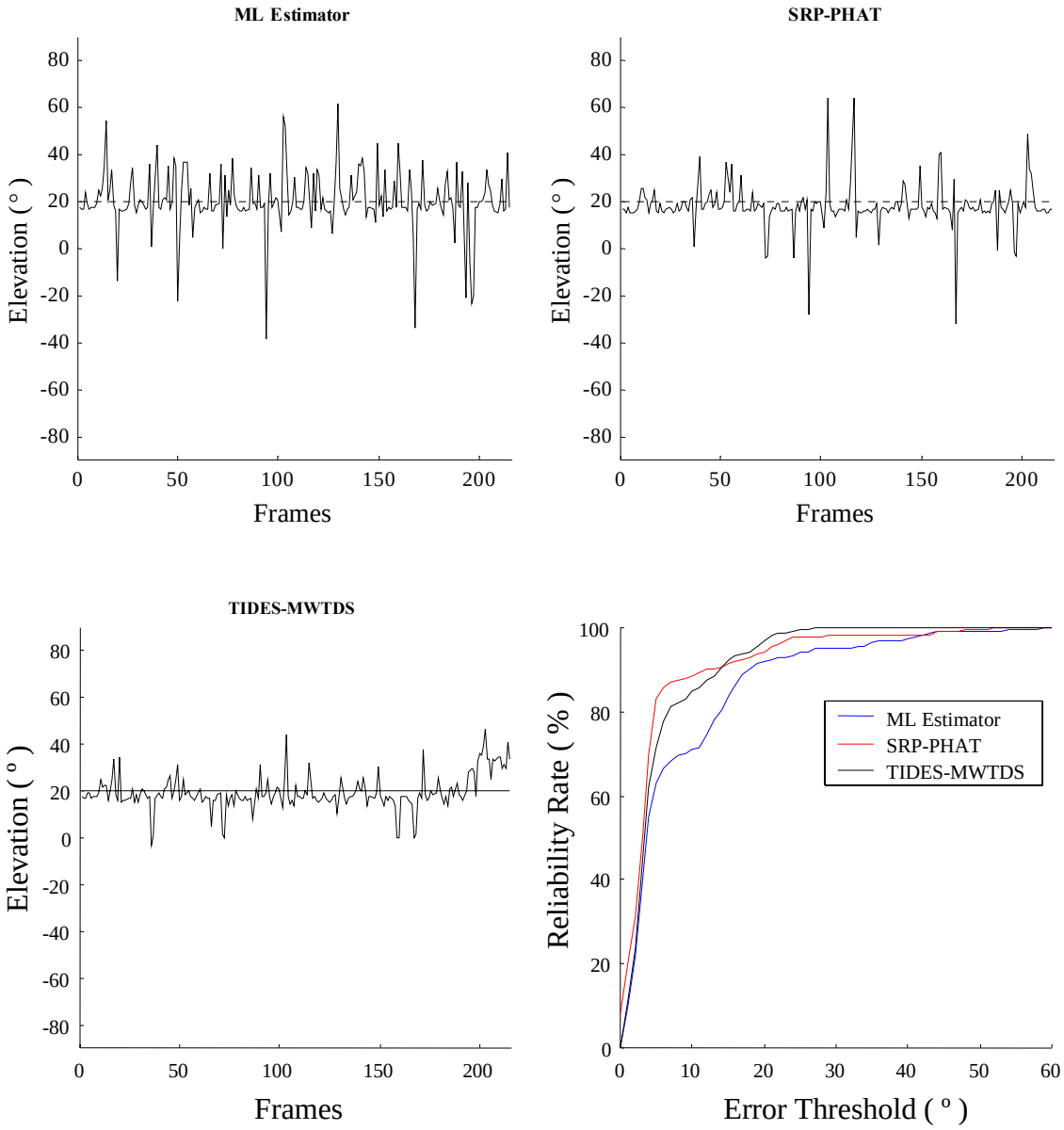

Figure 5.38 ***Framewise elevation estimates and reliability-rate for TIDES-MWTDS compared with other methods.***

5.7. Comprehensive Simulation Results

Comprehensive simulations were performed using both versions of the TIDES algorithm to study its strengths and weaknesses. Two of these results that illustrate the important features of the algorithmic performance are shown in this section. The first simulation was performed for the case of moderate signal to reverberation ratio (SRR). The room dimensions were set to $5m \times 5m \times 5m$ and the reverberation time of the room was set to 200 *ms*. A seven-element 3D

microphone array with three microphones in a straight line in each dimension was used to capture the signal. The middle microphone in each dimension was common. Microphones in each dimension were paired with the other two microphones in the same dimension and this gave rise to 3 pairs in each dimension and 9 pairs total. The source was placed 1.5 *m* away from the array at an azimuth of 43° and an elevation of 25°. Figure 5.39 shows the frame-wise azimuth estimates and Figure 5.40 shows the reliability rates of the azimuth estimates.

