Echo Delay estimation to aid source localization in noisy environments

by

Raghuprasad Shivatejas Bettadapura

Thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

A. A. (Louis) Beex, Chair
Richard M. Buehrer
Jeffrey H. Reed

July 3, 2014
Blacksburg, VA

KEYWORDS: Sub-sample delays, sensors, source localization, time-frequency ratio of mixtures (TFRM), chirplet signal decomposition.

Echo Delay Estimation to aid source localization in noisy environments

Raghuprasad Shivatejas Bettadapura

(Abstract)

Time-delay estimation (TDE) finds application in a variety of problems, be it locating fractures or steering cameras towards the speaker in a multi-participant conference application. Underwater acoustic OFDM source localization is another important application of TDE. Existing underwater acoustic source localization techniques use a microphone array consisting of three or four sensors in order to effectively locate the source. Analog-to-digital (ADC) converters at these sensors call for a non-nominal investment in terms of circuitry and memory. A relatively inexpensive source localization algorithm is needed that works with the output of a single sensor. Since an inexpensive process for estimating the location of the source is desired, the ADC used at the sensor is capable only of a relatively low sampling rate. For a given delay, a low sampling rate leads to sub-sample interval delays, which the desired algorithm must be able to estimate. Prevailing TDE algorithms make some *a priori* assumptions about the nature of the received signal, such as Gaussianity, wide-sense stationarity, or periodicity. The desired algorithm must not be restrictive in so far as the nature of the transmitted signal is concerned.

A time-delay estimation algorithm based on the time-frequency ratio of mixtures (TFRM) method is proposed. The experimental set-up consists of two microphones/sensors placed at some distances from the source. The method accepts as input the received signal which consists of the sum of the signal received at the nearer sensor and the signal received at the farther sensor and noise. The TFRM algorithm works in the time-frequency domain and seeks to perform successive source cancellation in the received burst. The key to performing source cancellation is to estimate the ratio in which the sources combine and this ratio is estimated by means of taking a windowed mean of the ratio of the spectrograms of any two pulses in the received burst. The variance of the mean function helps identify single-source regions and regions in which the sources mix.

The performance of the TFRM algorithm is evaluated in the presence of noise and is compared against the Cramer-Rao lower bound. It is found that the variance of the estimates returned by the estimator diverge from the predictions of the Cramer-Rao inequality as the farther sensor is moved farther away. Conversely, the estimator becomes more reliable as the farther sensor is moved closer.

The time-delay estimates obtained from the TFRM algorithm are used for source localization. The problem of finding the source reduces to finding the locus of points such that the difference of its distances to the two sensors equals the time delay. By moving the pair of sensors to a different location, or having a second time delay sensor, an exact location for the source can be determined by finding the point of intersection of the two loci.

The TFRM method does not rely on a priori information about the nature of the signal. It is applicable to OFDM sources as well as sinusoidal and chirp sources.

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. Michael Buehrer and Dr. Jeffrey Reed for being on my committee and reviewing this work. I am also grateful to the Bradley Department of Electrical and Computer Engineering for allowing me use of laboratory facilities during the course of my research.

I would like to express my appreciation for the endless hours of discussion, technical and otherwise, that I have had with my colleague Tamoghna Roy 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.

Raghuprasad Shivatejas Bettadapura

Table of Contents

List of Figures

List of Abbreviations

TDE	Time-Delay estimation
3D	3 Dimensional
TDOA	Time-delay-of-arrival
OFDM	Orthogonal Frequency Division Multiplexing
TFRM	Time-frequency ratio of mixtures
ADC	Analog-to-digital converter
MODE	Method of Direction Estimation
ESPRIT	Estimation of signal parameters via rotational inverse transform
PHAT	Phase transform
BPSK	Binary Phase Shift Keying
IFFT	Inverse Fast Fourier Transform
STFT	Short time Fourier Transform
MVU	Minimum Variance Unbiased
DFT	Discrete Fourier Transform
DOA	Direction of Arrival
DSP	Digital Signal Processing (Processor)
DSPRL	DSP Research Laboratory
FIR	Finite Impulse Response
GCC	Generalized Cross Correlation
IDFT	Inverse Discrete Fourier Transform
LTI	Linear Time Invariant
ML	Maximum Likelihood
MUSIC	Multiple Signal Classification
PSD	Power Spectral Density
SNR	Signal to Noise Ratio
TDE	Time Delay Estimate

1. Introduction

1.1. Motivation for Research

Time-delay estimation for source localization has several everyday applications. Locating the origin of vibroorthographic signals associated with fractured joints could help in precisely locating the site of the injury or strain [1]. In a multi-participant environment, it may be desirable to provide speaker-specific location information, including echo (possibly from other speakers) and noise cancellation [2, 3]. The time delay between two successive seismic events can be used to more precisely locate the epicenter of an earthquake [4]. Underwater source localization using a distributed system consisting of three hydrophones is presented by B. Hodgkinson et al. [5]. Ferreira et al. present an optimal positioning algorithm for the placement of four sensors for the purpose of underwater source localization [6]. Postolache et al. propose an intelligent distributed virtual system for underwater sound classification [7]. In the realm of underwater source localization, the key to note in the papers by Hodgkinson et al., Postolache et al., and Ferriera et al., is the use of multiple sensors.

Source localization techniques fall broadly into three categories: direction of arrival (DOA)-based methods, time difference of arrival (TDOA)-based methods, and interaural level difference methods [8]. DOA methods typically need a large number of microphones for accurate source localization. DOA estimates from a set of microphones placed on a conference table can be used to automatically steer cameras in the direction of the speaker [9]. In most conference applications, a single speaker is speaking and the others are listening. The speaker could however be moving around in the room. Single source localization can be performed using either visual or aural information. A comprehensive tracking system using visual data was developed by Wren et al. [10]. The algorithmic complexity of the methods developed by Wren et al. entails non-trivial levels of power consumption.

TDOA-based methods that use a high sampling rate are employed for the purpose of two dimensional wideband source localization with three microphones [11-13]. A high sampling rate typically involves higher cost, in terms of the analog-to-digital converter (ADC) and peripheral circuitry. TDOA-based methods can be classified as belonging to one of the following categories: correlation-based methods and adaptive filter-based methods. There are two types of correlation-based methods: cross-correlation techniques, and algorithms that use the phase transform (PHAT). Adaptive filter methods generally use either sync-filter interpolation or least mean square equalization. Maximum likelihood estimation techniques fall into a separate third category.

Jacob Benesty et al. [14] propose a time-delay estimation (TDE) method to combat reverberation occurring in a multipath scenario by utilizing the multichannel cross-correlation coefficient (MCCC). The method proposed by Benesty et al. takes advantage of the redundancy provided by multiple microphone sensors to improve the time-delay estimates. Heinrich Meyr [15] uses cross-correlation to find the delay between two versions of a stochastic signal. Joseph Hassab and Ronald Boucher [16] design optimum filters before the generalized cross-correlation stage to perform two basic functions: to maximize the peak of the cross-correlation at the expected delay and to minimize the error between the incoming signal at the estimated delay and its estimated value. Seymour Stein [17] proposes a method to minimize the computational burden of calculating the cross-ambiguity function.

Maximum likelihood (ML) methods are based on minimizing a criterion arrived at based on certain assumptions about the incoming signal and the characteristics of the noise. A method combining both ML estimation and delay estimation based on phase information contained in the discrete-time Fourier transform is proposed by Chaoshu et al. [18]. Harri Saarnisaari [19] has designed a receiver based on the ML criterion for the estimation of unknown parameters of a periodic signal in a multipath channel. A. Satish et al. [20] proposed a ML-based method to achieve target tracking in a near-field using outputs from a large array of sensors. Satish made the assumption that the targets are narrowband signals modeled as sample functions of a Gaussian random process.

The use of the phase transform (PHAT) is advantageous because it enables accurate delay estimation in the case of wideband, periodic, and quasi-periodic signals. PHAT-based TDE is generally used in cases where signals are detected using arrays of microphones and environmental and reflective noises are observed. The phase transform is defined as the reciprocal of the cross correlation-based power spectrum. The chief reason for using the phase transform is to sharpen the peaks observed in the correlation function leading to a refinement in the estimate of a time delay. The other reason is its relative simplicity. The methods proposed by Brandstein et al. [11] and E. Lleida et al. [12] are based on the phase transform while the performance considerations of the phase transform are treated by Julian [13].

1.2. Fundamental Principles

Source localization methods that use multiple microphones or sensors in order to perform delay estimation impose an additional cost in terms of the circuitry at the ADCs at the sensors. Typically, in such a scenario, there is one source and multiple sensors and the signals received by those sensors form the basis for the localization of the source. A simple and less costly method is desired that makes use of possibly only one sensor and correspondingly only one ADC.

TDOA-based methods generally employ a high sampling rate on the source signals but this comes at the cost of more memory requirements and higher computational complexity. In order to be inexpensive, not only must the method function on only a single version of the received signal but it must also incorporate a low sampling rate.

The explanation of the problem of interest will be made clearer with an illustration of the experimental set-up. The arrangement of source and microphones (sensors) in which the algorithm is expected to operate is shown in Fig. 1.1.

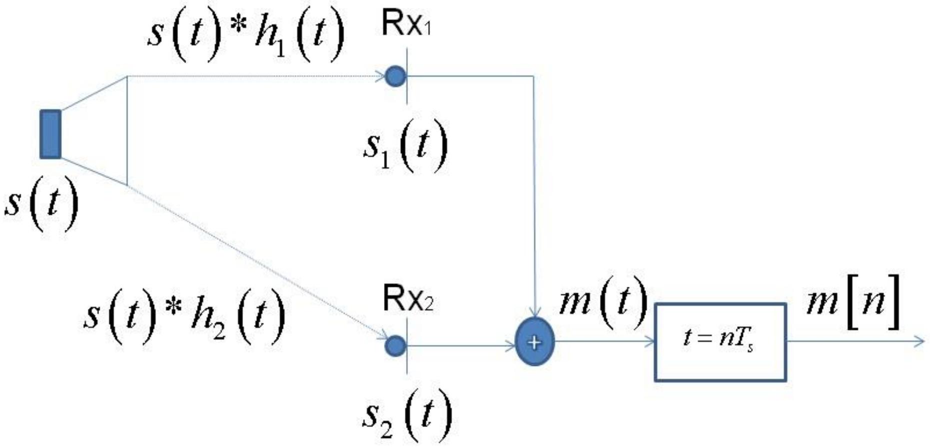


Fig. 1.1: The experimental set-up.

The source, denoted by a speaker in Fig. 1.1, emits a sound field that impinges on the two microphones Rx_1 and Rx_2. The microphones are not equidistant from the speaker: in this particular set-up Rx_2 is farther from the source than Rx_1, although this is not a requirement. As a consequence of the different distances from the source to the speakers, $s_2(t)$ is delayed with respect to $s_1(t)$. The problem is, given the mixture $m(t)$, which is the sum of the two microphone signals, to find the delay between $s_2(t)$ and $s_1(t)$. The sampling rate F_s must not be so high as to demand a prohibitively expensive investment in hardware. For a given delay in seconds between $s_2(t)$ and $s_1(t)$, a low sampling rate leads to a delay in terms of samples, that has both an integral and a fractional component. The mixture $m(t)$ consists of $s_1(t)$, an echo of $s_1(t)$ and noise and so, in totality, the delay estimation algorithm must be able to perform echo decomposition and detect sub-sample delays in the presence of ambient noise.

Given that only one version of the received signal is available to the algorithm, time delay estimation must be performed without *a priori* knowledge of the statistical nature of the signal. The algorithm must not be restrictive in its applicability to a narrow range of well-known and well-characterized signals. The ideal time-delay algorithm in this application would be universally applicable and would be blind to the nature of the signal transmitted by the source.

1.3 Overview of Research

The methods discussed for time delay estimation in this thesis fall into three categories: kernel-based signal transformation methods, correlation-based delay estimation, and – lastly - time delay estimation based on a ratio of mixtures in the time-frequency domain.

The organization of this thesis closely follows the progress of the research work over the last 3 semesters. The emphasis and objectives of the work being carried out evolved over time and this is reflected in the sequence of methods discussed.

Chapter 2 presents a brief survey of methods investigated and discarded due to one reason or another. The first method of Chapter 2 is one of homomorphic signal deconvolution, where echo estimation is performed by taking what is known in the literature as the “cepstrum” of the received signal. The cepstrum is, in essence, a logarithm of the magnitude of the Fourier transform of a signal. The limitations and inapplicability of homomorphic signal deconvolution to the problem at hand are discussed. Section 2.2 deals with the MODE-WRELAX algorithm and finds it to be inapplicable to the problem at hand because of the requirement that *a priori* information about the nature of the signal be available for correct functioning. The wavelet decomposition algorithm, treated in Section 2.3, pre-supposes continuity of both the signal and its echo and relies on a discontinuity at the point of superposition between the signal and its echo to detect the time delay.

In Chapter 3, a kernel-based chirplet signal decomposition algorithm is presented. When the material for Chapter 3 was being written, the aim was to perform time-difference-of-arrival (TDOA) estimation given knowledge of the nature of the source signal. The chirplet signal decomposition algorithm belongs to a general category of transform-based decomposition algorithms that transform a given signal into another domain in order to better glean information in that domain. The chirplet signal decomposition algorithm is able to successfully decompose a specific class of signals: the chirp signals. The algorithm matches the given received mixture to a kernel, constructed based on prior knowledge of the nature of the signal, and uses the cross-correlation between the mixture and the kernel to obtain delay information.

Having successfully executed TDOA for a source signal with known characteristics, Chapter 4 introduces a correlation-based delay estimation algorithm that estimates integral delays for a generic source. The existing correlation-based algorithms in the literature correlate two different versions of the source signal to find the delay between them. The method described in Chapter 4 uses a sub-sequence located near the beginning of the received signal and successively correlates it with the received burst. Due to the presence of an echo in the received signal, the cross-correlation function contains peaks at those points in time corresponding to the delay between the signal and its echo. The method is limited in that it is not able to estimate delays that possess a fractional part, and furthermore, it is generally difficult to accurately identify the beginning of a signal.

Chapter 5 presents a time-frequency ratio of mixtures (TFRM) method to estimate sub-sample delays given only a single mixture. The TFRM algorithm was a natural progression from the earlier two methods, with the assumptions made in Chapter 3 (*a priori* knowledge of the source signal) and Chapter 4 (integral delays) being relaxed. The aim is to identify single-signal regions in the received mixture and successively eliminate each of the signals $s_1(t)$ and $s_2(t)$ from the mixture. Single-signal regions are identified by relatively low variance in the ratio of the spectrograms of two independent instantiations of the mixtures. The mixtures must not be exactly identical because, in that case, the ratio function would be exactly one throughout. Slight differences in the mixtures, partially because of the random nature of the noise, contribute towards the successful functioning of the TFRM method. The method is evaluated in terms of the dependence of its performance on the strength of the observations in Section 5.7. The delay estimates obtained are used to arrive at source location information in Section 5.8. Essentially, the problem of pinpointing the co-ordinates of the source is one of finding the locus of a point which moves such that the difference of its distances from two fixed points is a constant. With one pair of microphones, there are potentially infinite locations for the source; narrowing the location down involves using more than one pair of microphone, or alternatively, the same pair of microphones at more than one location. A locus is obtained for each arrangement of the microphones and an intersection of more than one locus narrows the source location to a finite set.

2. Survey of TDE methods

Time-delay estimation (TDE) is by no means a novel problem: it has arisen in a variety of different contexts including angle-of-arrival estimation, source localization, and ultrasound in assessing tissue displacement, among others. A survey of methods that are relevant to the problem at hand are presented in this chapter, including homomorphic signal deconvolution, wavelet signal decomposition, and the MODE-WRELAX algorithm.

2.1. Homomorphic signal deconvolution

The term “Cepstrum” has its origins in a paper published in 1963 under the title “The Quefrequency Analysis of Time Series for Echoes: Cepstrum, Pseudoautocovariance, Cross-Cepstrum, and Saphe Cracking,” by Bogert, Healy, and Tukey [21]. The authors observed that the logarithm of the power spectrum of a signal containing an echo has an additive periodic component due to the echo, and thus the Fourier transform of the logarithm of the power spectrum should exhibit a peak at the echo delay. The Fourier transform of the logarithm of the power spectrum came to be called the “cepstrum,” an inversion of the word “spectrum,” resulting from what the authors felt were characteristics observed in the time domain that were generally associated with the frequency domain.

Almost contemporaneously, Oppenheim in 1967, proposed a new class of systems called *homomorphic systems.* Although non-linear in the traditional sense, they satisfy the principle of superposition. The concept of homomorphic filtering is very general but has been studied most extensively for combining the operations of multiplication and convolution because many signal models involve these operations [22]. The transformation of a signal into its cepstrum is a homomorphic transformation and the cepstrum is a fundamental part of the theory of homomorphic systems for reversing the process of convolution.

2.1.1 The complex cepstrum

Consider a stable sequence $x[n]$ whose z-transform expressed in polar form is

$$X(z) = |X(z)| e^{j\measuredangle X(z)} \tag{2.1}$$

where $|X(z)|$ and $\sphericalangle X(z)$ are the magnitude and angle, respectively, of the complex function $X(z)$. Since $x[n]$ is stable, its region of convergence includes the unit circle and therefore, its Fourier transform is defined and is represented as $X(e^{j\omega})$. The complex cepstrum corresponding to $x[n]$ is defined to be the stable sequence $\hat{x}[n]$ whose z-transform is

$$\hat{X}(z) = \log\left[X(z)\right] \tag{2.2}$$

As we require $\hat{x}[n]$ to be stable, the region of convergence includes the unit circle, and the complex cepstrum can be represented using the inverse Fourier transform as

$$\begin{aligned}\hat{x}[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\log\left[X\left(e^{j\omega}\right)\right]e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\log\left|X\left(e^{j\omega}\right)\right| + j\sphericalangle X\left(e^{j\omega}\right)\right]e^{j\omega n}d\omega\end{aligned} \tag{2.3}$$

In contrast to the complex cepstrum, the real cepstrum of a signal $c_x[n]$ is defined as the inverse Fourier transform of the logarithm of the magnitude of the Fourier transform, i.e.

$$c_x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\log\left|X\left(e^{j\omega}\right)\right|e^{j\omega n}d\omega \tag{2.4}$$

The real cepstrum is useful in many applications, and since it does not depend on the phase of $X(e^{j\omega})$, it is much easier to compute than the complex cepstrum. However, because it is based only on the Fourier magnitude it is not invertible, i.e. $x[n]$ cannot be recovered from $c_x[n]$. The complex cepstrum is somewhat more difficult to compute, but it is invertible.