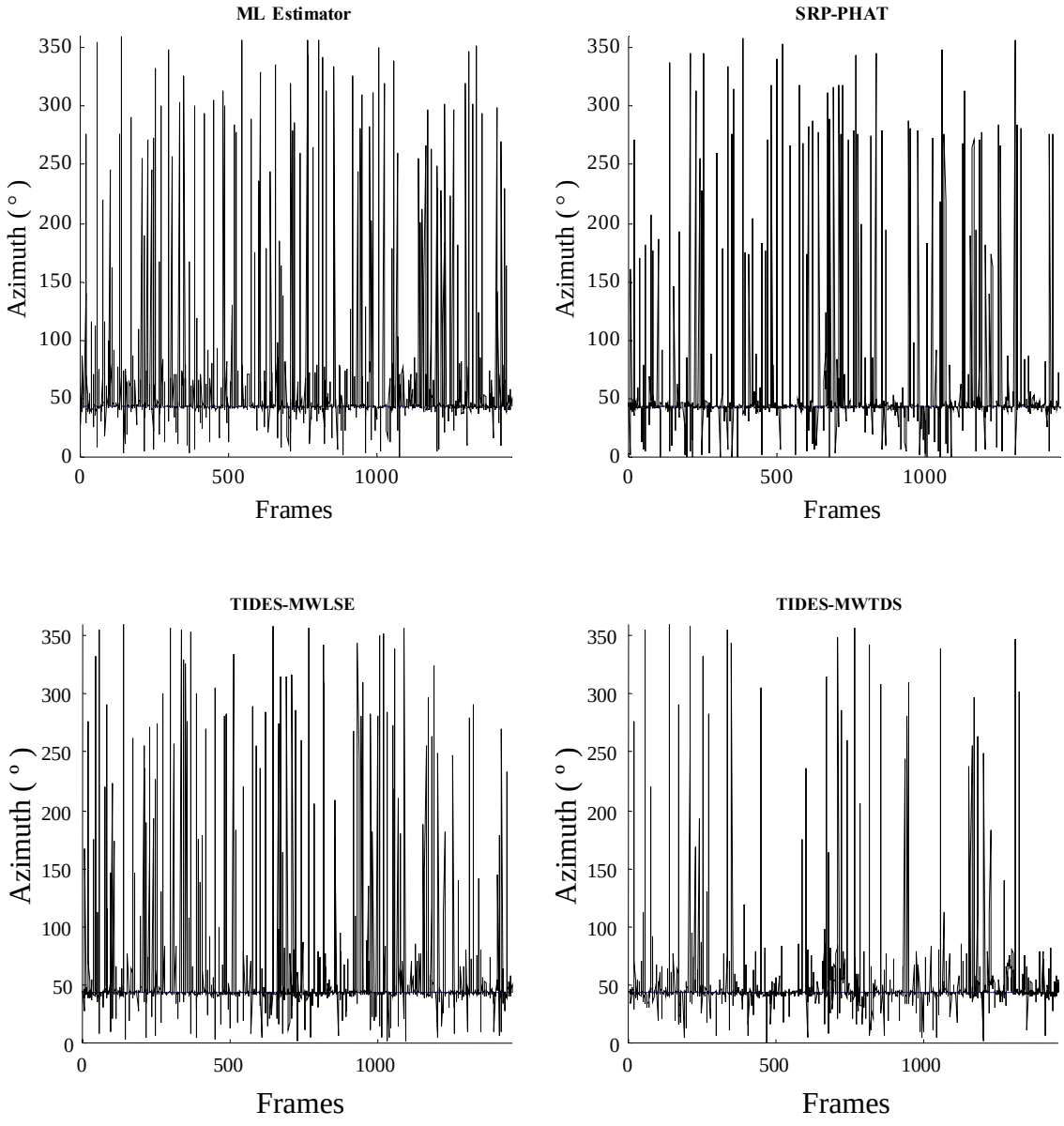

Figure 5.39 ***Azimuth Estimates using the four methods with the source separated from the array by 1.5 m and room reverberation time = 200 ms.***

The frame-wise azimuth estimates for TIDES-MWTDS clearly show a lowering of large errors. This is reflected in the reliability rate curve for TIDES-MWTDS, which crosses the curve for TIDES-MWLSE at error threshold of approximately 12° and crosses the curve for SRP-PHAT at approximately 22°. This means that the SRP-PHAT has greater probability of giving azimuth errors in excess of 22° than the TIDES-MWTDS algorithm. Both proposed algorithms clearly perform better than the algorithm based on ML-TDE.

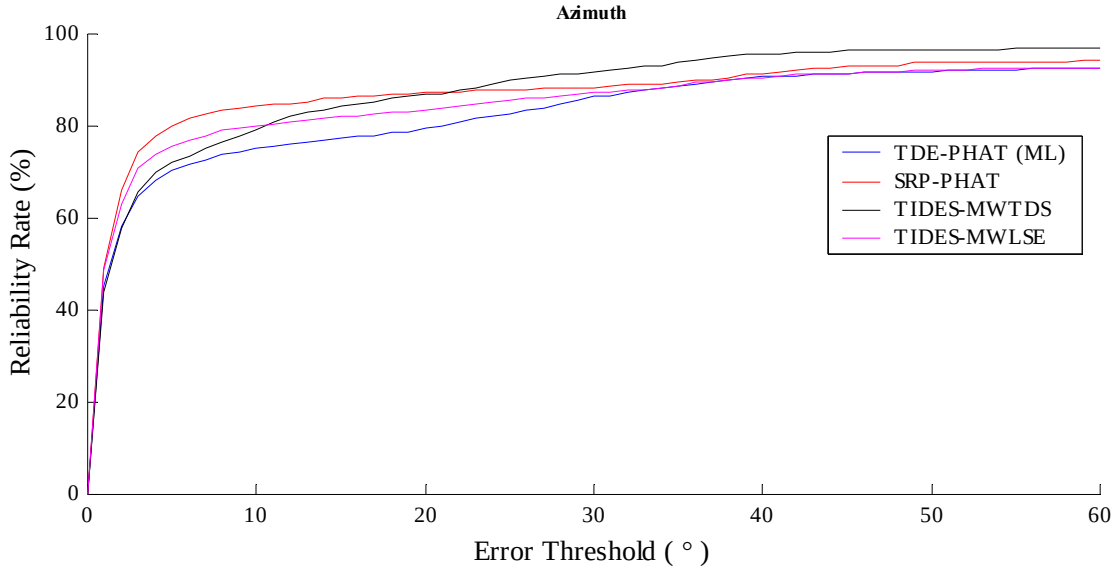

Figure 5.40 ***Reliability rates of the azimuth estimates using the four methods with the source separated from the arrays by 1.5 m and room reverberation time = 200 ms.***

A similar result is observed for the frame-wise elevation estimates shown in Figure 5.41 for all four methods and the reliability rates shown in Figure 5.42. In this case the curve for TIDES-MWTDS crosses over the one for TIDES-MWLSE at approximately 5° and the one for SRP-PHAT at approximately 7°. Hence the SRP-PHAT has a greater probability of giving elevation errors in excess of 7° than the TIDES-MWTDS algorithm. Figure 5.43 combines the reliability rates of the azimuth and elevation to give a single performance index for the performance of the algorithm. Here the error-threshold is a root mean square of the errors in azimuth and elevation.

$$e_t = \sqrt{\frac{e_{az}^2 + e_{el}^2}{2}} \tag{5.8}$$

This combined error is computed for each of the framewise estimates and the percentage of estimates that have the combined error less than the threshold is plotted against the threshold. This gives us an approximate measure of the error in the estimated unit-vector along the direction of the source. We observe that TIDES-MWTDS performs best for large combined errors.

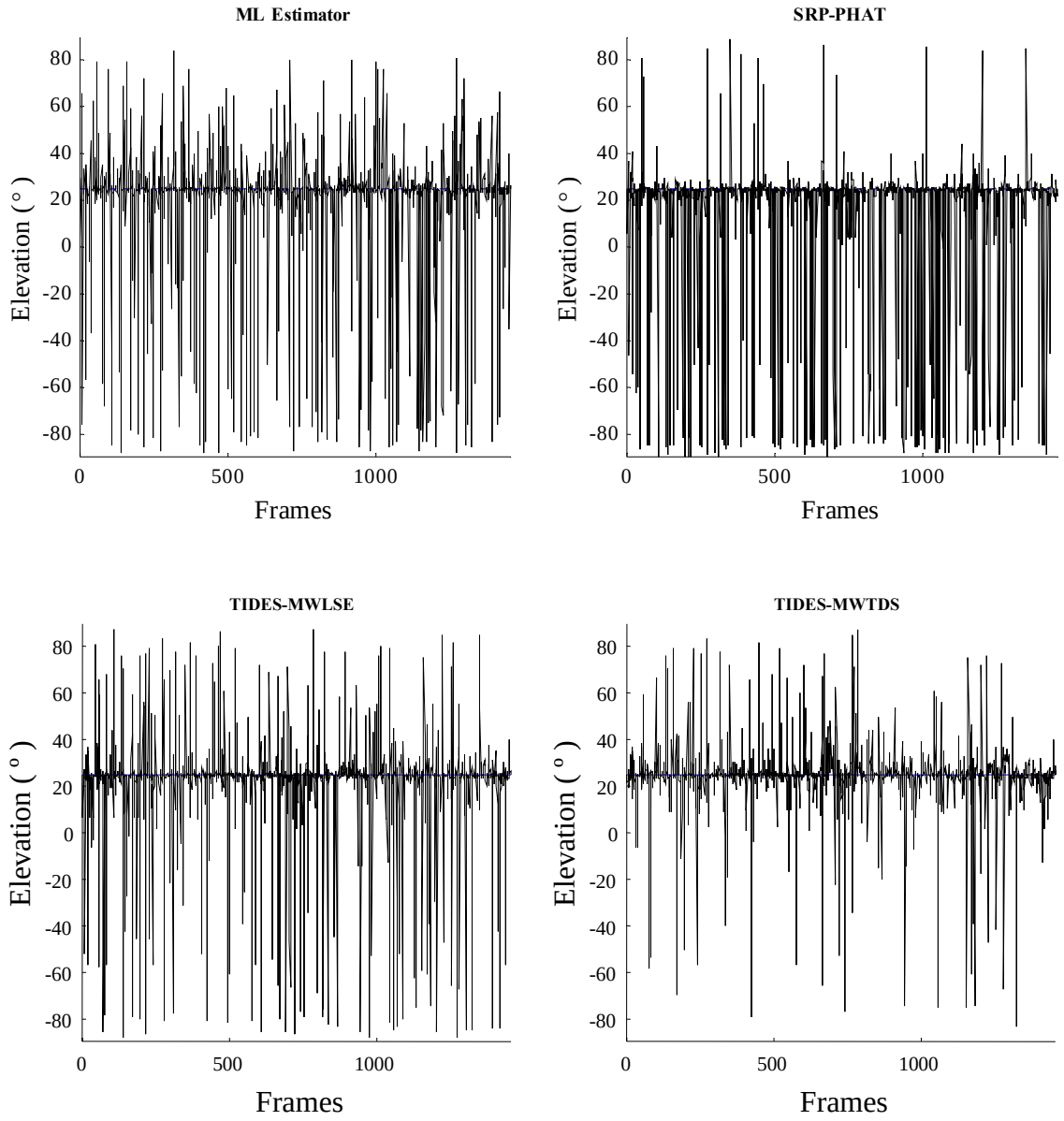

Figure 5.41 ***Elevation estimates with the four methods with the source separated from the source by 1.5 m and room reverberation time = 200 ms.***

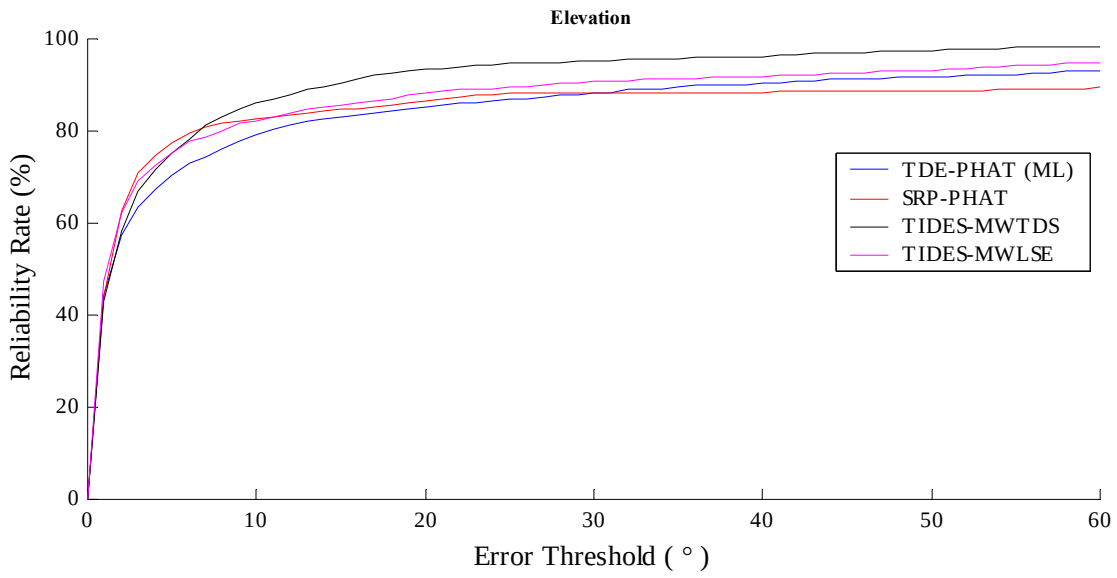

Figure 5.42 ***Reliability rates of the elevation estimates using the four methods with the source separated from the array by 1.5 m and room reverberation time = 200 ms.***

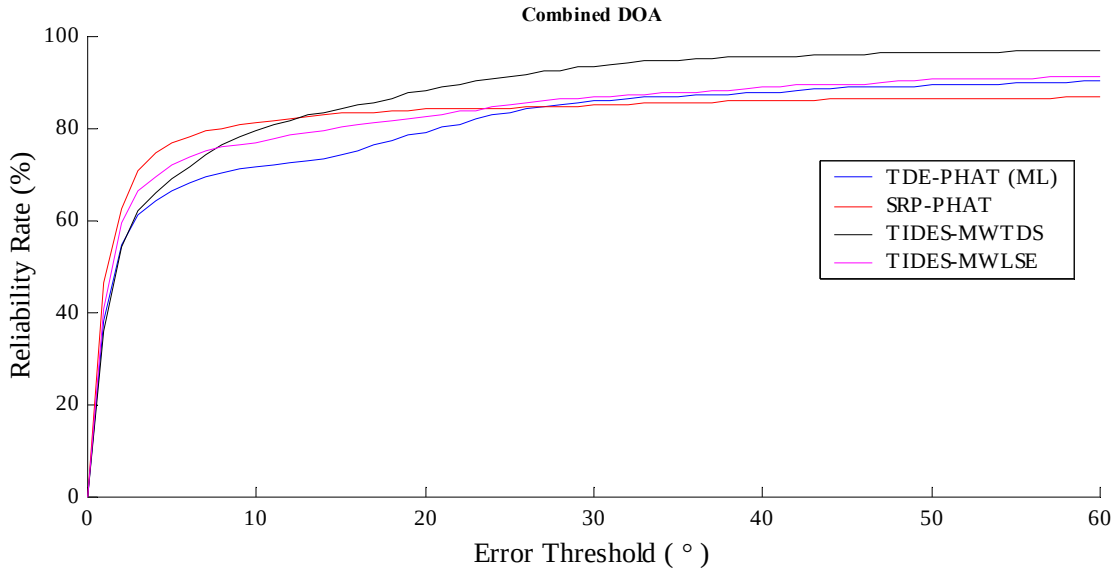

Figure 5.43 ***Reliability rates using combined errors from azimuth and elevation with the source separated from the array by 1.5 m and room reverberation time = 200 ms.***

To study the performance of the TIDES algorithm at low SRR, we changed the setting a little bit. This time the room was changed to one of size $3m \times 3m \times 3m$ with a reverberation time of 100 *ms*. To increase the reverberation power relative to the signal power, we placed the array in one end of the room and the source in the other end of the room. This resulted in a distance of 3.6 *m* separating them. The true azimuth was 26.57° and the true elevation was 41.81°.