2.1.2 Delay estimation and the cepstrum

The cepstrum is useful in the decomposition of a given signal into its composite echoes. If a signal and its replica both exist in the received signal, separated by a definite shift, then the cepstrum has a periodic nature to it, with the period being equal to the shift. This has been observed by Tukey et al. [21] and only the result is stated here. Consider, a mixture $m[n]$ such that:

$$m[n] = x[n] + x[n-D] \tag{2.5}$$

where D is an integer sample delay. The real cepstrum $c_m[n]$ will have peaks at integral multiples of D. This is best illustrated by an example.

2.1.3 Simulation example

The signal $x[n]$ was modeled as a sinusoid with a decaying exponential envelope. The plot of $x[n]$ is shown in Fig. 2.1.

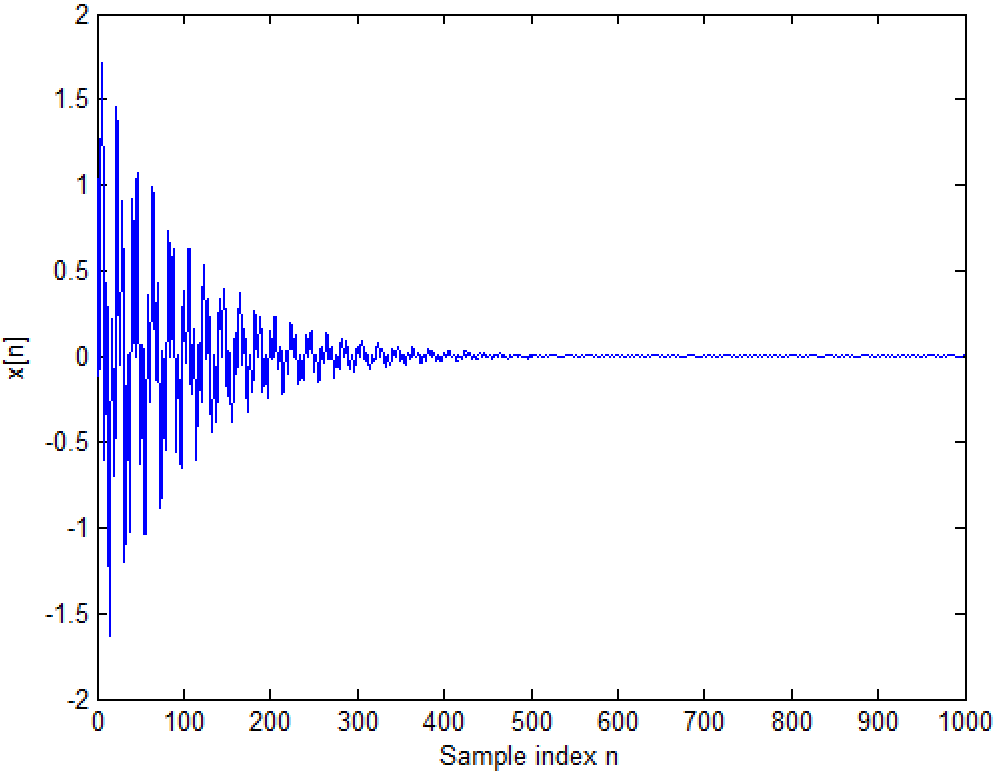


Fig. 2.1: Exponentially decaying sinusoid generated in MATLAB.

$x[n]$ is the function $\left(\sin(2\pi*0.05*n)+\cos(2\pi*0.17*n)\right)e^{-0.01n}$. A delay of 50 samples is applied to $x[n]$ and then summed with $x[n]$ to generate $m[n]$. The real cepstrum definition of

(2.4) was modified by replacing the integral with a summation, $X\left(e^{j\omega}\right)$ with the discrete Fourier transform, and $d\omega$ with a small increment. The real cepstrum is shown in Fig. 2.2.

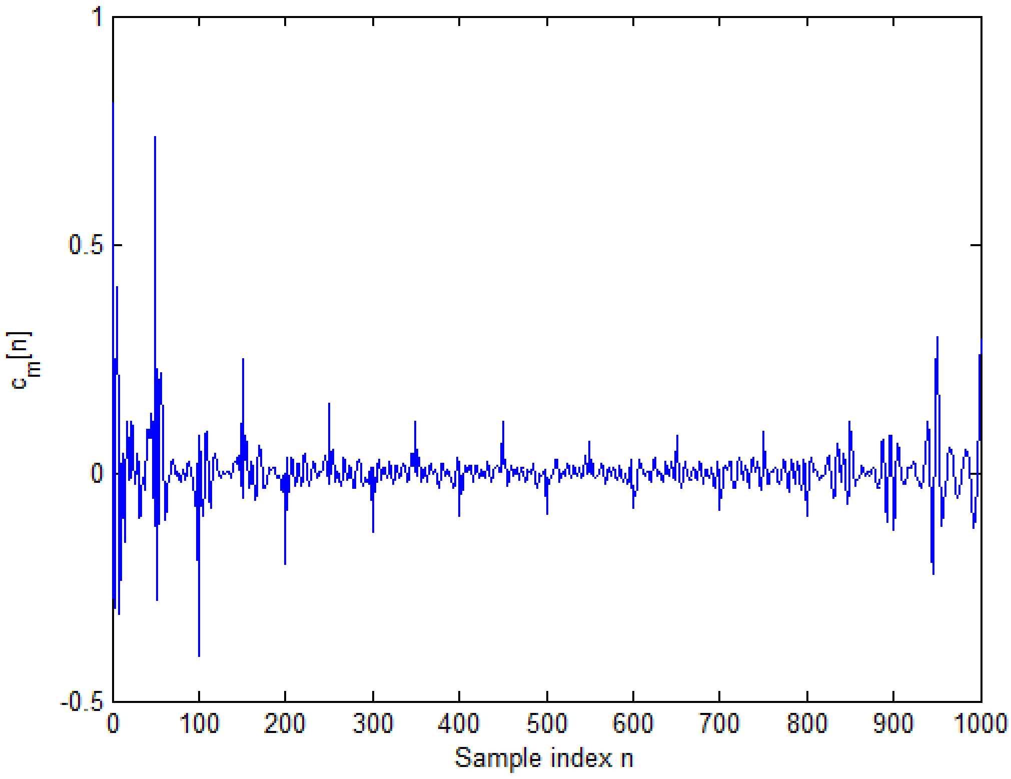


Fig. 2.2: Cepstrum of waveform shown in Fig. 2.1.

We observe that there are peaks of diminishing magnitude at integer multiples of 50, the delay applied to $x[n]$ with the largest of the peaks being at $n = 50$.

2.1.4 Experimentation on OFDM

The cepstrum method for time-delay estimation was employed on a simulated OFDM burst. The signal $x[n]$ was simulated in MATLAB as three identical OFDM bursts, each of length 512, with zeros in between. First, a bit stream of size 512 was created using MATLAB's *randsrc* function, which creates a random array of scalars, either -1 or 1, of the length specified by the user. This set of bits was passed through a BPSK modulator where the data bits 1 were mapped to $-\frac{1}{\sqrt{2}}$ and the data bits 1 were mapped to $\frac{1}{\sqrt{2}}$. A 512-point IFFT was then taken of the modulated bits to generate the OFDM burst. Two replicas of the created OFDM burst were appended to the first one, with 200 zeros interspersed between successive OFDM bursts; the result is the signal $x[n]$ shown in Fig. 2.3.

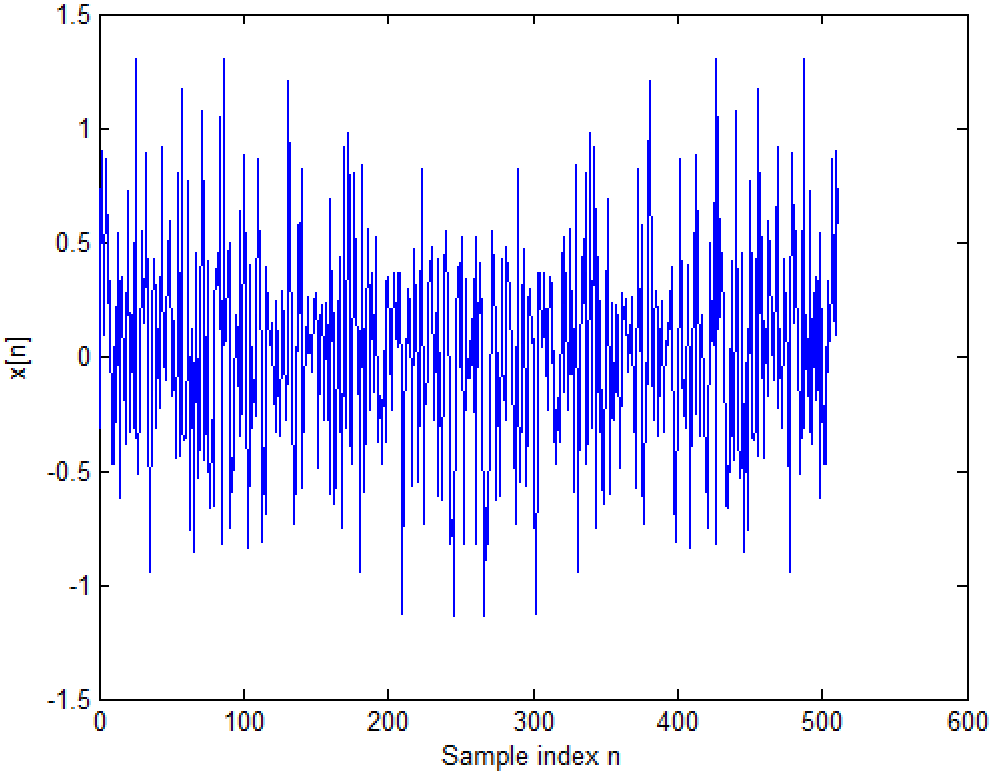


Fig. 2.3: Simulated OFDM burst $x[n]$.

A delay $D = 50$ samples was applied to the signal plotted in $x[n]$. $x[n]$ and $x[n-D]$ were summed and the summed signal is shown plotted in Fig. 2.4.

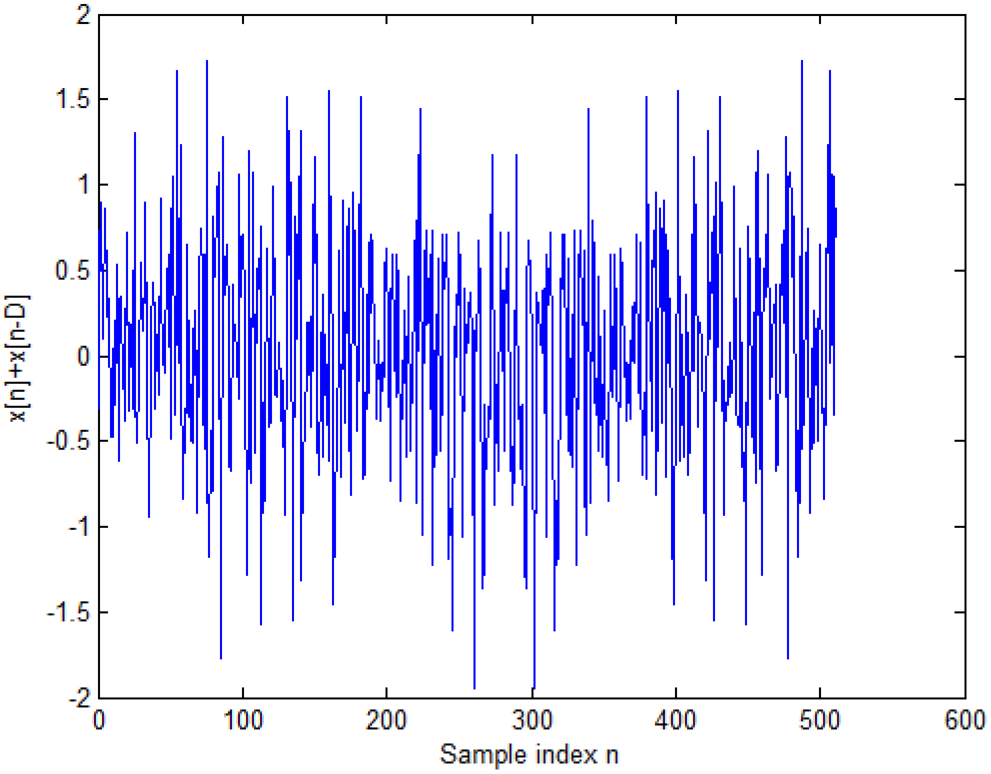


Fig. 2.4: Mixture consisting of $x[n]$ and its echo.

The cepstrum of the mixture $m[n]$ was calculated as explained in (2.4) and the result is shown plotted in Fig. 2.5.

In order to better illustrate the behavior of $c_m[n]$ near $n=50$, a zoomed version of Fig. 2.5 is shown in Fig. 2.6, focusing on $0 \le n \le 100$.

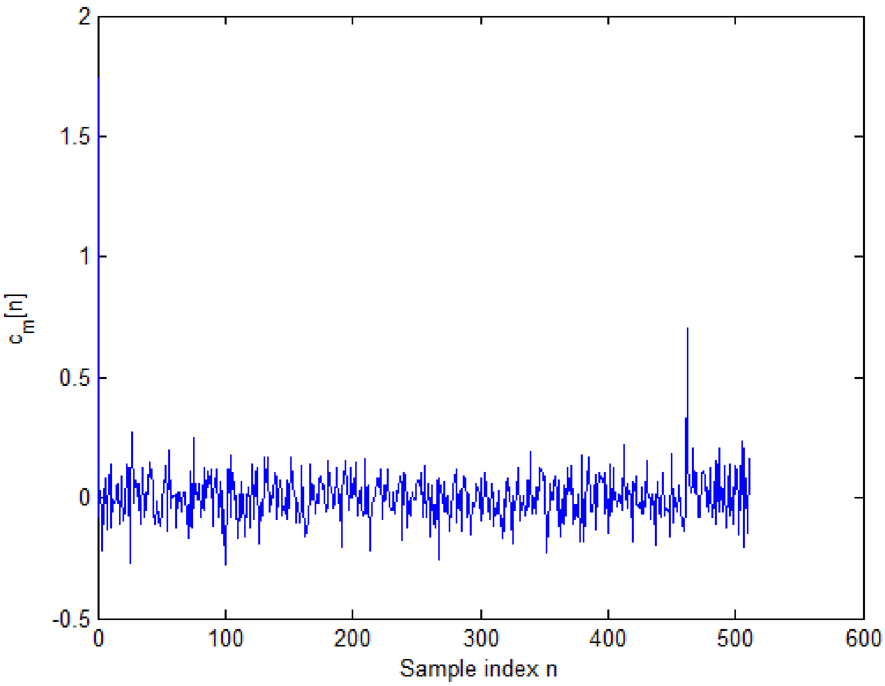


Fig. 2.5: Cepstrum of the mixture $m[n]$.

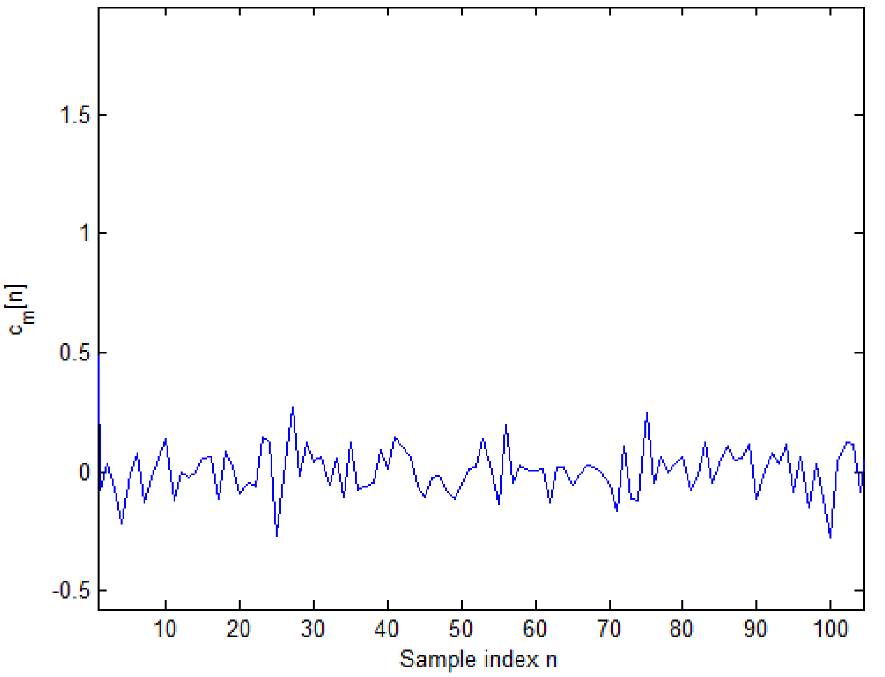


Fig. 2.6: Zoomed version of the cepstrum of the mixture $m[n]$.

In the zoomed version of the cepstrum of the mixture $m[n]$ shown in Fig. 2.6, a peak does not appear at $n = 50$, as theory dictates it should. The cepstral method thus fails to identify the echo deliberately introduced in the simulation. While the cepstral method works in theory on a decaying sinusoid such as the one in Fig. 2.1, it is unable to decompose a composite OFDM burst consisting of an OFDM signal and its echo.