The frame-wise azimuth estimates for all four methods are shown in Figure 5.44 and the resultant reliability rates in Figure 5.45. Also the frame-wise elevation estimates for all four methods are shown in Figure 5.46 and the resultant reliability rates in Figure 5.47. The combined azimuth and elevation performance of the algorithms is shown in Figure 5.48.

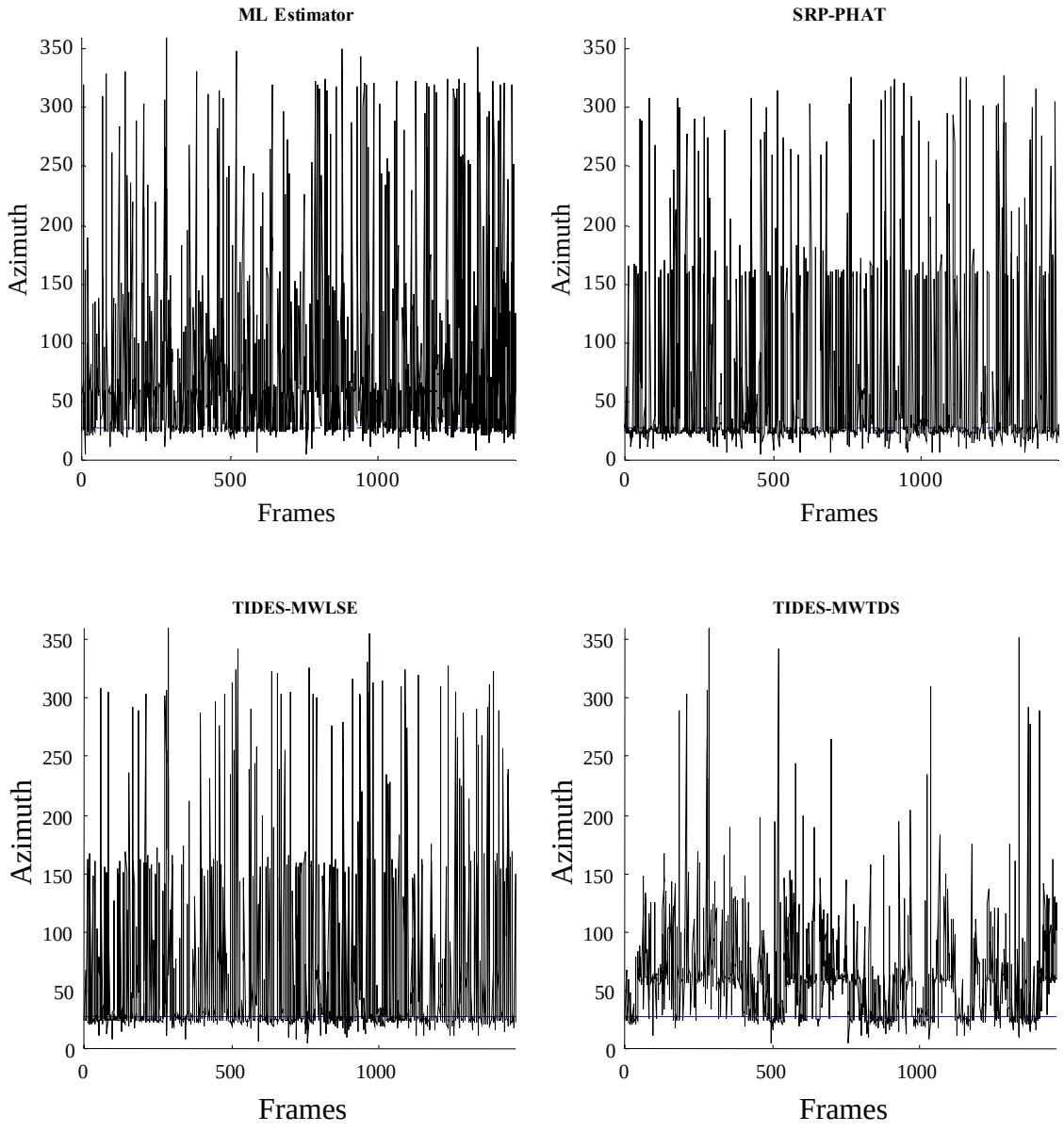

Figure 5.44 ***Azimuth Estimates using the four methods with the source separated from the array by 3.6 m and room reverberation time = 100 ms.***

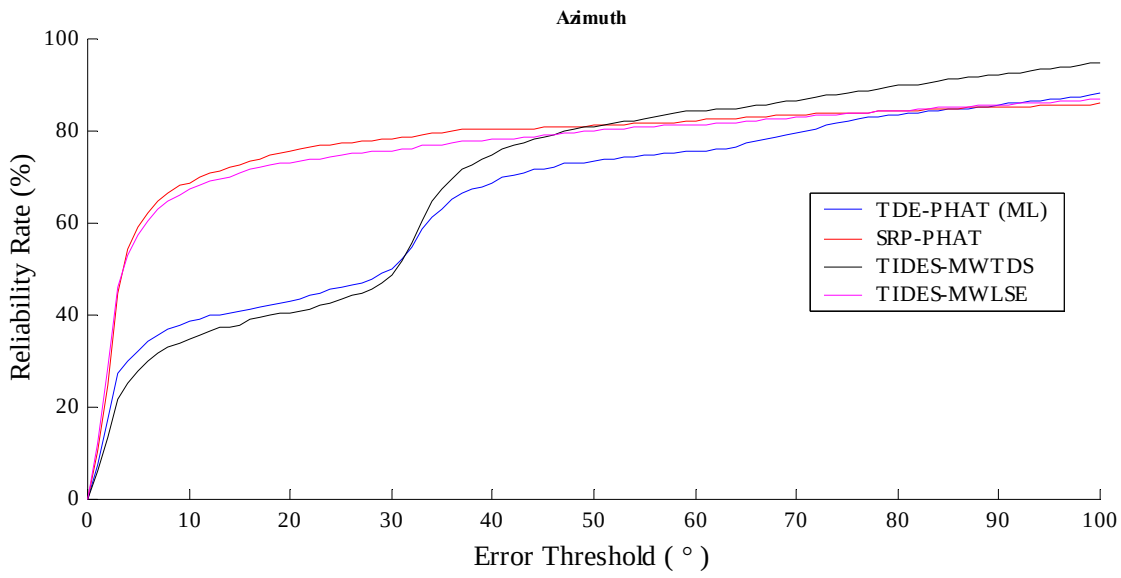

Figure 5.45 ***Reliability rates of the azimuth estimates using the four methods with the source separated from the arrays by 3.6 m and room reverberation time = 100 ms.***

The reliability rate curves for the azimuth show that the TIDES-MWTDS algorithm has broken down in this case. It shows bad performance for small error thresholds and catches up with the other methods only at approximately 50°. The reason for this can be observed from the frame-wise estimates where the TIDES-MWTDS method struggles to get back to good estimates once it gets stuck into a bad estimate. This is a consequence of using information from the previous frames. The estimates from all the erroneous frames are more or less the same wrong azimuth. This is because the GCC-PHAT between microphone-pairs is consistently similar in all these frames and thus the time-delay candidate sets are also consistently similar. Since the TIDES-MWTDS algorithm tries to minimize the change in time-delay estimates between frames it consistently selects the same wrong time-delay set from among the candidates. This results in the algorithm consistently estimating the same wrong azimuth. On the other hand the TIDES-MWLSE algorithm shows very good performance in azimuth estimation since it does not use information from previous frames. Its performance is comparable to SRP-PHAT.

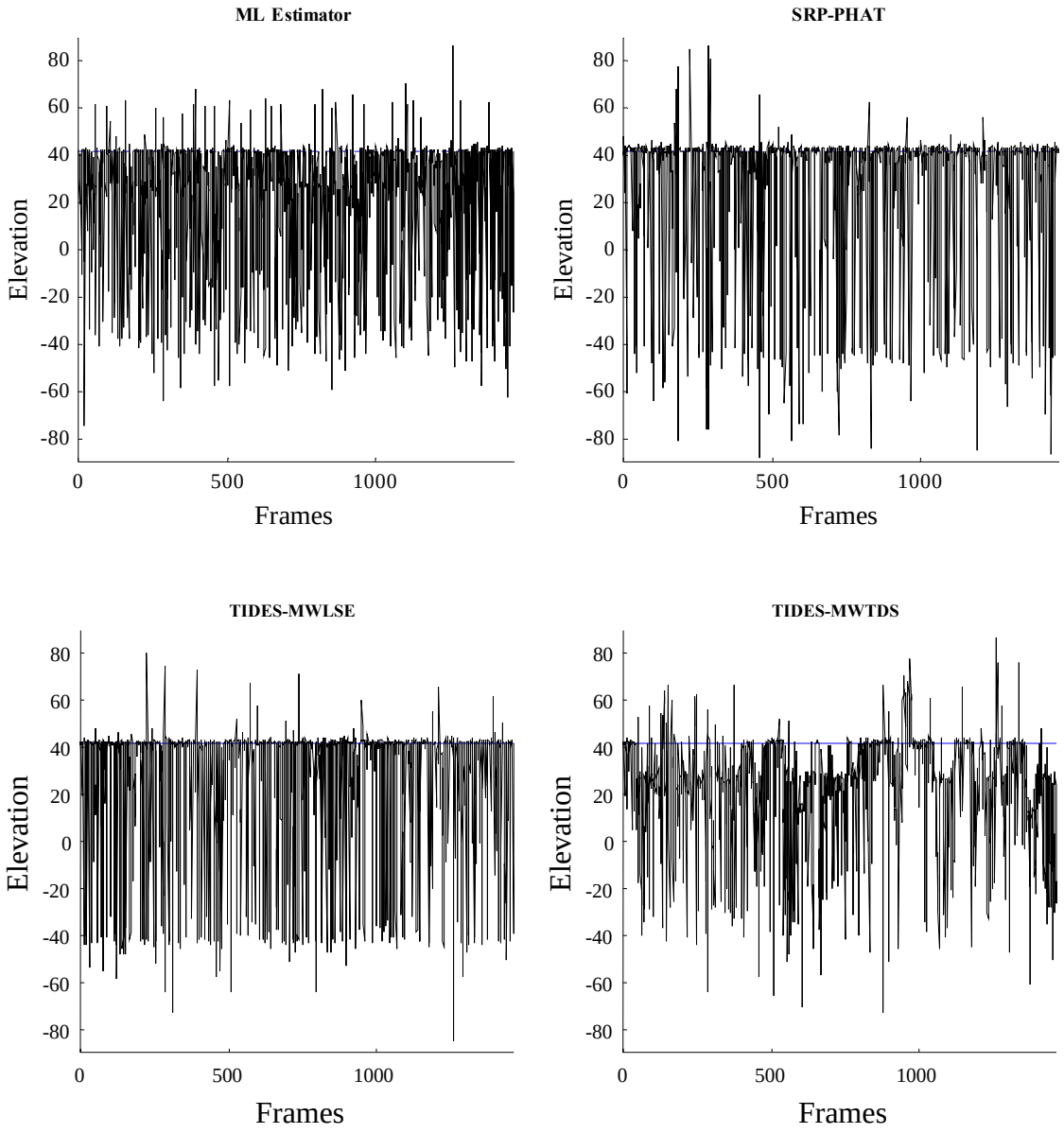

Figure 5.46 ***Elevation estimates with the four methods with the source separated from the source by 3.6 m and room reverberation time = 100 ms.***