2.1.5 Shortcomings

The cepstrum method is simple but it has some limitations, chief among them being the inability to detect sub-sample delays, i.e. delays that have both an integral and a fractional part. When the algorithm was tried on simulated OFDM bursts, it also failed. The cepstral decomposition method, while instructive is not very useful in the context of the problem to be solved, as explained in Section 1.2.

2.2. The MODE-WRELAX algorithm

The MODE-WRELAX method is a variant of the method of direction estimation algorithm (MODE) [23], [24], which employs advanced signal processing techniques to resolve overlapping pulses. The MODE algorithm is a type of eigenanalysis method and is closely related to such methods as MUSIC and ESPRIT [25], [26]. The MODE algorithm is based on a weighted Fourier transform and the WRELAX component refers to the relaxation applied to it. It has been used in the context of decomposition of superimposed time domain reflectometry (TDR) observations by Takeshi Ikuma in his Master’s thesis [27] but the underlying problem is one of resolution of closely spaced overlapping pulses.

2.2.1 Formulation of Problem

The MODE-WRELAX algorithm resolves closely spaced overlapping pulses described in the continuous-time domain by:

$$y(t) = \sum_{l=1}^{L} a_l s(t - \tau_l) \qquad (2.6)$$

There are L pulses and the corresponding linear combination coefficients a_l and time-delays τ_l. The sampled version of $y(t)$ can be written as:

$$y(nT_s) = \sum_{l=1}^{L} a_l s(nT_s - \tau_l) \tag{2.7}$$

where T_s is the sampling period. Ignoring time-domain aliasing, the N-point DFT Y_k of $y(nT_s)$ can be expressed as:

$$Y_k = S_k \sum_{l=1}^{L} a_l e^{j\omega_l k} \tag{2.8}$$

with

$$\omega_l = -\frac{2\pi\tau_l}{NT_s} \tag{2.9}$$

Equation (2.8) can be rewritten in matrix form as follows:

$$\mathbf{y} = \mathbf{SEa} \tag{2.10}$$

where

$$\mathbf{y} = \left[Y_{-N/2}\ Y_{-N/2+1}\ \ldots\ Y_{N/2-1}\right]^T \tag{2.11}$$

$$\mathbf{S} = diag\left\{S_{-N/2}, S_{-N/2+1}, \ldots, S_{N/2-1}\right\} \tag{2.12}$$

$$\mathbf{a} = \left[a_1\ a_2\ \ldots\ a_L\right]^T \tag{2.13}$$

and

$$\mathbf{E} = \left[\mathbf{e}(\omega_1)\ \mathbf{e}(\omega_2)\ \ldots\ \mathbf{e}(\omega_L)\right] \tag{2.14}$$

with

$$\mathbf{e}\left(\omega_l\right) \triangleq \left[e^{j\omega_l(-N/2)}\ e^{j\omega_l(-N/2+1)} \ldots e^{j\omega_l(N/2-1)} \right]^T \tag{2.15}$$

2.2.2 MODE-WRELAX Algorithm

Both the MODE and the WRELAX algorithm are approximations of the maximum likelihood method [3, 5, 6, 8]. Both algorithms aim at obtaining an optimal solution by minimizing the criterion $C_1\left(a,\omega\right)$

$$\arg\min_{a,\omega}\ C_1\left(a,\omega\right) = \arg\min_{a,\omega} \left\| \mathbf{y - SEa} \right\|^2 \tag{2.16}$$

Defining $\mathbf{\bar{E} = SE}$, we also define

$$\mathbf{P_E = \bar{E}\left(\bar{E}^H\bar{E}\right)^{-1}\bar{E}^H} \tag{2.17}$$

and

$$\mathbf{P_E^{\perp} = I - \bar{E}\left(\bar{E}^H\bar{E}\right)^{-1}\bar{E}^H} \tag{2.18}$$

Both $\mathbf{P_E}$ and $\mathbf{P_E^{\perp}}$ are projectors onto the span of the columns of $\mathbf{\bar{E}}$ and onto its orthogonal complement, respectively.

The MODE algorithm, based on an eigenanalysis technique, determines an optimal solution such that it minimizes the projection of $\mathbf{y}$ onto $\mathbf{P_E^{\perp}}$:

$$\arg\min_{a,\omega}\left\{C_{MODE}\right\} = \arg\min_{a,\omega}\left\{\mathbf{y}^H \mathbf{P}_E^{\perp}\mathbf{y}\right\} \tag{2.19}$$

The WRELAX algorithm, on the other hand, maximizes the projection of $\mathbf{y}$ onto $\mathbf{P_{\bar{E}}}$ through a series of iterations:

$$\arg\min_{a,\omega}\left\{C_{WRELAX}\right\} = \arg\min_{a,\omega}\left\{\mathbf{y^H P_{\bar{E}} y}\right\} \tag{2.20}$$

The MODE-WRELAX algorithm combines the efforts of both the MODE and WRELAX algorithms, aiming for improved accuracy and efficiency. The result of the MODE algorithm, the first stage, is expected to provide a good initialization for the WRELAX stage, which attempts to improve the MODE estimates.

2.2.3 Shortcomings

The MODE-WRELAX algorithm suffers from a crucial deficiency: it requires knowledge of the reference signal $s(t)$ in (2.6). This is a drawback because, very often, it is not possible to obtain a suitable reference signal for the time-delay estimation situation that is of interest to us. A method is sought that would be able to estimate sub-sample delays without *a priori* information about the nature of the signal. The MODE-WRELAX clearly fails on the former condition.

2.3. Wavelet Decomposition

Wavelet analysis finds application in detecting discontinuities or sudden jumps in signals. Given a continuous, differentiable signal $x(t)$, superimposing a delayed version of $x(t)$, say $x(t-\tau)$, would cause a jump or discontinuity at $t=\tau$. Wavelet decomposition gives us a way of detecting this discontinuity and hence to estimate the delay.

Wavelet analysis has been used in edge detection in image processing. In the case of delay estimation, the exercise is one of finding the point where the second derivative fails. The exact point at which differentiability fails has been called an *epoch* in the literature. The chirplet signal decomposition algorithm presented in Chapter 3 is, in the main, a means of detecting epochs in the received burst. In Fig. 2.7 an epoch is pointed to by arrows in both the plots.

Consider the mixture $m(t)$ constructed the following way:

$$m(t)=s(t)+s(t-\tau) \tag{2.21}$$

where $s(t)$ is a continuous, differentiable function. The addition of another, similar function, in this case $s(t-\tau)$, produces an epoch. This is shown in Fig. 2.7 by means of arrows.

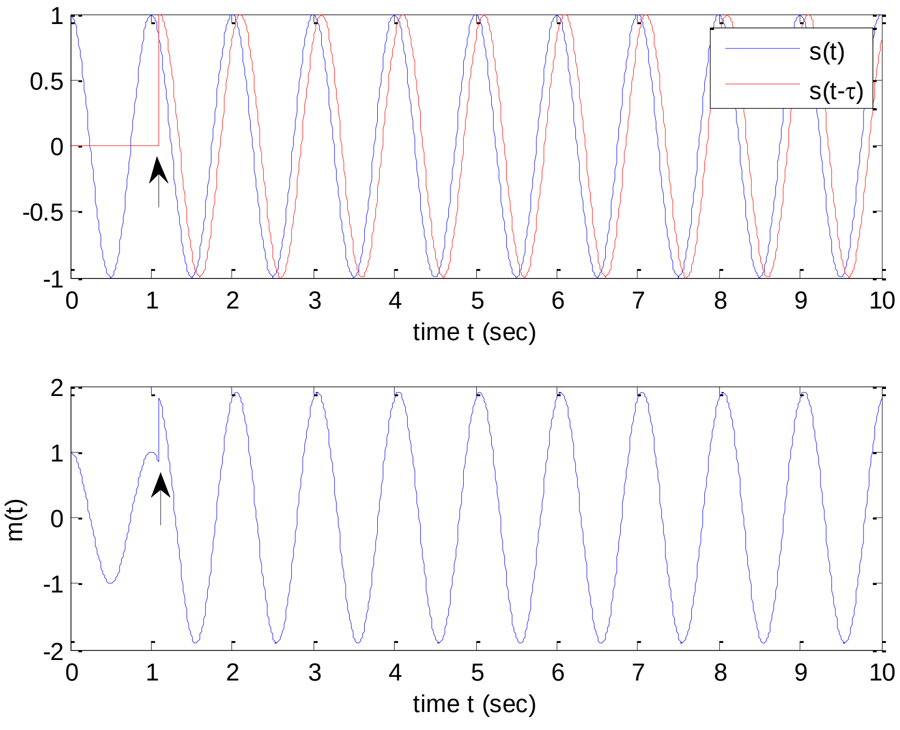


Fig. 2.7: Sinusoidal signal, delayed sinusoid and mixture of the two.

A discontinuity is clearly visible at $t = 1.1\,\text{s}$, the shift between $s(t)$ and $s(t-\tau)$. The wavelet decomposition method is illustrated for sinusoidal signals only. A wavelet decomposition of $m(t)$ up to level 2 is performed using MATLAB's wavelet decomposition algorithm *wavedec*. The wavelet used for the decomposition is a "Daubechies 4" wavelet. The Daubechies 4 wavelet is shown in Fig. 2.8.

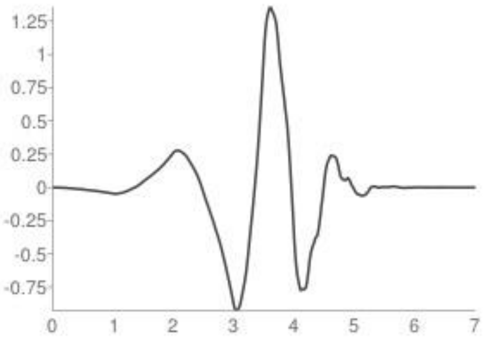


Fig. 2.8: The Debauchies 4 wavelet.

After extracting the wavelet decomposition at level 2, detail coefficients are extracted up to that level by using MATLAB's *detcoef* function which accepts as its arguments the wavelet decomposition coefficients and the level up to which detail is desired. More details about the process of extracting detail coefficients and how wavelet decomposition works can be found in MATLAB's documentation for *wavedec* and *detcoef*.

The detail coefficients at level 1 and 2 are shown in Fig. 2.9 below the mixture $m(t)$.

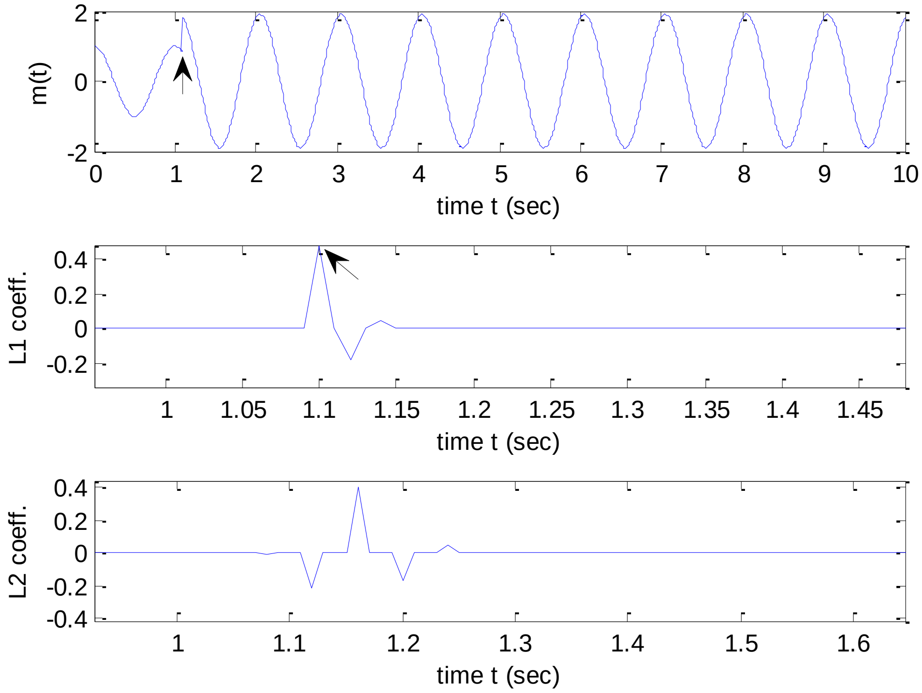


Fig. 2.9: Level 1 and level 2 coefficients of the mixture $m(t)$.

In the plot of the L1 coefficients in Fig. 2.9, a peak at the applied delay, 1.1 s, is visible. The plot of the level 2 coefficients is not as illuminating as regards the delay but some activity is observable in the region approximately bounded by $1.09\text{ s} \le t \le 1.22\text{ s}$. The behavior of the L2 coefficients serves as a coarse estimate of where the delay might be and the level 1 coefficients return a peak at the applied delay.

2.3.1 Shortcomings

The success of the wavelet decomposition approach is predicated on the assumption that the individual signals in the mixture are continuous and differentiable. In the real-world, this assumption is at best, unrealistic, and at worst, highly faulty. Information-bearing OFDM signals, or, for that matter, any acoustic signal conveying information is not likely to be continuous and differentiable in the time domain, definitely not when they are sampled at the receiver end. It is therefore not possible to detect echoes by looking for discontinuities in the received mixture because, frankly, they are to be found everywhere.

In conclusion, three methods: homomorphic signal deconvolution, MODE-WRELAX, and wavelet signal decomposition were described and their shortcomings stated. Homomorphic signal de-convolution is unable to find sub-sample delays, MODE-WRELAX relies on prior knowledge of the form of the signal, and wavelet signal decomposition for echo estimation fails in the face of lack of continuity and differentiability of the source signal.

3. Chirplet signal decomposition

In the previous chapter, we surveyed a few techniques that solve the problem of echo decomposition in a variety of contexts. Due to one reason or another, these methods were rejected as not being suitable for the problem under consideration. In this chapter a chirplet signal decomposition algorithm is presented.

Before solving the problem of separating OFDM signals that overlap at sub-sample delays, two sub-problems are considered. The first sub-problem, elaborated in this chapter, is of performing time-delay-of-estimation given knowledge of the nature of the source signal. Chapter 3 deals with time-delay estimation in the context of a chirp source.

Information-bearing OFDM signals are generated by taking the inverse Fourier transform (IFFT) of a modulated sequence of bits. The process of generating an OFDM burst is described in Section 2.1.4 and is presented here in equation form in (3.1):

$$o_1\left[n\right] = \mathrm{Re}\left(\left(\frac{1}{N}\sum_{k=0}^{N-1} b\left[k\right] e^{\frac{j2\pi kn}{N}}\right) e^{j\omega_b n}\right), n = 0,1,...N-1 \qquad (3.1)$$

$o_1\left[n\right]$ is the real portion of the OFDM burst that is fit for transmission. The point to be made here is that the OFDM burst can be visualized as a sum of sinusoids at equally spaced frequencies. A chirp signal is a variable-frequency sinusoid whose frequency is a function of time. Essentially, a chirp signal is a sinusoid whose frequency is a continuous function of time. There are many types of chirps: linear, quadratic, etc., reflecting the exact relationship between frequency and time. The motivation for exploring a chirplet (a more general term for a chirp) signal decomposition technique as a prelude to the ultimate problem of being able to separate information-bearing OFDM signals is the similarity between OFDM signals and a chirp signal: they can both be visualized as consisting of sinusoids of different frequencies.

The next section goes into the history of chirp signal decomposition algorithms after which a chirplet signal decomposition algorithm is presented and analyzed.

3.1. History of signal decomposition algorithms

Jafar Saniie and Ramazan Demirli [29] have proposed a sparse signal decomposition algorithm based on the matching pursuit (MP) method using a generic time-frequency dictionary.

The idea is to express a given signal as the sum of a number of small functions chosen from a redundant dictionary. Saniee and Demirli use the Gabor dictionary [29]. The Gabor dictionary is named after Dennis Gabor who proposed the Gabor transform [30].

The Gabor transform expresses any given function $f\left(t\right)$ in terms of a sum of Gabor elementary functions. These Gabor elementary functions are a product of a chosen window function and exponentials.

Yinghui and Michaels [31] present a matching pursuit algorithm for the analysis of ultrasonic signals by using a small set of Gabor functions tailored for a structured health monitoring system. H. Jin-Chul *et al*. [32] utilize a matching pursuit algorithm with a chirp dictionary to perform non-destructive evaluation of waveguide transmissions. The chirp dictionary was originally developed to represent echoes propagating through dispersive media. Cardoso and Saniie [33] have presented a modified, continuous wavelet transform (MCWT) based on the Gabor-Helmstrom transformation to perform decomposition of ultrasonic echoes. However the MCWT has not proven effective in decomposing ultrasonic echoes with chirp characteristics.

A successive parameter estimation algorithm in the context of ultrasonic imaging signals with chirplet characteristics is presented by Demirli and Saniie [34]. The algorithm uses a chirplet signal decomposition algorithm followed by an iterative procedure to determine such parameters as delay, bandwidth factor, center frequency, and phase. The algorithm aims to express a chirp-like signal in terms of Gaussian chirplets, which are sparse and energy-preserving. The sparseness property aims for a compact representation of the complex signal by targeting a limited dictionary of chirp signals. The energy preservation property ensures coherent distribution of energy into constituent functions, thereby enabling the linear addition of the individual time-frequency (TF) functions to obtain the TF representation of the signal as a whole. The method proposed by Demirli and Saniie [34] is replicated in this chapter with a focus on the delay parameter.