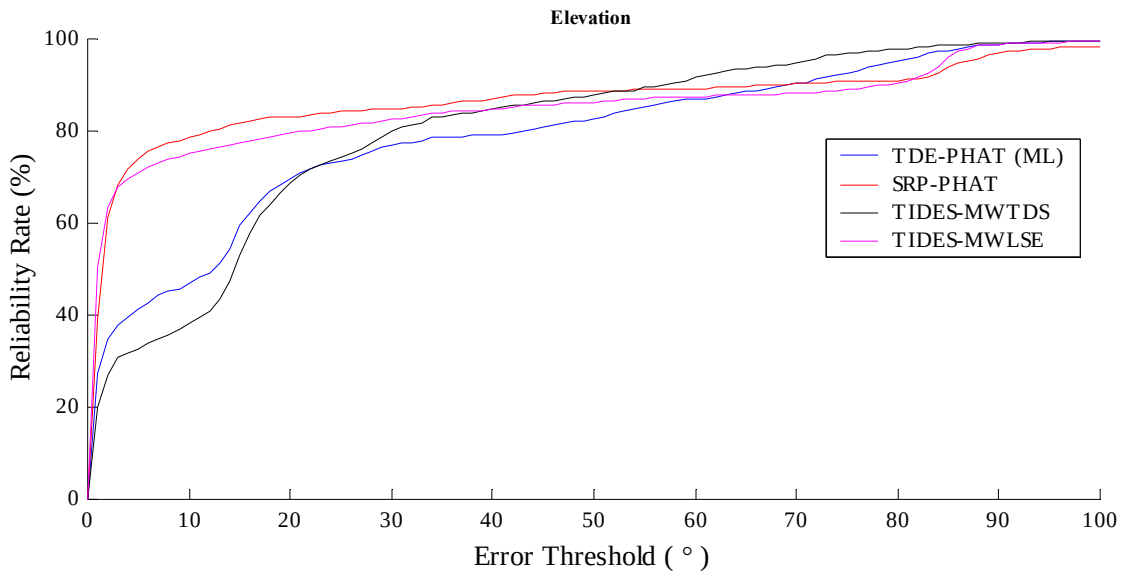

Figure 5.47 ***Reliability rates of the elevation estimates using the four methods with the source separated from the array by 3.6 m and room reverberation time = 100 ms.***

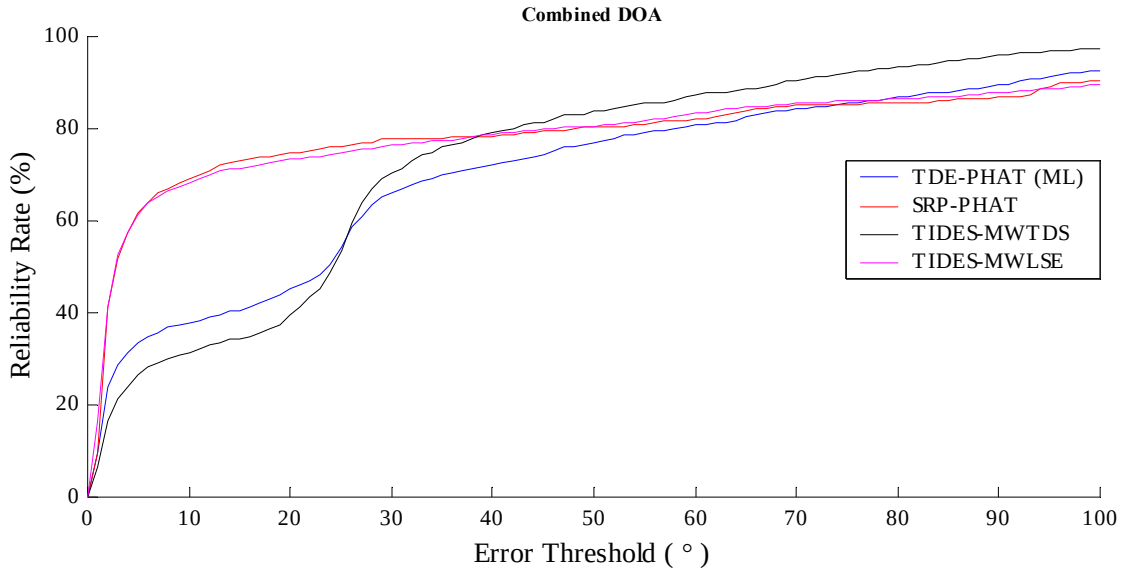

Figure 5.48 ***Reliability rates using combined errors from azimuth and elevation with the source separated from the array by 3.6 m and room reverberation = 100 ms.***

In the case of the elevation estimates the TIDES-MWTDS algorithm also performs the poorest for error thresholds less than 20°. Again this is due to the fact that the algorithm tends to get into a wrong estimate and tends not to be able to recover from it. This is again clearly visible in the frame-wise elevation estimates using the TIDES-MWTDS. For many consecutive frames the estimated elevation was closer to 25° rather than the actual 41.81°. The algorithm catches up

with the other algorithms at error thresholds approximately equal to 50°. Again we notice that for elevation the TIDES-MWLSE method performs very well in low SRR situations. The combined reliability rates of each of the algorithms possess the general characteristics of the individual azimuth and elevation reliability rates.

Next a simulation was run under low SRR to determine whether the GCC-PHAT between microphone pairs do indeed posses the information required to estimate the correct DOA. The results from this simulation are shown in Figure 5.49. The reliability rates of azimuth estimates for four of the methods are shown in Figure 5.50. TDE-PHAT performed the worst and its results are not shown here.