3.2. Mathematical description

A single chirp echo is modeled as follows:

$$f_{\Theta}\left(t\right)=\beta e^{-\alpha_1\left(t-\tau\right)^2}e^{i\left(2\pi f_c\left(t-\tau\right)+\phi+\alpha_2\left(t-\tau\right)^2\right)} \tag{3.1}$$

where $\Theta = [\alpha_1, \alpha_2, \beta, f_c, \phi, \tau]$ is the parameter vector; α_1 is the bandwidth factor, α_2 is the chirp rate, β is the amplitude, f_c is the center frequency, ϕ is the phase, and τ is the time-of-arrival of the chirp echo [34]. The chirplet transform (CT) is defined as

$$CT\left(\hat{\Theta}\right) = \int_{-\infty}^{\infty} f_{\Theta}(t)\psi_{\hat{\Theta}}^{*}(t)\,dt \tag{3.2}$$

where $\psi_{\hat{\Theta}}^{*}(t)$ is the conjugate of the chirplet kernel $\psi_{\hat{\Theta}}(t)$. The chirplet kernel $\psi_{\hat{\Theta}}(t)$ is defined as:

$$\psi_{\hat{\Theta}}(t) = \eta e^{-\gamma_1(t-b)^2} e^{i\left(\omega_0\left(\frac{t-b}{a}\right)+\theta+\gamma_2(t-b)^2\right)} \tag{3.3}$$

The maximum similarity between $f_{\Theta}(t)$ and the chirplet kernel $\psi_{\hat{\Theta}}(t)$ can be used to correctly estimate the parameters of interest. $CT\left(\hat{\Theta}\right)$ is a function in six-dimensional space but by keeping all but two parameters constant, it is reduced to two dimensions. The peak of $CT\left(\hat{\Theta}\right)$ can be used to estimate the center frequency f_c and the time-of-arrival τ. The chirplet transform can be expressed as:

$$CT\left(\hat{\Theta}\right) = \beta\left(2\pi\gamma_1\right)\frac{1}{\sqrt{\alpha_1+\gamma_1-i\alpha_2+i\gamma_2}}e^{\left[\frac{4(K_1-K_2)-K_\omega}{\alpha_1+\gamma_1-i\alpha_2+i\gamma_2}-i(\phi-\theta)\right]} \tag{3.4}$$

where,

$$\begin{aligned} K_\omega &= \left(\omega_c - \frac{\omega_0}{a}\right)^2, K_1 = \left(i\frac{\omega_0}{a}(\alpha_1 - i\alpha_2) + i\omega_c(\gamma_1 + i\gamma_2)\right)(b-\tau) \\ K_2 &= (\alpha_1 - i\alpha_2)(\gamma_1 + i\gamma_2)(b-\tau)^2 \end{aligned} \tag{3.5}$$

The derivative of a function at its maximum is zero. In order to find the derivative of the chirplet transform, its magnitude is estimated, which is given by:

$$|CT\left(\hat{\Theta}\right)| = \beta\left(2\pi\gamma_1\right)^{\frac{1}{4}}\left[\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right]^{-\frac{1}{4}}\exp\left(\begin{array}{l} \dfrac{-\left(\omega_c-\dfrac{\omega_0}{a}\right)^2+\left(\alpha_1+\gamma_1\right)^2}{4\left(\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right)} \\ -\dfrac{\left(\omega_c-\dfrac{\omega_0}{a}\right)\left(\alpha_1\gamma_2+\alpha_2\gamma_1\right)\left(b-\tau\right)}{\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2} \\ -\dfrac{\left(\alpha_1{}^2\gamma_1+\alpha_2{}^2\gamma_1+\gamma_1{}^2\alpha_1+\gamma_2{}^2\alpha_1\right)\left(b-\tau\right)^2}{\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2} \\ \end{array}\right) \tag{3.6}$$

The maximum of (3.6) can be obtained by taking partial derivatives of $|CT\left(\hat{\Theta}\right)|$ with respect to the parameters *a* and *b*. It was mentioned earlier that the only parameter of interest was the delay parameter *b*. The partial derivative with respect to *b* is:

$$\frac{\partial|CT\left(\hat{\Theta}\right)|}{\partial b} = |CT\left(\hat{\Theta}\right)|\left\{\begin{array}{l} \dfrac{-\left(\alpha_1{}^2\gamma_1+\alpha_2{}^2\gamma_1+\alpha_1\gamma_1{}^2+\alpha_1\gamma_2{}^2\right)\left(b-\tau\right)}{2\left(\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right)} \\ +\dfrac{\left(\dfrac{\omega_0}{a}-\omega_c\right)\left(\alpha_1\gamma_2+\alpha_2\gamma_1\right)}{4\left(\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right)} \end{array}\right\} = 0 \tag{3.7}$$

In order to estimate the delay parameter in (3.7), another equation is needed and this is supplied by partially differentiating with respect to a. Taking the partial derivative with respect to *a*, we have:

$$\frac{\partial|CT\left(\hat{\Theta}\right)|}{\partial a} = |CT\left(\hat{\Theta}\right)|\left\{\frac{-\omega_0 a^{-2}\left(\alpha_1\gamma_2+\alpha_2\gamma_1\right)\left(b-\tau\right)}{2\left(\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right)}+\frac{\dfrac{\omega_0}{a^2}\left(\dfrac{\omega_0}{a}-\omega_c\right)\left(\alpha_1+\gamma_1\right)}{2\left(\left(\alpha_1+\gamma_1\right)^2+\left(\alpha_2-\gamma_2\right)^2\right)}\right\} = 0 \tag{3.8}$$

Solving (3.7) and (3.8) yields, as a condition for maximum,

$$b=\tau, \frac{\omega_0}{a}=\omega_c \tag{3.9}$$

This forms the basis for the time-of-arrival calculation approach. Given a received mixture

$$m(t) = s_1(t) + c_1 s_2(t) \tag{3.10}$$

where $s_1(t)$ and $s_2(t)$ are two chirp-like signals intended to represent the signals received along two separate paths from the source to the two microphones, a kernel function is constructed whose parameter *b* is variable. Real-world computations are carried out in discrete-time rather than continuous-time and so the integral in (3.2) is replaced by a summation. The summation is iteratively evaluated for values of *b* and the points of maximum correlation are the times-of-arrival of the signals $s_1(t)$ and $s_2(t)$. Rewriting (3.2) to make it suitable for discrete-time signals, we have,

$$CT\left[\hat{\Theta}\right] = \sum_{n=-\infty}^{\infty} m[n]\psi_{\Theta}^{*}[n]\Delta n \tag{3.11}$$

where Δn is a small quantity. Since all parameters except the delay parameter *b* are held constant, it makes more sense to rewrite (3.11) as exclusively a function of *b* yielding ,

$$CT[b] = \sum_{n=-\infty}^{\infty} m[n]\psi_{b}^{*}[n]\Delta n \tag{3.12}$$

The condition in (3.9) calls for the maximum of the chirplet transform to be reached at $b = \tau$, where τ is the time-of-arrival of the signal. In this case, since there are two signals that constitute the mixture $m[n]$, two peaks will occur corresponding to the two times-of-arrival of the signals $s_1[n]$ and $s_2[n]$ (the discrete-time versions of $s_1(t)$ and $s_2(t)$).

3.3. Experimentation

3.3.1 Simulation exercise

A chirp signal is generated with the parameters $\alpha_1 = 0.02$, $\alpha_2 = 4$, $f_c = 20$ Hz, and $\phi = \pi$ and $\tau = 0$. The signal $s_1(t)$ is obtained by taking the real part of the chirp signal generated with these parameters and shown in Fig. 3.1.

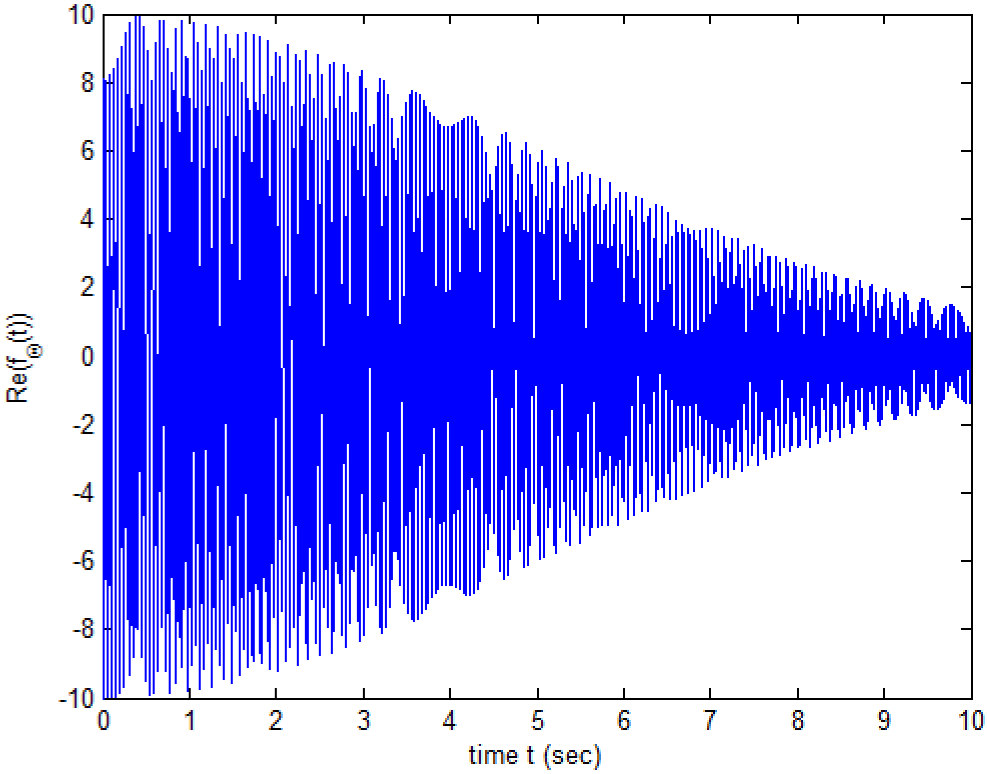


Fig. 3.1: Real part of the chirp signal generated in MATLAB.

A second chirp signal was generated by changing only the delay parameter τ and keeping all the other parameters unchanged. In this case τ was changed from 0 to 2.5. $s_2(t)$ was generated by taking the real part of this second chirp signal and $s_1(t)$ and $s_2(t)$ were mixed as shown in (3.10) with $c_1 = 0.4$. The delayed signal $s_2(t)$ is shown plotted in Fig. 3.2. The mixed signal $m(t)$ appears in Fig. 3.3.

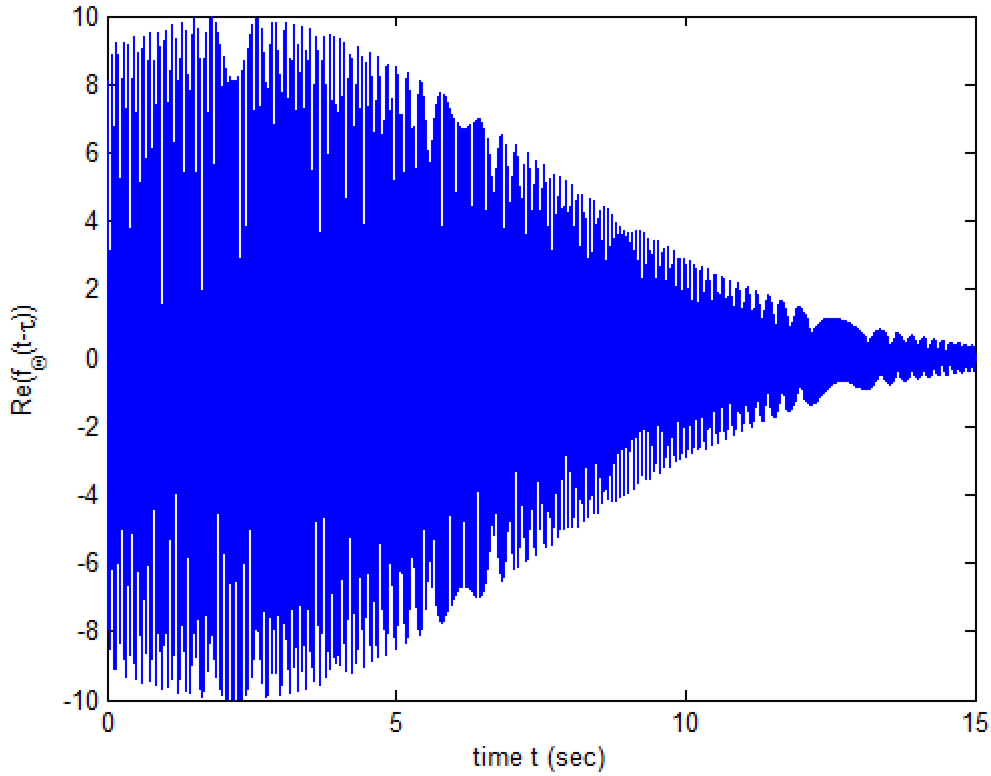


Fig. 3.2: Chirp signal of Fig. 3.1 delayed.

The chirplet kernel was generated with the same parameter vector as both the chirp signals above except the parameter *b* which was held variable. Under these conditions, the chirplet transform reduces to a matched filter correlation between the received mixture and the kernel. The delay parameter *b* was varied from 0 to 10 in steps of 0.01. The chirplet transform as defined in (3.12) was estimated with $\Delta n = 0.01$, for each value of *b*, and its magnitude stored in an array defined to be as long as the array containing the test values of the parameter *b*. The magnitude of the chirplet transform as a function of the delay parameter *b* is shown in Fig. 3.4.

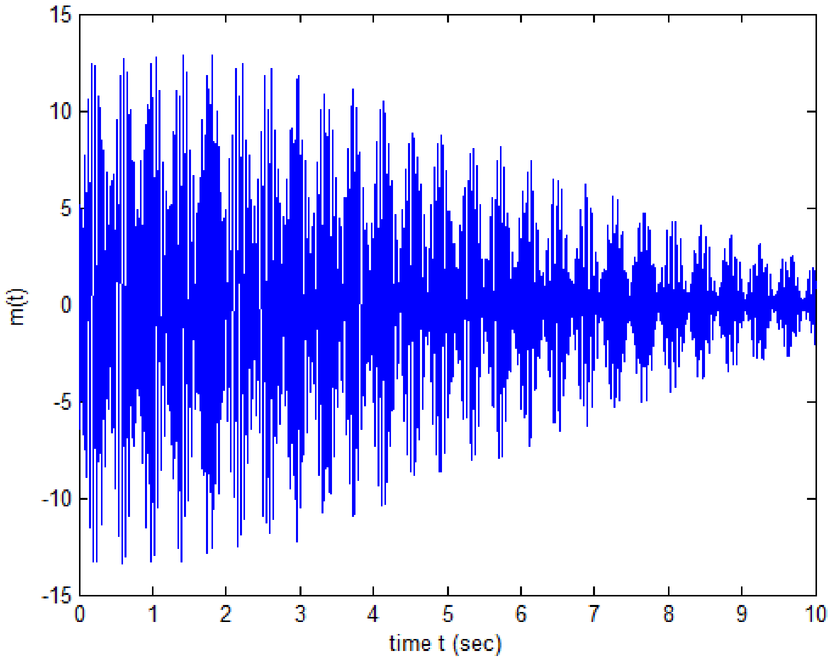


Fig. 3.3: The mixed signal $m(t)$.

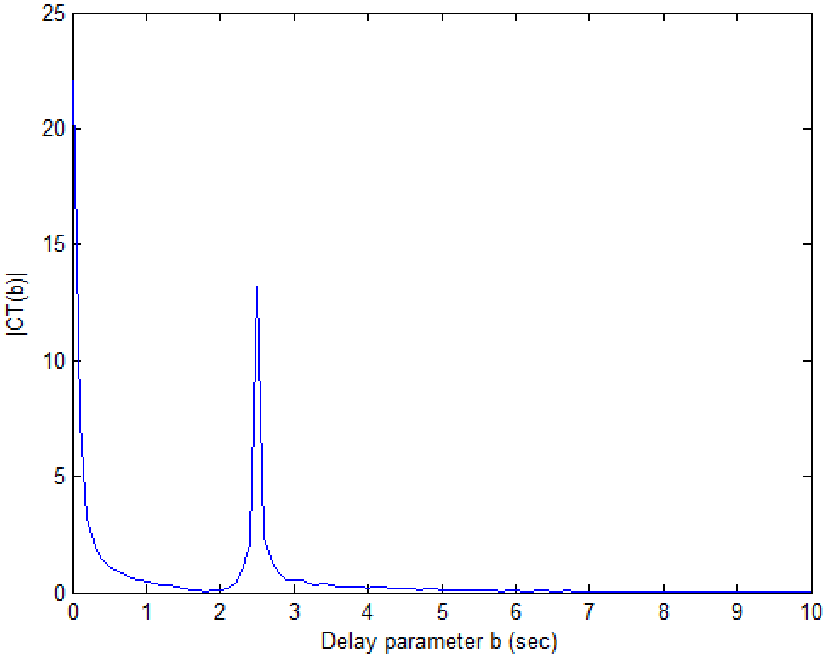


Fig. 3.4: The chirplet transform of the mixture $m(t)$.

The chirp transform magnitude in Fig. 3.4 decays from $b = 0$ sec before a peak is reached at $b = 2.5$ sec, corresponding to the time-of-arrival of the signal $s_2(t)$. The second peak appears diminished relative to the peak at b=0 sec and this is the result of the scaling factor c_1 which was set to 0.4.

The above-mentioned experiments were carried out in the DSPRL with one microphone placed at a distance of 36 cm from the source and the other at a distance of 69 cm from the source. The experimental set-up, although repeated at other points in the thesis is shown below in Fig. 3.5.

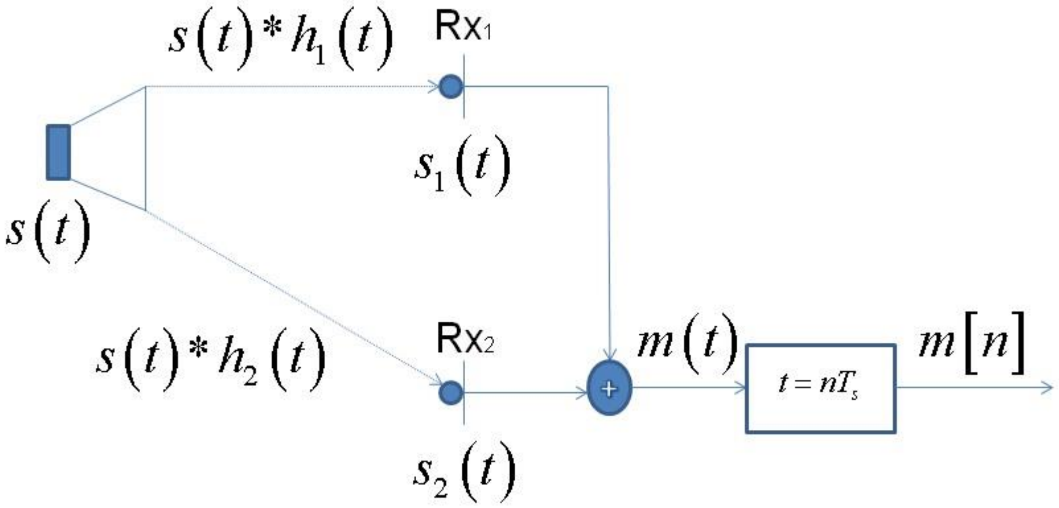


Fig. 3.5: Experimental set-up for mixture recording.