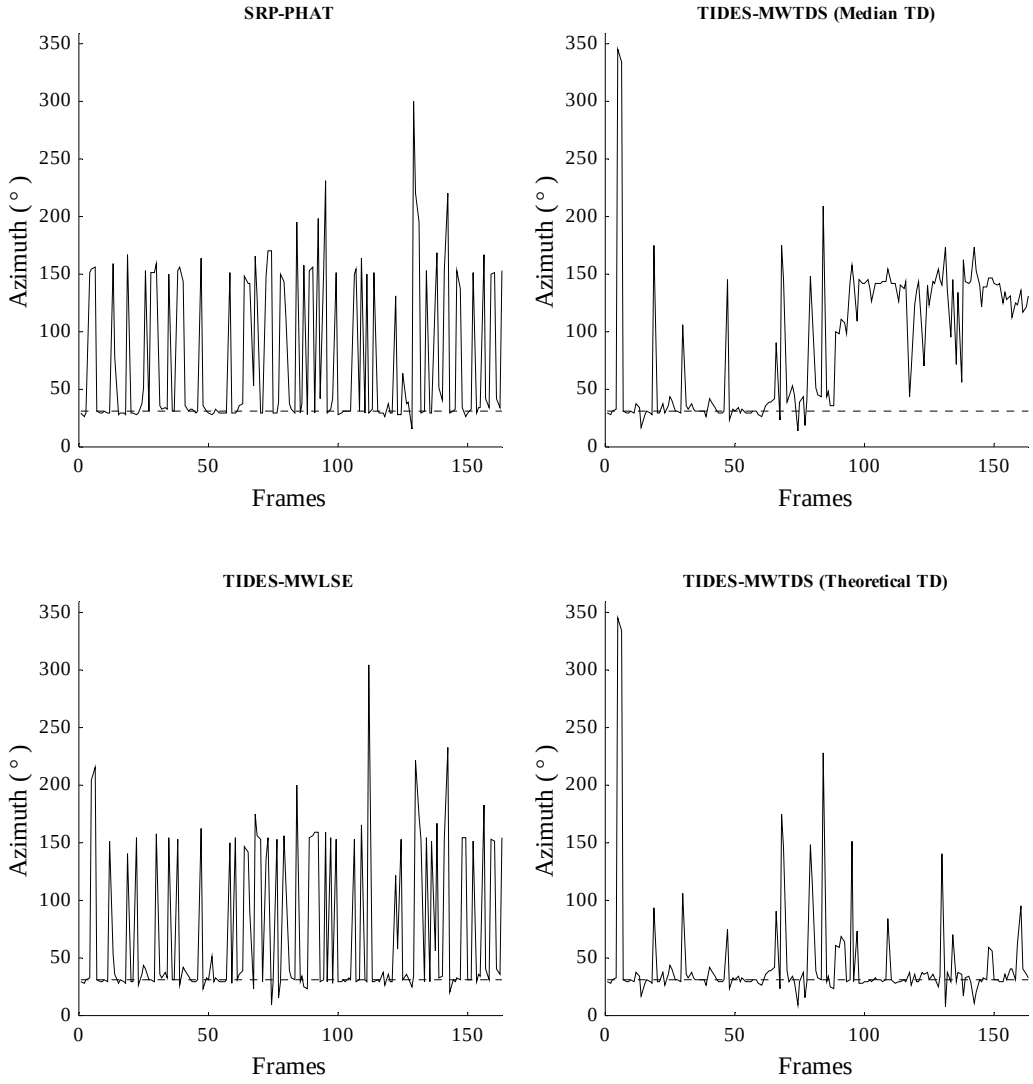

Figure 5.49 ***Framewise azimuth estimates under severe SRR conditions showing that improvement in performance is possible using better time-delay selection criteria.***

In this simulation, a 7-element microphone array was placed at one corner of a room of dimensions $3\,m \times 3\,m \times 3\,m$. The position of the microphone-array was $\begin{bmatrix} 0.2 & 0.2 & 0.2 \end{bmatrix}^T$ in *meters*. The speech source was place at another end of the room at $\begin{bmatrix} 1.7 & 2.7 & 2.7 \end{bmatrix}^T$, again in *meters*. This resulted in a distance of 3.84 *m* between the source and the microphone array. The true azimuth and elevation for these array and source positions were 30.96° and 40.61° respectively. Array signals were simulated at a room-reverberation time of 100 *ms*. DOA estimation was performed from 32 *ms* frames using all the algorithms discussed in this chapter. Additionally, the TIDES-MWTDS was run using the theoretical time-delays instead of the median filtered time-delays. Also, for this run, the parameter *k* was set to zero so that the relative-strength based weighting is not performed. This means that for this run, the algorithm is forced to select a set of time-delay estimates that is closest to the theoretical time-delays.

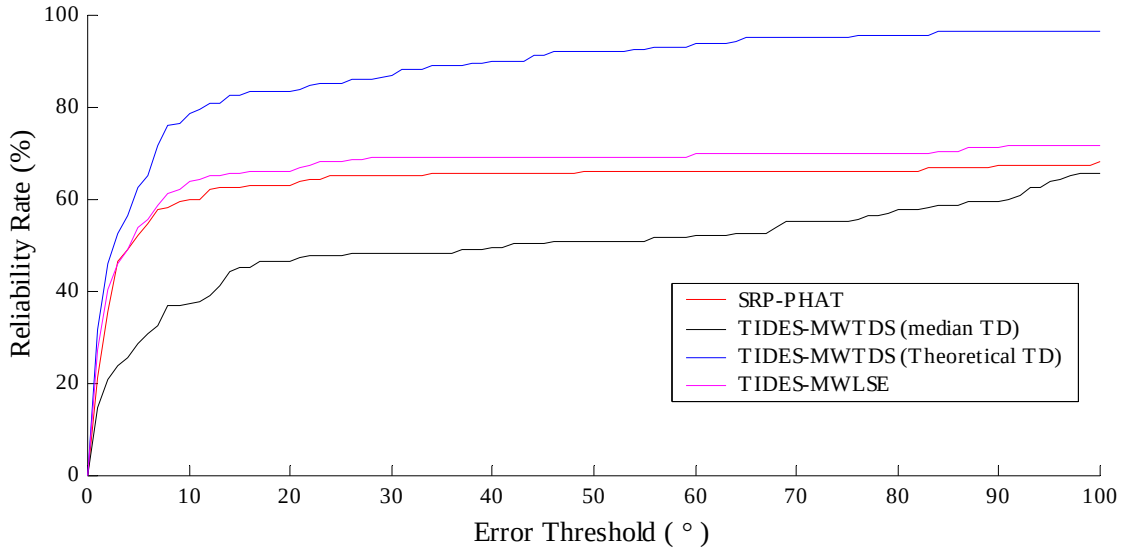

Figure 5.50 ***Relaibility rates for the four methods showing the potential for improvement with better time-delay selection criteria.***

The figures show that under severe SRR conditions even the SRP-PHAT algorithm becomes very unreliable. The framewise estimates are seen to jump around a lot pointing to a large variance in the estimates. The reliability rate for SRP-PHAT shows that only approximately 55% of the estimates have an error less than 10°. The results for TIDES-MWLSE are very similar. However, the reliability rates show that TIDES-MWLSE performs marginally better than SRP-PHAT. The TIDES-MWTDS algorithm again breaks down because it gets stuck

in a wrong DOA estimate. This result is also very similar to the results in Figure 5.44 and Figure 5.46. However, we observe that if we had used the theoretical time-delays to select from a set of candidate time-delays we get much better performance. This implies that even under severe SRR conditions there exist peaks at the correct time-delays. But we do not have a criterion that selects those correct time-delays from incorrect ones. Thus it would be worthwhile to pursue other more reliable criteria to perform time delay selection.

6. Conclusions and Future Work

A new time delay estimate based direction of arrival estimation algorithm is proposed in this thesis. The algorithm is aimed for use in automatic camera steering applications and high quality speech capture using microphone arrays in moderate signal to reverberation ratio environments. Such environments are commonly found in moderately sized rooms when the source is less than 2 *m* from the microphone array. We would like to obtain robust and reliable DOA estimates with as little data as possible (32 – 64 *ms*) in order to perform fast tracking of the source and to reduce computation. Time delay estimate based algorithms are the most popular algorithms in use today because of the relative simplicity of computation when compared to more computationally expensive methods like SRP-PHAT.

The proposed algorithm is a two-stage process. In the first stage time-delays between several microphone pairs are computed using the generalized cross-correlation of the pairs of signals. The pre-filter used in the GCC is the phase transform (PHAT). The traditional approach would have been to find the delay at which the GCC-PHAT maximizes and assume that delay to be the maximum likelihood (ML) estimate of the time delay. During the course of this research we found that the results were not as reliable as we would like them to be. Sometimes the estimated time-delays had their strongest peaks at completely different delays corresponding to strong reflections. DOA estimates from these wrong TDEs had very large errors in them, which would have been unsuitable for smooth camera steering.