The chirp signal at the transmission side was generated in MATLAB with the parameters $\alpha_1 = 2$, $\alpha_2 = 100$, $f_c = 700$ Hz, $\phi = 1$, and $\beta = 100$. The generated chirp signal, before transmission, is shown in Fig. 3.6.

If the transmission was performed from one laptop, then, in order to avoid coupling between the transmission and reception apparatus, the reception of the signal was done in another laptop. The summing junction of Fig. 3.5 was realized as two resistors in series and each of the "operands", or the signals on the two paths in this case, are voltages across the individual resistors. The voltage across the series arrangement is the sum of the signals $s_1(t)$ and $s_2(t)$. The recording of the received signal was done on 2 channels and the signal on each was recorded using MATLAB's *wavrecord* for a duration of 15 sec. The signal on each of the channels was identical and equal to $m(t)$, defined in (3.10). The received burst consists of the sum of the signals arriving on the two paths and is shown in Fig. 3.6.

A zoomed version of Fig. 3.6 is shown in Fig. 3.7 focusing on the signal portion of the received burst.

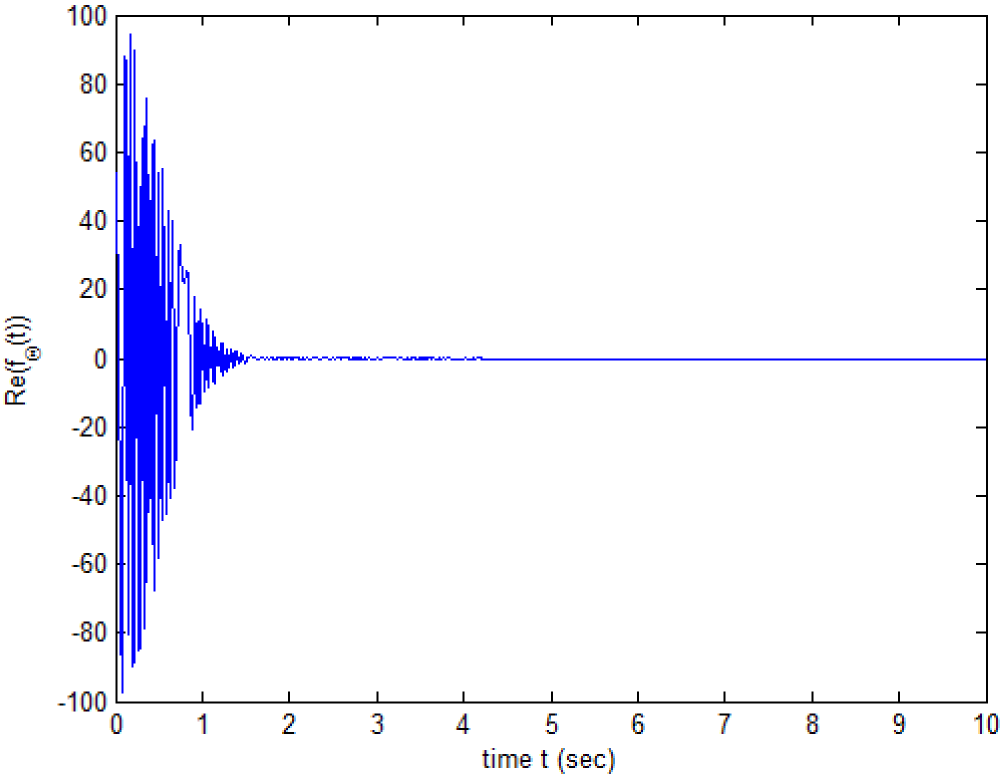


Fig. 3.6: Chirp signal before transmission.

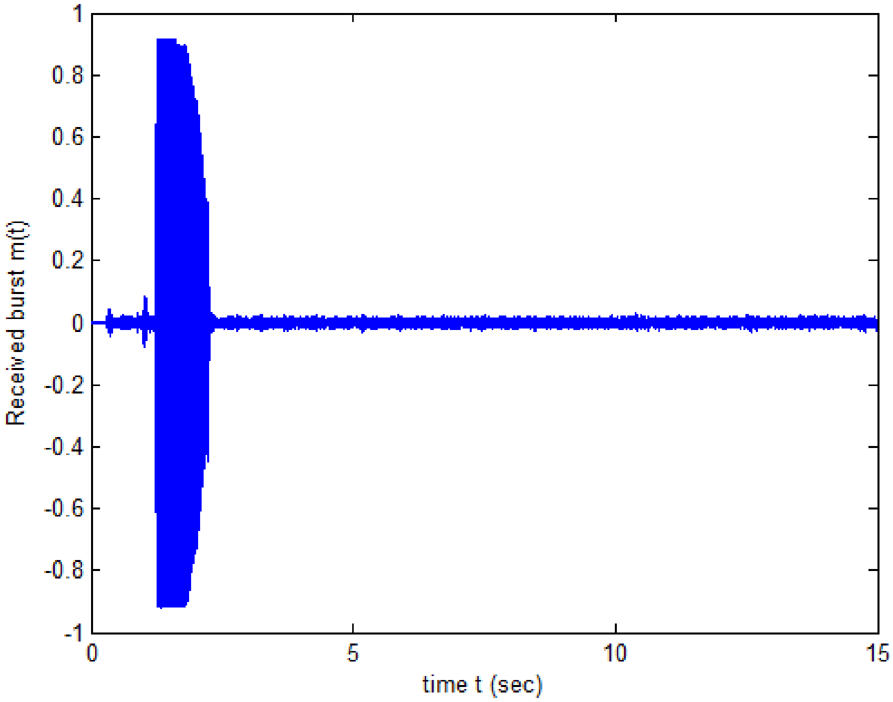


Fig. 3.7: Received signal $m(t)$.

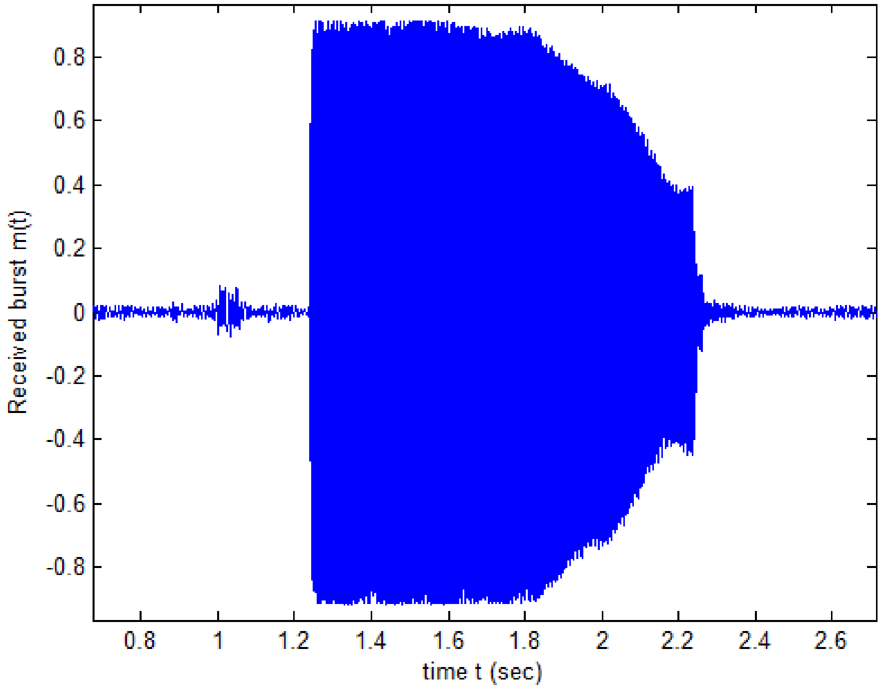


Fig. 3.8: Zoomed version of Fig. 3.6.

The chirplet signal decomposition method was applied to the received burst $m(t)$ and the search for the times-of-arrival was carried out between 0 and 15 ms. The chirplet transform magnitude, $|CT(b)|$, is shown as a function of b in Fig. 3.9. As only the time-of-arrival parameter is of interest, all parameters except the time-of-arrival parameter b are known and the chirplet transform is essentially a single-dimensional function and the problem of estimating the time-of-arrival is one of matching the received signal to the kernel function constructed beforehand.

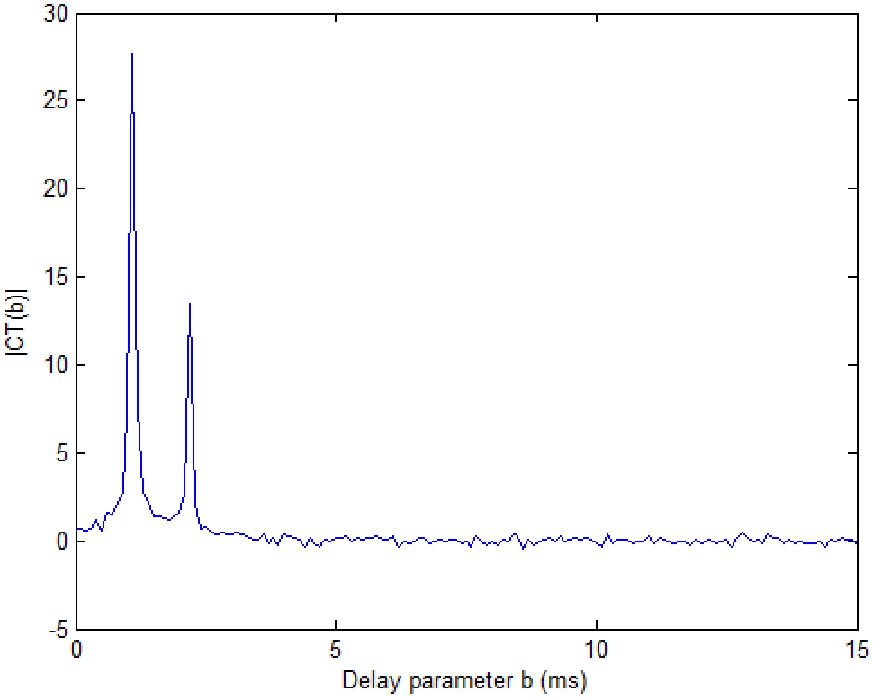


Fig. 3.9: The chirplet transform of the received mixture.

A zoomed version of Fig. 3.9 appears in Fig. 3.10 to better illustrate the location of the individual peaks.

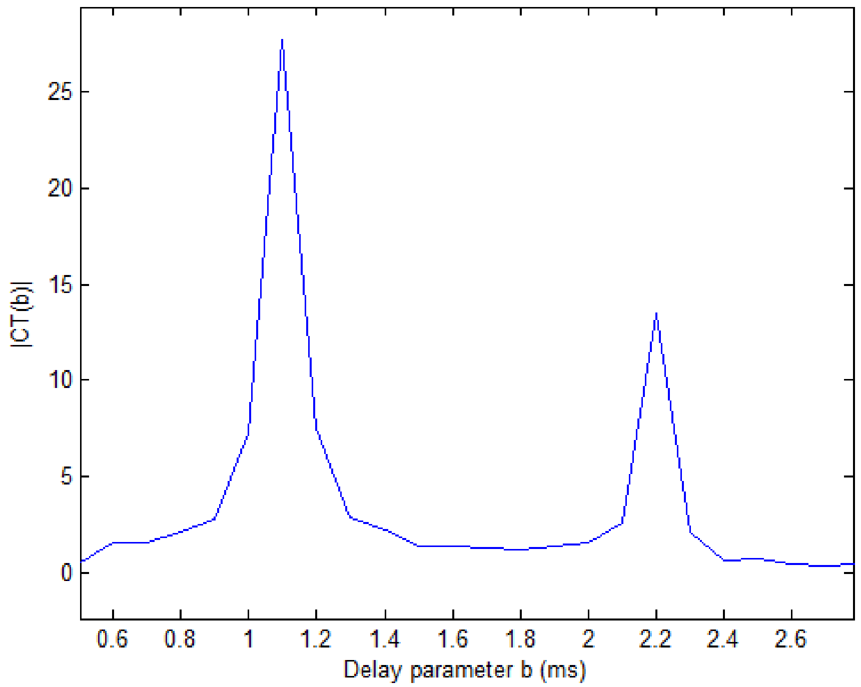


Fig. 3.10: Zoomed version of Fig. 3.8.

The two peaks occur at $b = 1.1\text{ ms}$ and $b = 2.2\text{ ms}$. Let us see how this compares with a back-of-the-envelope calculation. The nearer microphone is at a distance of 36 cm from the source and the farther microphone is at a distance of 69 cm from the source. The time of arrival of $s_1(t)$ is

$$t_1 = \frac{0.36\text{ m}}{341.2\text{ m/s}} = 0.0011\text{ s} \tag{3.12}$$

and the time-of-arrival of $s_2(t)$ is

$$t_2 = \frac{0.69\text{ m}}{341.2\text{ m/s}} = 0.0020\text{ s} \tag{3.13}$$

The waveform in Fig. 3.8 is flat-topped. The transmitted signal shown in Fig. 3.6 has a peak level of 100, enough to drive the ADC in the receiver into saturation. Any sample that exceeds the range of the ADC is mapped to one of the extremes of the range.

The experiments were repeated with the transmitted signal scaled down with a scaling factor of 0.01, so that the transmitted signal was restricted between 1 and -1 and the other parameters were, $\alpha_1 = 2$, $\alpha_2 = 100$, $f_c = 700$ Hz, $\phi = 1$, and $\beta = 100$, identical to the parameters used in generating the transmitted signal in Fig. 3.6.. The scaled down transmitted signal is shown in Fig. 3.11.

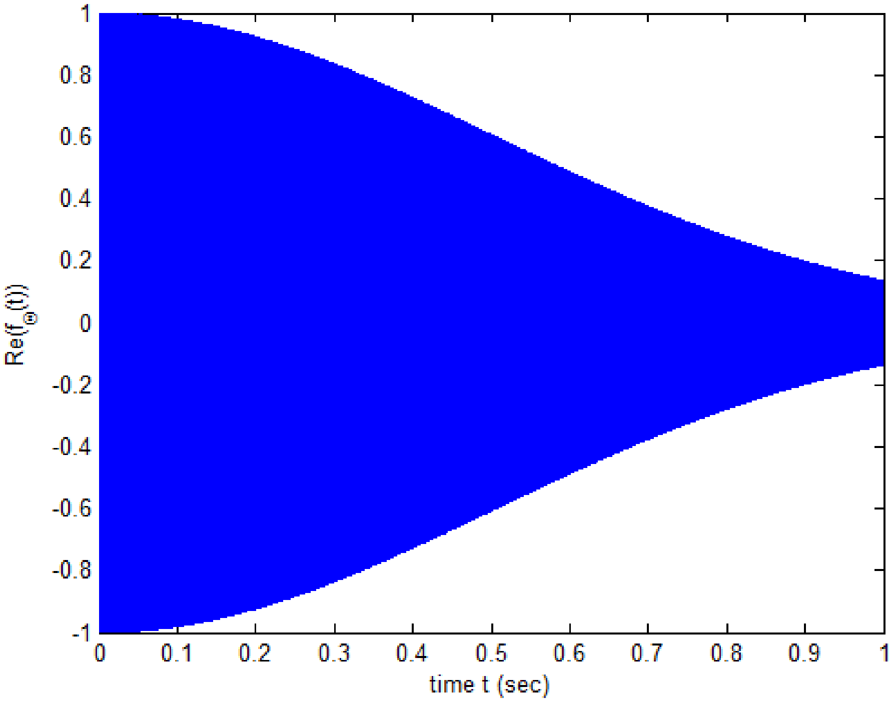


Fig. 3.11: Scaled-down transmitted signal.

The received signal is shown plotted in Fig. 3.12.

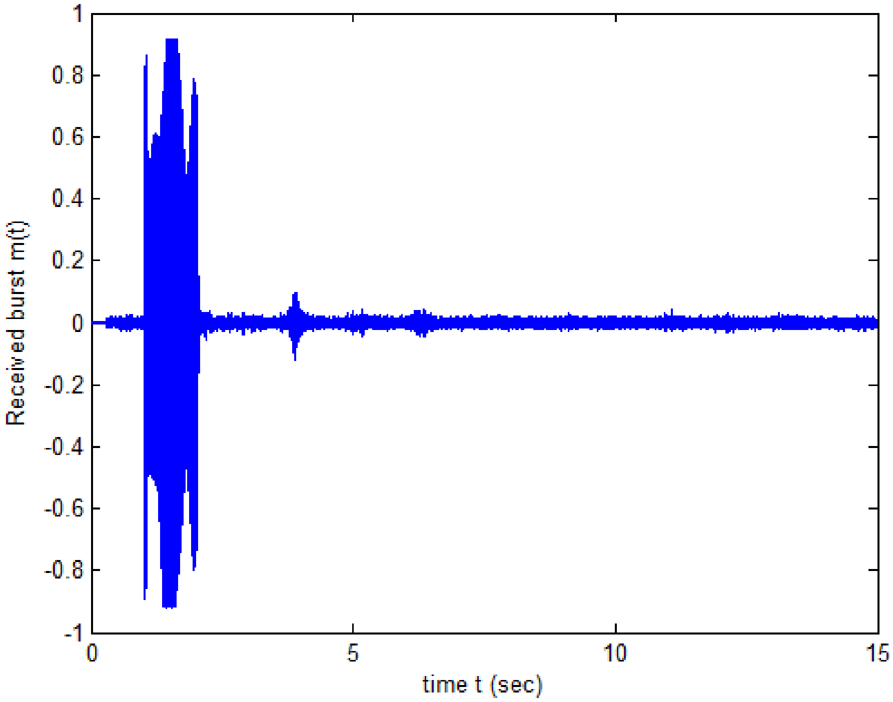


Fig. 3.12: Received signal $m(t)$.