We found that even though the ML estimator picked the wrong delay, the GCC-PHAT usually contains a weaker peak at the correct delay also. Our algorithm, called the Time Delay Selection (TIDES) algorithm, is based on not discarding these secondary peaks. The secondary delays are also collected as candidate TDEs and one of two conditions is used to select the best set of time delays from these candidates. The first selection criterion, called the Minimum Weighted Least Squares Error (MWLSE), selects that set of time delays that gives minimum least squares error when solved in the least squares sense. The least squares error is also weighted by the norm of the relative GCC strengths at those delays to give higher preference for the corresponding ML estimates. The second selection criterion, called the Minimum Weighted Time Delay Separation (MWTDS), selects that set of time delays that is closest in Euclidean

distance to the set of time delays that is a statistical average of previously selected time delays. A median filter of length 5 is used in this thesis to compute the statistical average of previously selected sets of time delays.

The TIDES-MWTDS algorithm was found to out-perform the ML estimator in moderate signal to reverberation ratio environments. These are environments where the distance between the source and the microphone array is much less than the distance of the array from walls and other reflective surfaces. For example a room of dimensions $4\,m \times 4\,m \times 3\,m$, with the array located at the center of the room and at a distance of up to $2\,m$ from the source, would be a good candidate for this kind of environment. This agrees well with the environment in a typical video conferencing room, with the microphone array placed on the table and people sitting around it. The algorithm was tested out using both simulations and actual recorded data and in some cases was found to be comparable to the SRP-PHAT algorithm. It was found that with the TIDES-MWTDS algorithm the number of small errors was larger than for SRP-PHAT, but the number of large errors was drastically reduced. This is ideally suited for a video camera steering application, since small errors in camera bearing are more acceptable than large errors. The TIDES-MWLSE algorithm also out-performs the ML estimator in moderate signal to reverberation ratio environments. This algorithm, like SRP-PHAT, shows better performance for small errors than the TIDES-MWTDS and poorer performance for large errors.

In the case of low signal to reverberation ratio environments (where the distance between the array and the source is comparable to the dimensions of the room), the TIDES-MWLSE was again found to out-perform the ML estimator though its performance was not as good as that of SRP-PHAT. The performance of the TIDES-MWTDS algorithm suffered badly because once it got stuck in a wrong estimate, it found it difficult to get back to the correct estimate. In summary, this thesis proposes two new methods to perform DOA estimation based on TDEs that show improvement over the ML estimator and sometimes comparable performance with the SRP-PHAT method, but at a much smaller computational requirement than the SRP-PHAT algorithm.

As interesting open issues yet to be addressed, we can look at better ways to make the weighted least squares errors for the candidate sets of time delays more comparable to each other. This will provide much better results with the TIDES-MWLSE algorithm and reduce the number of large errors that it produces in its current incarnation. Another avenue for improvement is to come up with better statistical averages to use with the MWTDS criterion for selecting the set of

time delays closest to the ones picked before. Another interesting study would be the performance of the TIDES algorithms when the source is moving. Because we are using information from previous frames in the TIDES-MWTDS algorithm, it would be interesting to study how the algorithm behaves when the source is moving at different speeds. Also since we found that information that corresponds to the correct time-delays is indeed present in the GCC-PHAT, one can try to come up with more reliable criteria to select those correct time-delays.

References

[1] Y. Huang, J. Benesty, and G. W. Elko, "Microphone Arrays for Video Camera Steering," Acoustic Signal Processing for Telecommunications, ed. S. L. Gay and J. Benesty, Kluwer Academic Publishers, 2000.

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *Proc. on Automatic Face and Gesture Recognition*, 1996, pp. 51-56.

[3] M. S. Brandstein and S. M. Griebel, "Nonlinear, model-based microphone array speech enhancement," Acoustic Signal Processing for Telecommunications, ed. S. L. Gay and J. Benesty, Kluwer Academic Publishers, 2000.

[4] J. H. DiBiase, H. F. Silverman, and M. Brandstein, "Robust Localization in Reverberant Rooms," Microphone Arrays, Springer-Verlag, 2001.

[5] B. V. Veen and K. M. Buckley, "Beamforming Techniques for Spatial Filtering," CRC Digital Signal Processing Handbook, 1999.

[6] H. Kamiyanagida, H. Saruwatari, K. Takeda, and F. Itakura, "Direction of arrival estimation based on non-linear microphone array," *IEEE Conf. On Acoustics, Speech and Signal Processing*, Vol. 5, pp. 3033-3036, 2001.

[7] C. H. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoustics, Speech and Signal Proc.*, vol. ASSP-24, No. 4, August 1976.

[8] K. Varma, T. Ikuma, and A. A (Louis) Beex, "Robust TDE-based DOA estimation for compact audio arrays," *IEEE Sensor Array and Multichannel Signal Proc. Workshop (SAM)*, August, 2002.

[9] J. P. Ianiello, "Time delay estimation via cross-correlation in the presence of large estimation errors," *IEEE Trans. Acoust. Speech, Signal Processing,* vol. 30, no. 6, pp. 998-1003, December 1982.

[10] L. B. Jackson, Digital Filters and Signal Processing, pp. 462-464, Kluwer Academic Publishers, 1996.

[11] DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, NIST Speech Disc 1-1.1, Oct. 1990.

[12] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *Proceedings, RADC Spectral Estimation Workshop*, pp. 243-258, October 1979.

[13] J. S. Bay, Fundamentals of Linear State Space Systems, pp. 125-126, WCB/McGraw-Hill, 1996.

[14] D. R. Raichel, The Science and Applications of Acoustics, pp. 16-17, Springer-Verlag, 2000.

[15] H. Kuttruff, Room Acoustics, 2nd Ed., Applied Science Publishers Ltd., 1979.

[16] J. B. Allen and A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943-950, April 1979.

[17] P. M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," *Journal of the Acoustical Society of America*, vol. 80, no. 5, pp. 1527-1529, November 1986.

[18] H. Kuttruff, Room Acoustics, 3rd Ed., Elsevier, 1991.

[19] J. G. Proakis and D. G. Manolakis, Digital Signal Processing, 3rd Ed., Prentice Hall, 1996.

[20] B. Champagne, S. Bedard, and A. Stephenne, "Performance of time-delay estimation in the presence of room reverberation," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, Issue 2, pp. 148-152, March 1996.

Vita

Krishnaraj M. Varma received his Bachelor of Technology degree in Applied Electronics and Instrumentation Engineering in October 1997 from the College of Engineering, Trivandrum, University of Kerala, India. After receiving his degree he worked for Tata Consultancy Services (TCS) as Assistant Systems Engineer. After a brief stint at the TCS center in Bombay, India, he was sent on assignment to the National Association of Securities Dealers (NASD) located in Rockville, MD, to take over the charge of production support for crucial market regulation software systems. In August 2000, he left TCS to start his graduate study at Virginia Tech. After completing the MSEE, he will continue his study at Virginia Tech, pursuing a Ph.D in Electrical Engineering. His research interests are in the areas of digital signal processing, controls and communications.