The chirplet transform of the received burst was calculated in the same manner as previously described and is shown plotted in Fig. 3.13.

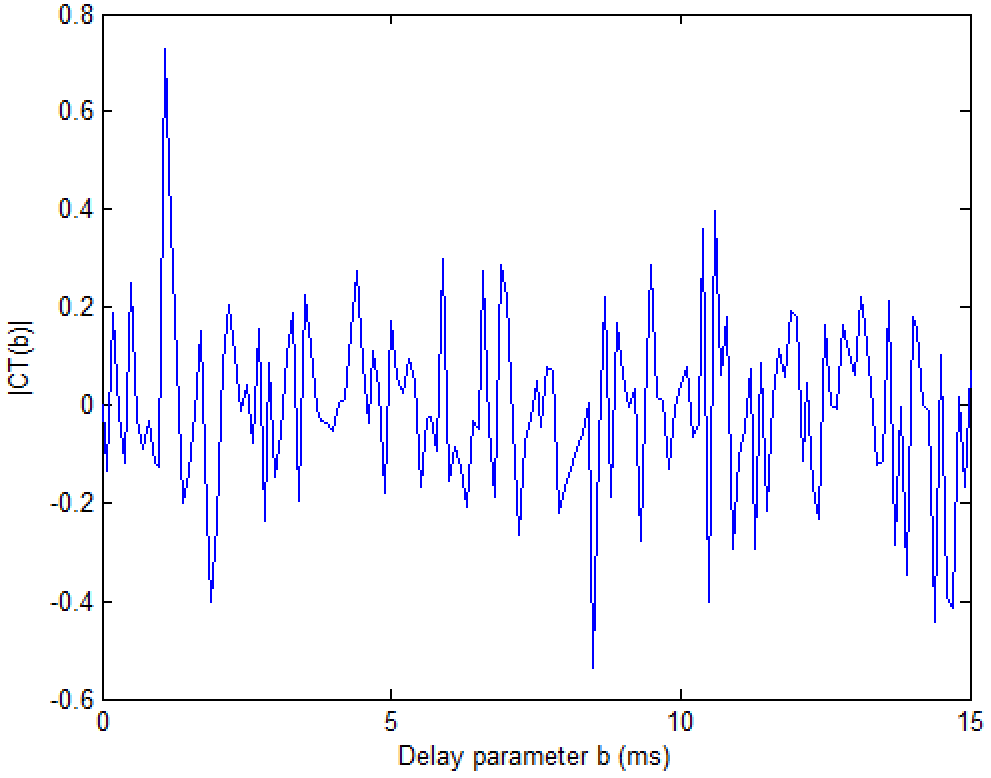


Fig. 3.13: The chirplet transform of the received unsaturated signal.

The first peak occurs at $t = 1.1\,\text{ms}$, indicating correctly the time-of-arrival of $s_1(t)$, the signal that travels the shorter distance. Due to the transmitted signal being scaled down by a factor of 0.01, the signal that reaches the farther of the two microphones, $s_2(t)$ is not of a level high enough for its time-of-arrival to be detected by the chirplet transform algorithm.

The results above indicate that saturation is not a grave issue when it comes to the working of the chirplet signal decomposition algorithm. Despite there being some saturation at the receiver, the times-of-arrival could be accurately detected by the chirplet decomposition algorithm. In trying to reduce saturation, the weaker signal $s_2(t)$ became so weak that the chirplet decomposition algorithm was unable to find its time-of-arrival. In general, it is true that the stronger signal component $s_1(t)$ can be identified more accurately than the weaker signal which in this case is $s_2(t)$. A recurring theme throughout the work is that a lower SNR or a poorer magnitude implies more uncertainty about the estimate. This is best expressed by the Cramer-Rao bound, fleshed out and analyzed in more detail in Chapter 5.

3.4. Shortcomings

The chirplet signal decomposition algorithm presented in this chapter is not original and, in essence, simply serves as a verification of the method presented by Demirli and Saniie [34]. However, explicit assumptions need to be made about the nature of the signal as a prelude to generating the kernel function required to locate the echoes. In the case of OFDM signals, it is difficult or nearly impossible to create a kernel that would effectively decompose any given random OFDM signal. The chirp signal is essentially a single waveform whereas an OFDM signal is a sum of several sinusoids. The chirplet decomposition algorithm works similar to a matched filter in that the received signal is correlated with a kernel signal that is "matched" to the transmitted waveform. Since the OFDM signal consists of multiple waveforms, it is impossible to arrive at a single kernel that would work for any arbitrary OFDM burst. The algorithm was tried because of the supposed resemblance between a chirp signal and an OFDM burst but it was found on closer inspection that this supposition was less than true. Would multiple kernels matching each of the sinusoids in the OFDM burst work? What about decomposing an OFDM burst representing a single bit? These are interesting and open questions that are outside the scope of the present work.

Chirp signals, as mentioned in Section 3.1, are similar to OFDM signals but, ultimately, the chirp signals that were experimented with do not carry any data or information and hence, are uninteresting from an application-level standpoint.

Due to the difficulty in arriving at a generalized kernel that carries the ability to decompose any OFDM signal and the lack of information present in a chirp signal, the search for an effective method to estimate sub-sample delays in a given received burst without making too many *a priori* assumptions continues.

4. Pattern-based correlation algorithm for TDE

In the previous chapter, the chirplet signal decomposition algorithm was discussed. The chirplet signal decomposition algorithm was successfully able to estimate the delay of arrival of chirp signals but depended on the chirplet kernel function for delay estimation. Due to the random nature of the OFDM bursts that are desired to be used for the source localization problem, it is difficult to find a kernel function that would be able to decompose any given OFDM burst into its composite echoes.

In this chapter, a pattern-based correlation method is presented that relies on the fact that when two identical discrete-time signals separated by a definite integer shift, say $s_1[n]$ and $s_1[n-D]$, are superimposed, then some characteristics of the signal $s_1[n]$, for $n<D$, would be similar to characteristics of the summed, or composite signal. Consider the following model:

$$m[n]=s_1[n]+c_1 s_1[n-D] \tag{4.1}$$

where $s_1[n]$ is a finite-duration, causal, real-valued, discrete-time sequence of length *L* such that $s_1[n]$ is strictly zero outside an interval of *L* samples, and *D* is a positive integer. Since $s_1[n]$ is causal, the relation $s_1[n]=0$, for $n<0$ holds. So, when $s_1[n]$ is shifted by *D* samples, then the first *D*-1 samples of $s_1[n-D]$ are zero. Putting these two observations together,

$$m[n]=\left\{\begin{matrix} s_1[n], & n<D \\ s_1[n]+s_1[n-D], & n\geq D \end{matrix}\right\} \tag{4.2}$$

From (4.2) it is clear that $s_1[n]$ appears in two places, when $n<D$ as well as when $n\geq D$. This repetition of the earlier part of the mixture in the latter part of the mixture is best captured by a correlation function. The method presented in this chapter consists of isolating a sub-sequence of $m[n]$ of length smaller than *D* and finding the correlation between this sub-sequence and $m[n]$.

The next section presents a review of the correlation-based TDE methods proposed in the literature and how they relate to the problem in question.

4.1. Review of Correlation-based methods

The use of auto-correlation or cross-correlation functions to estimate the time delay of arrival (TDOA) is not new. Carter [35] proposes a coherence function and shows how it can be used for time-delay estimation (TDE). The coherence function between two wide-sense stationary processes $x(t,\omega)$ and $y(t,\omega)$, where t represents time and ω the outcome of an experiment, is defined as the ratio of the cross-spectral density $G_{xy}(f)$ and the square root of the product of the individual auto power spectra. Carter discusses deriving a maximum-likelihood (ML) estimator of the delay based on the coherence function but assumes Gaussianity of the processes in doing so. The assumption of Gaussianity of the signals renders it inapplicable to the case when these signals are information-bearing bursts since information cannot be assumed to obey a specific distribution.

One of the problems that Ianniello [36] solves is what he calls the "two-path resolution problem." The transmitted signal in propagating through the channel experiences multipath and it is assumed that there are only two paths between the source and the receiver. The delay between the signals traveling on the two paths can be estimated by isolating the peaks of the auto-correlation function of the received signal. Assuming the delay is resolvable, Ianniello says there will be a peak in the auto-correlation function at the time corresponding to the delay caused by traveling along different paths. For the author's method to work, the transmitted signal needs to be Gaussian distributed and the delay between received signal components must be less than the inverse of the bandwidth of the transmitted signal. So, in order for the method to successfully resolve small delays, the signal transmitted from the source must be broadband. The OFDM information-bearing bursts that are of interest in this thesis are in actuality acoustic-range signals with most of the signal content in the range 3.5-5.25 kHz. While the OFDM signal is indeed broadband, it is not clear whether the amplitude levels in the transmitted OFDM pulse are Gaussian distributed.

Varma et al. [37] propose a cross-correlation based TDE method to estimate direction of arrival (DOA) of signals received in a microphone array. Varma proposes a TDE method that uses the phase transform (PHAT) on generalized cross-correlation estimates between signals received from a pair of microphones in the microphone array. The method, like the method proposed in [36], rests on the availability of two independent versions of the transmitted signal. As (4.1) makes clear, only one mixture consisting of two signal components is to be used for TDE. Hence, it is not possible to directly apply the method described in [37] to the problem we are interested in solving.

4.2. Description of the algorithm

4.2.1 Mathematical derivation

The $s_1[n]$ in (4.1) is simulated as a burst consisting of three OFDM pulses, separated from one another by 200 zeroes. The manner of generation is described in Fig. 5.2 but is repeated here in equation form for convenience:

$$o_1[n] = \operatorname{Re}\left(\left(\frac{1}{N}\sum_{k=0}^{N-1} b[k] e^{\frac{j2\pi kn}{N}}\right) e^{j\omega_b n}\right), n = 0,1,\ldots N-1 \tag{4.3}$$

where $b[n]$ is a BPSK-modulated bit-sequence, N is the size of the IFFT, and ω_b is the modulating frequency that converts the OFDM pulse from baseband to bandpass. In generating $o_1[n]$, BPSK modulation was applied to the incoming bit stream to generate $b[n]$, the size of the IFFT was chosen to be 512, and the modulating frequency ω_b was 1 rad/s. The signal $s_1[n]$ was generated in the same manner as described in Section 5.3.1.

The received mixture $m[n]$ is modeled as a sum of un-delayed $s_1[n]$ and a delayed and attenuated $s_1[n]$, as shown in (4.1). The approach to estimating *D* consists of finding the cross-correlation between $m[n]$ and a sub-sequence consisting of the first *S* samples of $m[n]$ and identifying the peaks of the cross-correlation function. Let the sub-sequence $v[n]$ be defined as:

$$v[n] = \begin{Bmatrix} m[n], 0 \le n \le S-1 \\ 0 \quad\quad \text{, otherwise} \end{Bmatrix} \tag{4.4}$$

S is chosen such that it is less than the integer delay *D,* measured in samples. Some prior information about the experimental arrangement is used in deciding the value of *S*. Another way of looking at this method is to note that the minimum delay that can be resolved is lower-bounded by the size of the window.

Recall that, in Section 4.1, mention was made that $s_1[n]$ is a causal and finite-length signal of length *L* samples. The cross-correlation function essentially consists of a convolution of time-inverted $v[n]$ and $m[n]$. The sample cross-correlation function is formally defined as follows:

$$R_{mv}[n]=\sum_{k=0}^{L-1} m[k]v[k-n] \tag{4.5}$$

The MATLAB command *fliplr* is used to time-reverse the window $v[n]$ and the mixture $m[n]$ is passed through a filter whose impulse response is $v[-n]$. The filtering process is accomplished through MATLAB's *filter* command.

4.2.2 Simulation examples

The signal $s_1[n]$ was generated in MATLAB in the manner explained in Section 4.2.1, as a succession of three OFDM pulses, interspersed by 200 zeros. A plot of $s_1[n]$ is shown in Fig. 4.1.

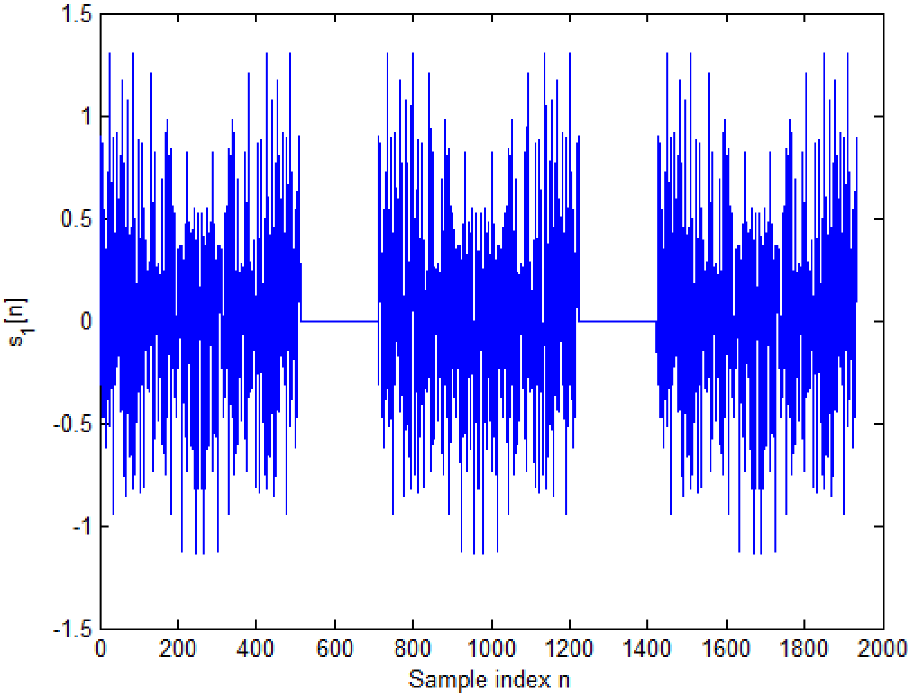


Fig. 4.1: Simulated OFDM burst consisting of three pulses.

Two illustrative examples follow:

Example 1: *D* = 60 *samples*, *X* = 20 *samples*, $c_1 = 0.3$

A delay of 60 samples was applied to $s_1[n]$ and the delayed $s_1[n]$ was summed to $s_1[n]$. The expression for $m[n]$ appears below:

$$m[n] = s_1[n] + 0.3s_1[n-60] \tag{4.6}$$

The size of the sub-sequence $v[n]$ was chosen to be 20 samples long and so $v[n]$ consisted of the first 20 samples of $m[n]$. The cross-correlation function $R_{mv}[n]$, defined in (4.5), is shown in Fig. 4.2 below :

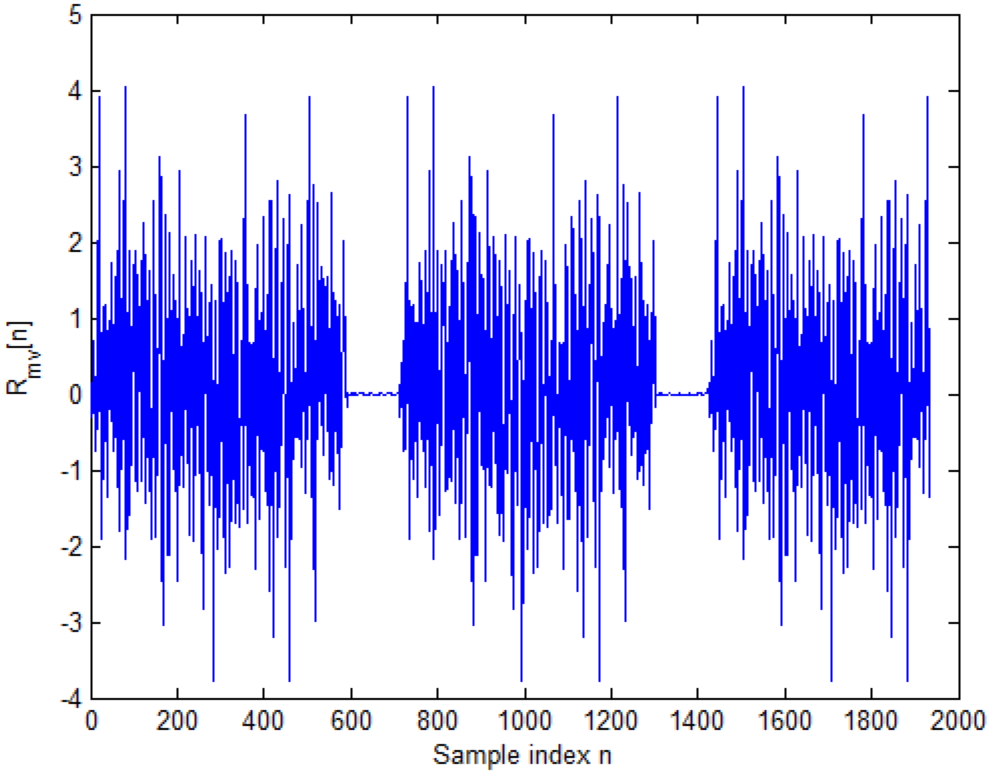


Fig. 4.2: Pattern-based correlation between mixture and sub-sequence .

Two peaks appear in the early portion of each pulse, at $n = 19$ and $n = 79$ in the first pulse, at $n = 731$ and $n = 791$ in the second pulse, and at $n = 1443$ and $n = 1503$ in the third pulse. A zoomed version of Fig. 4.2 appears below in Fig. 4.3 concentrating on $0 \le n \le 100$:

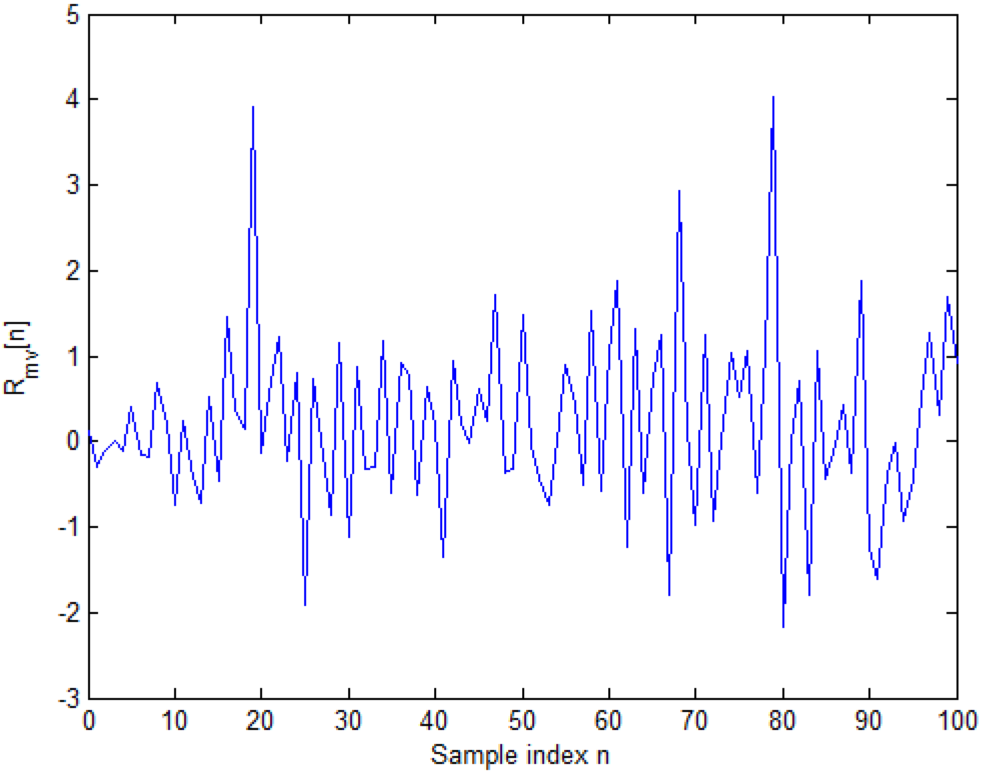


Fig. 4.3: Zoomed version of Fig. 4.2.

Two peaks occur, one at $n=19$ and one at $n=79$. The separation between the peaks is $D=60$ samples. The MATLAB function *findpeaks* was used to find the peaks of the cross-correlation function $R_{mv}[n]$ and the shift between these peaks was taken as the delay.

Another example follows with a smaller window size and the same delay and attenuation coefficient.

Example 2: $D=60$ samples, $X=10$ samples, $c_1=0.3$

The size of the window was reduced to 10 samples and the experiment was repeated with the parameters D and c_1 fixed at 60 samples and 0.3 respectively . The plot of the cross-correlation function is shown below in Fig. 4.4:

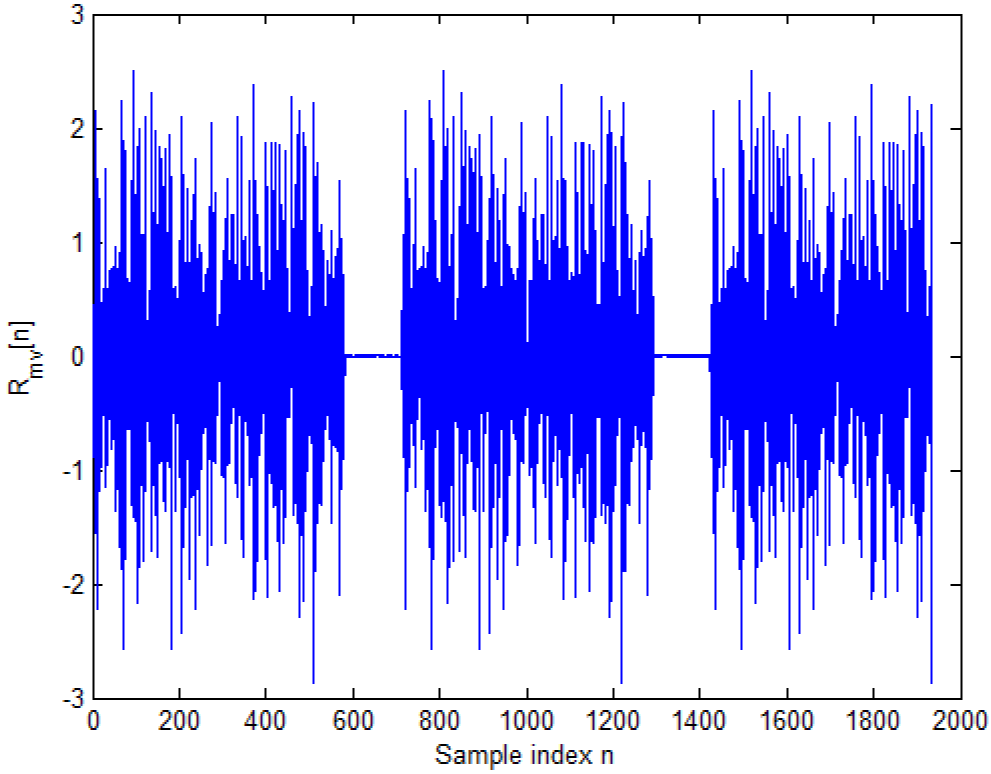


Fig. 4.4: Pattern-based correlation with reduced sub-sequence size of $X = 10$ samples.

It appears that the peaks which were prominent earlier have disappeared. A zoomed-in version of the above figure is shown in Fig. 4.5 focusing on the neighborhood of $0 \le n \le 140$.

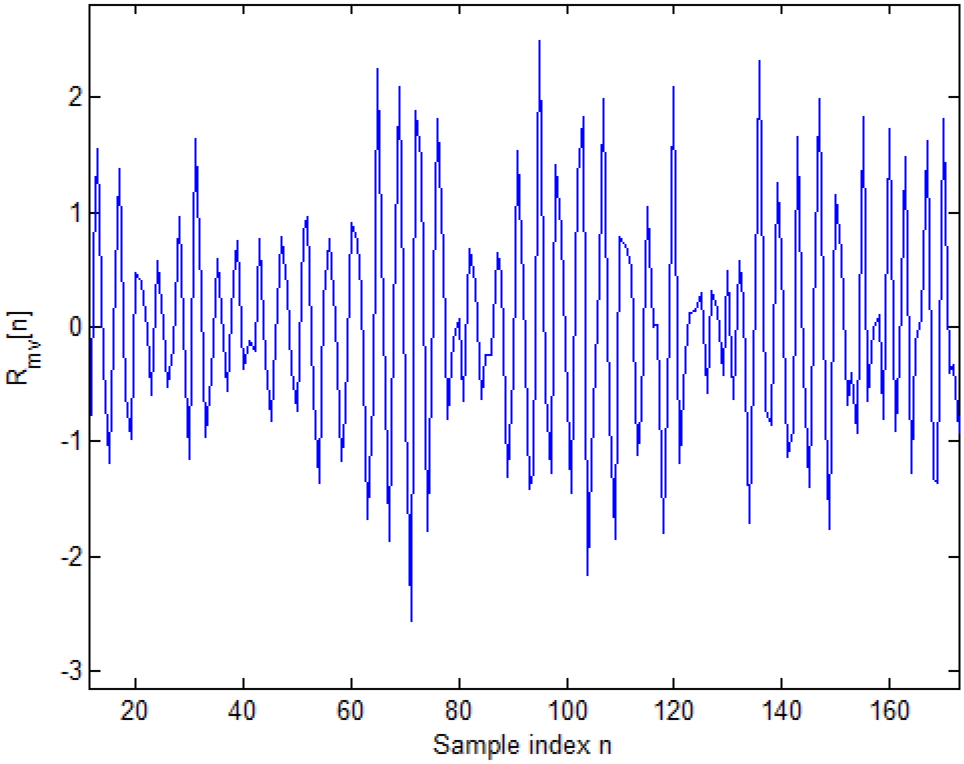


Fig. 4.5 : Zoomed in version of Fig. 4.4.

Peaks are expected at $n=19$ and $n=79$ but this is not what is observed in Fig. 4.5. Instead a local maximum is observed at $n=95$ which is unrelated to anything that is applied to the sequences. This has arisen out of choosing too small a sub-sequence. The cross-correlation function does not exhibit peaks at the right place because the sub-sequence is not representative enough of the patterns in the signal. In the extreme case of $S=1$ the cross-correlation function $R_{mv}[n]$ would be $m[n]$ itself and would contain as much information about the echoes present in $m[n]$ as $m[n]$ itself – which would not be very much, usually.

4.3. Discussion of limitations

The examples treated so far have not included the case when the delay between $s_1[n]$ and $s_2[n]$ consists of a fractional quantity. In a practical scenario, this is a potentially crippling shortcoming. The delay between the signals reaching the two microphones from the single source is a continuous variable τ that can take on any value. Converting the time-delay into delay in terms of samples is done by multiplying it with the sampling frequency F_s. The product τF_s is not necessarily an integer with no fractional part. The inability of the correlation-based algorithm to estimate sub-sample delays, or delays that are not integer multiples of the sampling period, renders it unsuitable for the problem at hand. Nevertheless, in order to make comparisons between this method and the method presented in Chapter 5 possible, the performance of the correlation-based delay estimation algorithm on recordings collected in the DSPRL appears in Section 4.4.

The algorithm considered in this chapter is an important milestone towards the ultimate goal of resolving sub-sample delays in information-bearing OFDM signals. The pattern-based correlation method proves that it is possible to estimate integer delays in a single sensor case where only one version of the mixture is available and consequently there is no *a priori* information about the nature and characteristics of the source signal.

$v[n]$ is a sub-sequence consisting of the first few samples of the received burst. Ideally, $v[n]$ should neither be too big nor too small. The sub-sequence should not be so big that it contains the echo but must contain enough information for the results to be meaningful. Most importantly, the sub-sequence $v[n]$ is a sub-sequence of the signal received and must not contain only noise. In practical situations, it is difficult to determine the onset of the transmitted burst at the receiver. More precisely, the point of transition from noise to information is challenging to pinpoint. Usually, characteristics gleaned from repeated observations could be used prior to the execution of the algorithm in order to ensure that the sub-sequence under consideration does not contain a preponderance of noise samples.

4.4. Performance of algorithm

Case 1: $\hat{D}_{theory} = 56.86$ samples**.**

Experiments conducted in the DSPRL involved transmitting a burst consisting of four OFDM bursts interspersed with zeros. The burst was generated in MATLAB and that signal before transmission is shown in Fig. 4.6.

Recording was performed on both channels and consisted of running MATLAB's *wavrecord* function for a duration of six seconds. The sampling frequency F_s was 44.1 kHz. The received signal is shown in Fig. 4.7.

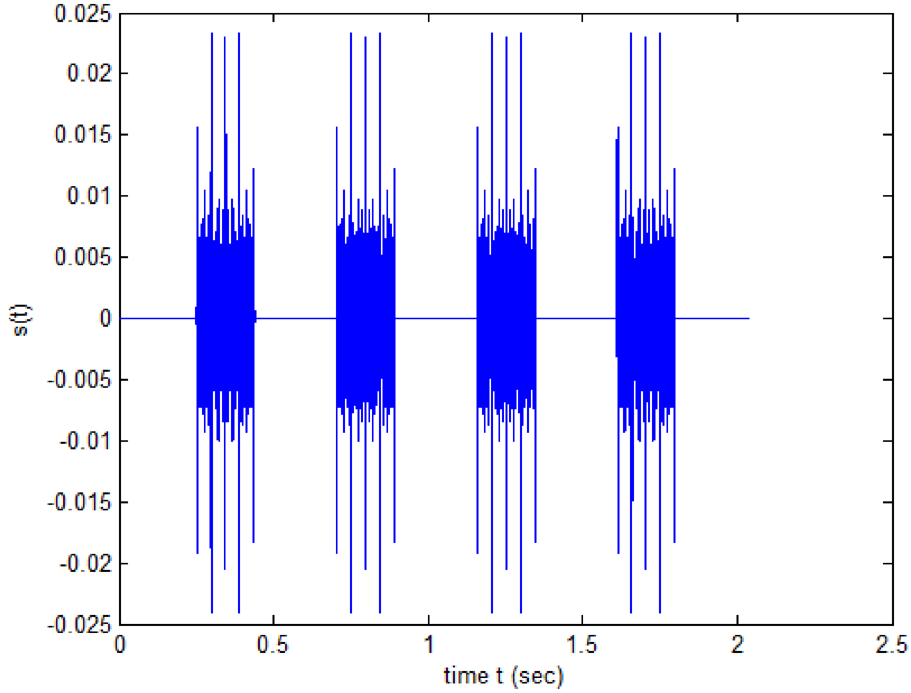


Fig. 4.6: Transmitted signal $s(t)$ generated in MATLAB.

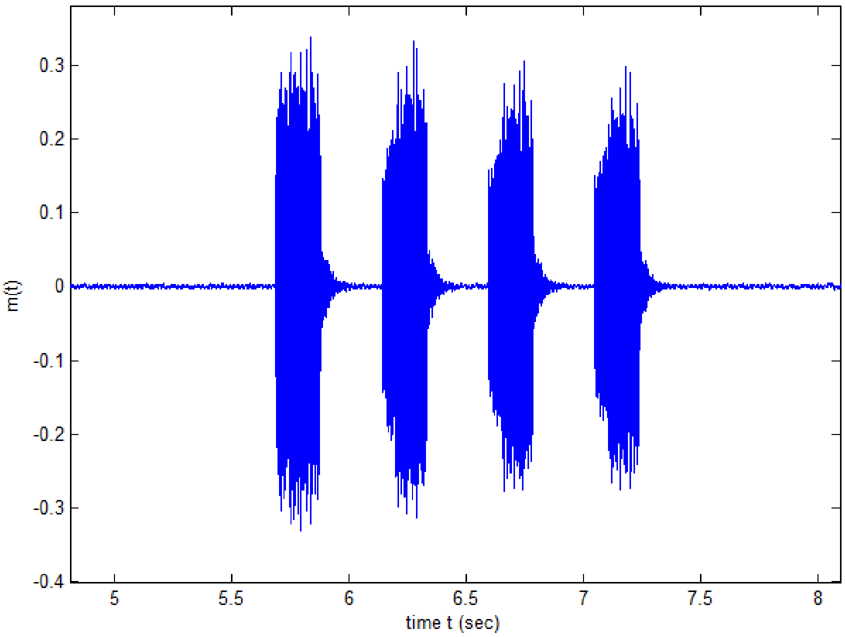


Fig. 4.7: Received burst $m(t)$.

The signal-to-noise ratio was calculated as a ratio of the variances of the signal received and that of the noise. Four distinct pulses may be identified in Fig. 4.7 and the variance of the signal was calculated by isolating these pulses and taking the mean of the variances of these 4 pulses. The variance of the pulses were $147 \text{ x } 10^{-4}$, $72 \text{ x } 10^{-4}$, $98 \text{ x } 10^{-4}$, and $97 \text{ x } 10^{-4}$. The average variance taken over these four pulses is $101 \text{ x } 10^{-4}$. The variance of the noise, isolated by considering the samples between the pulses in the received burst, in Fig. 4.7 is $1.8156 \text{ x } 10^{-6}$. The ambient SNR can now be calculated according to (4.7) as:

$$SNR_{dB} = 10\log_{10}\left(\frac{101 \text{ x } 10^{-4}}{1.8156 \text{ x } 10^{-6}}\right) = 37.303 \text{ dB} \tag{4.7}$$

The two microphones were placed at distances of 0.80 m and 0.36 m from the source. The time shift between the signals traveling to the two microphones is calculated as:

$$\hat{D}_{theory} = \left(\frac{0.80 \text{ m} - 0.36 \text{ m}}{341.2 \text{ m/s}}\right) * 44.1 \text{ kHz} = 56.86 \text{ samples} \tag{4.8}$$

The most important step in the correlation-based TDE algorithm is the first: isolating a big enough sub-sequence of the received burst. In order to do this, a rectangular window must be placed such that it includes as little of the noise that precedes the onset of the burst and as much of the actual signal itself as possible. In order to better understand how this may be done, a zoomed version of Fig. 4.7 is shown in Fig. 4.8 showing the early portion of the burst.

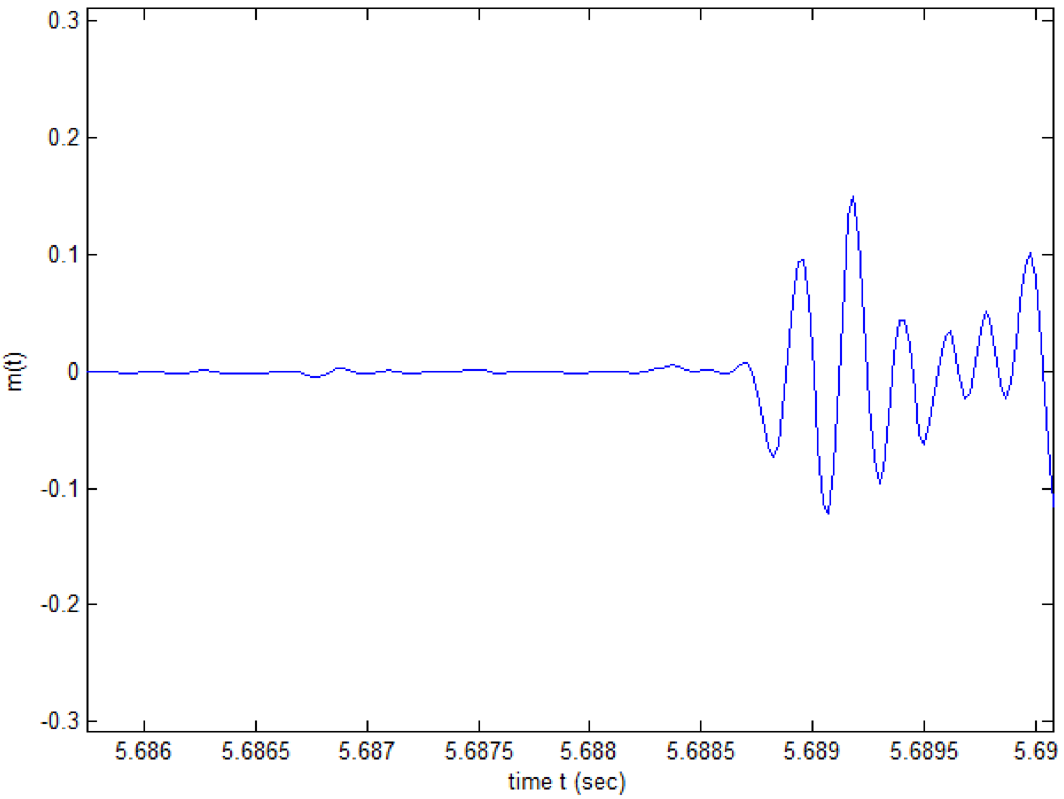


Fig. 4.8: Zoomed version of Fig. 4.7.

We observe that until around $t = 5.688\text{ s}$, $m(t)$ does not register much perturbation. Beginning around $t = 5.6885\text{ s}$, the signal $m(t)$ begins to show some activity and this is evidenced by the relatively high energy in $m(t)$. This suggests a way to pinpoint the exact beginning of the received burst: take a moving rectangular window of appropriate length and calculate the energy of $m(t)$ in each position of that moving window. Wherever the window includes only noise samples, the energy would presumably be lower than when it included genuine signal samples. The transition from those samples in $m(t)$ that are noise to those that contain information, as exemplified by the time instant $t = 5.6885$ s in Fig. 4.8, is isolated by taking the derivative of the windowed energy function. A zoomed version of the windowed energy function $E(t)$ computed with a rectangular window 5 samples long with a 1 sample increment between windows is shown in Fig. 4.9.

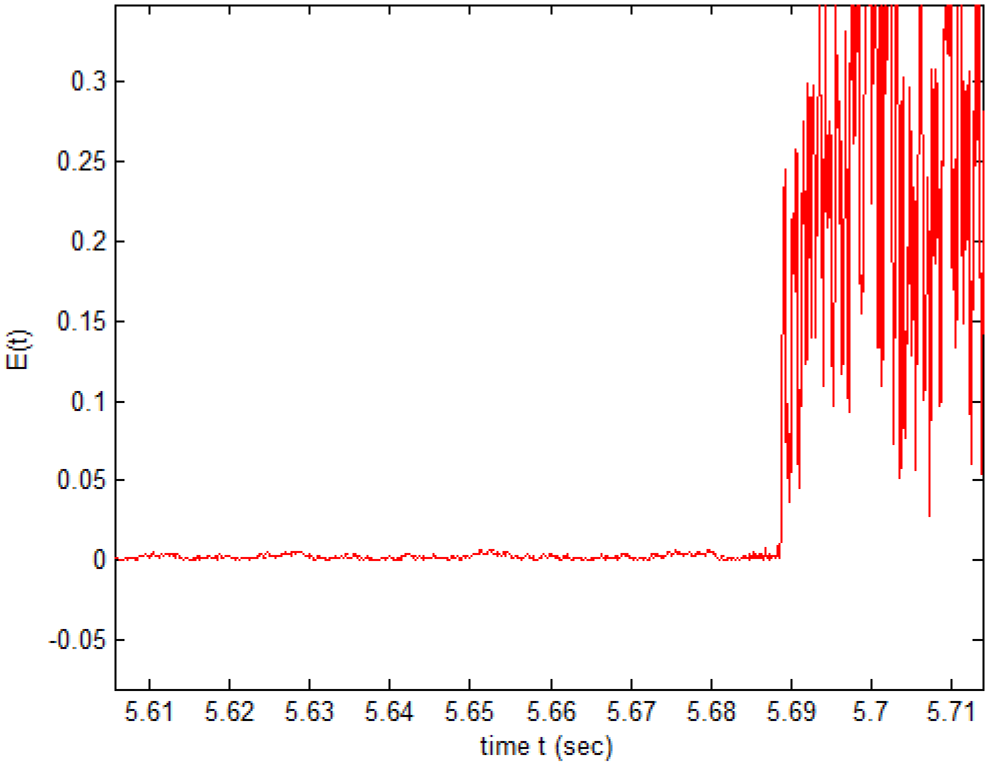


Fig. 4.9: Zoomed version of the windowed energy function $E(t)$.

As seen in the above plot, there is a jump a little before $t = 5.69$ s which roughly corresponds to the point in time in Fig. 4.8 where the signal begins.

The method described above was employed to all the recorded signals and a sub-sequence of length twenty-five including the start was chosen to form $v[n]$. Twenty-five was chosen keeping in mind the true delay $\hat{D}_{true}$ calculated earlier. Recall, it was mentioned in the exposition of the algorithm that the length of the window S has to be less than the true delay in order for the algorithm to work. Four pulses are visible in the received signal of Fig. 4.7. For each of those four pulses, it is possible to obtain a delay estimate. The cross-correlation function was calculated by sampling the received burst shown in Fig. 4.7 and so the x-axis in Fig. 4.10 is marked in samples and not in seconds.

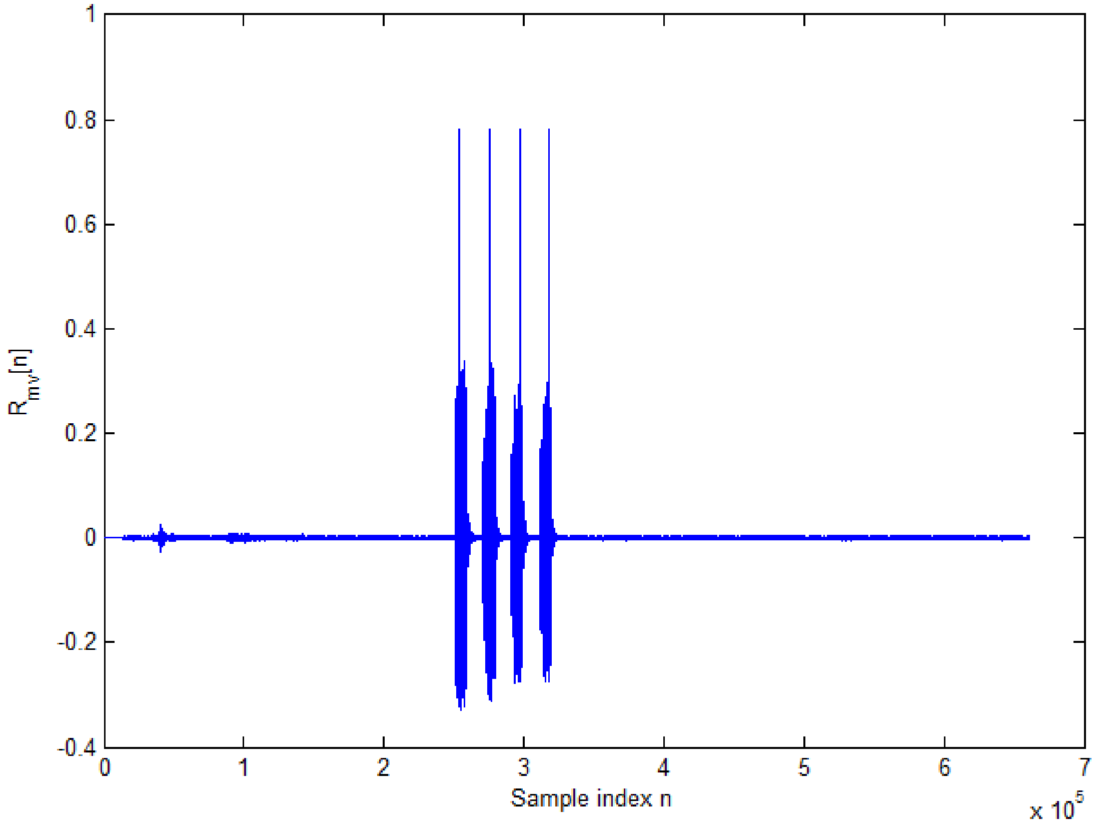


Fig. 4.10: Cross-correlation function $R_{mv}[n]$ for $\hat{D}_{theory} = 56.86$ samples.

A zoomed version of Fig. 4.10 focusing on the first pulse is shown in Fig. 4.11 to better illustrate the delay between the two peaks.

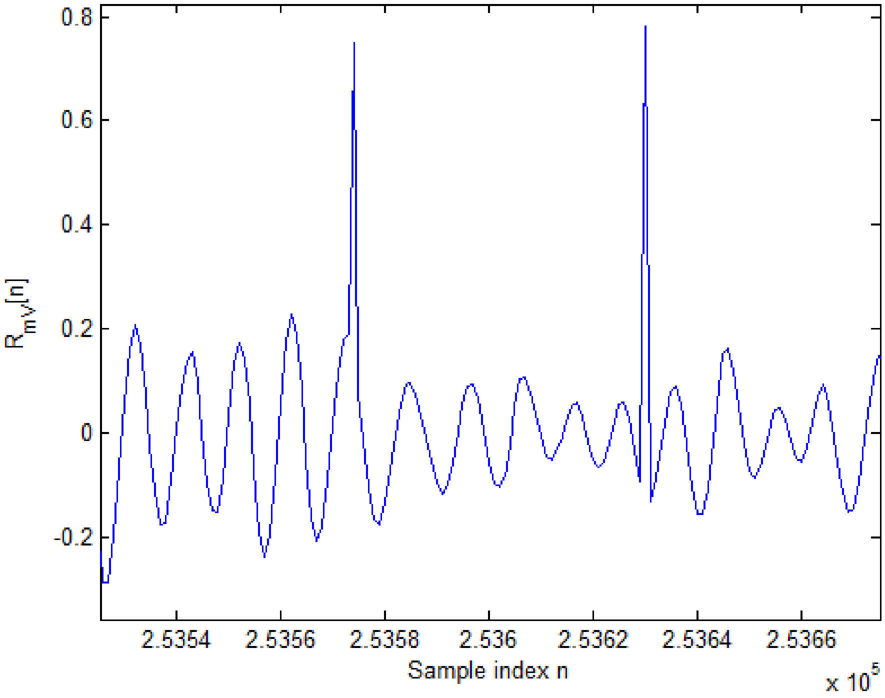


Fig. 4.11: Zoomed version of $R_{mv}\left[n\right]$.

The first peak occurs at a sample index of $n = 253575$ and the second at $n = 253630$. The difference between the two is $n = 55\text{ samples}$. The actual delay is $\hat{D}_{theory} = 56.86\text{ samples}$**.** Not only is the algorithm unable to estimate the fractional portion of the true delay, it is inaccurate in estimating the integer portion as well.

From the four pulses, four delay estimates may be obtained and from the ten recordings, forty delay estimates were obtained for this distance between the microphones. The distribution of these forty delay estimates is shown in Fig. 4.12.

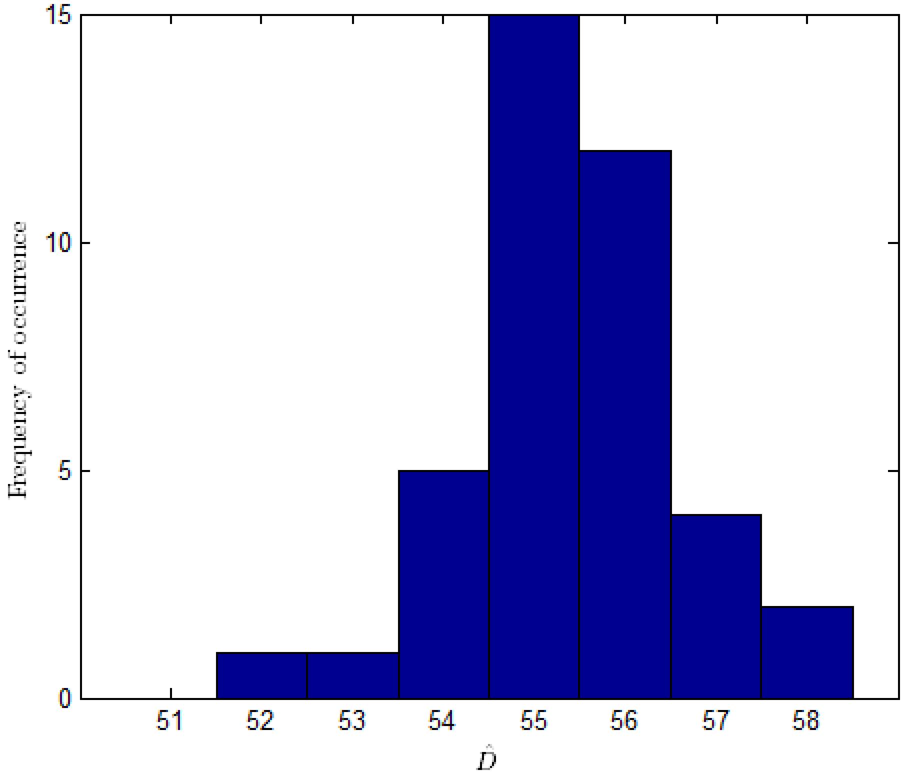


Fig. 4.12: Distribution of delays $\hat{D}$ for $\hat{D}_{theory} = 56.86$.

Case 2: $\hat{D}_{theory} = 123.48$ samples**.**

The microphones were placed at distances of 131 cm and 36 cm from the source. The theoretical delay between the signals traveling on the two paths is:

$$\hat{D}_{theory} = \left(\frac{1.31\text{ m} - 0.80\text{ m}}{341.2\text{ m/s}}\right) * 44.1\text{ x }10^3 = 123.48\text{ samples} \tag{4.9}$$

The procedure described under Case 1 was repeated for this larger distance for the same number of recordings, i.e. 10. The forty delay estimates obtained are shown in Fig. 4.13.

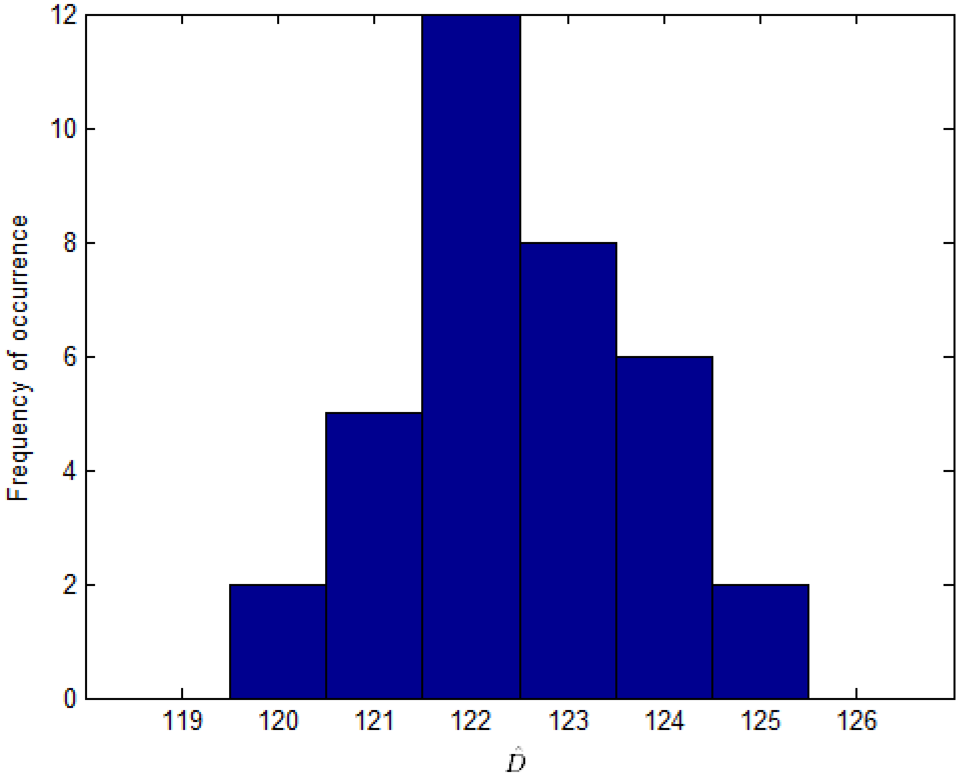


Fig. 4.13: Distribution of delay estimates for $\hat{D}_{theory} = 123.48$ samples.

Two things stand out from even a cursory glance at Fig. 4.12 and Fig. 4.13. In Fig. 4.12, the delay estimates are much more concentrated than they are in Fig. 4.13. As the estimates are farther apart, it is possible to infer greater variance brought about by an increase in the separation between the microphones. Secondly, the mean of the estimator in the two cases is $55.4\ samples$ and $122.48\ samples$ respectively and neither of these means is equal to the true mean at those distances which is 56.86 *samples* and 123.48 *samples* respectively.

The variances of the individual mixtures in the received burst $m(t)$ in Case 2 are $6.25 \text{ x } 10^{-6}$, $6.75 \text{ x } 10^{-6}$, $5.90 \text{ x } 10^{-6}$, $4.33 \text{ x } 10^{-6}$. The mean of these variances is $5.81 \text{ x } 10^{-6}$. The variance of the noise is $0.19 \text{ x } 10^{-6}$. Recalculating the signal-to-noise ratio, we have:

$$SNR_{dB} = 10\log_{10}\left(\frac{5.81 \text{ x } 10^{-6}}{0.19 \text{ x } 10^{-6}}\right) = 14.85 \text{ dB} \tag{4.10}$$

The increase in variance with the increase in distance of the farther microphone from the source is due to the decrease in SNR as demonstrated in (4.10). As the signal to the farther microphone travels a greater distance, it experiences greater attenuation and hence the SNR of the mixture $m(t)$ decreases. The relationship between the variance of the delay estimates σ_D^2 and SNR_{dB} is shown in Fig. 4.14.

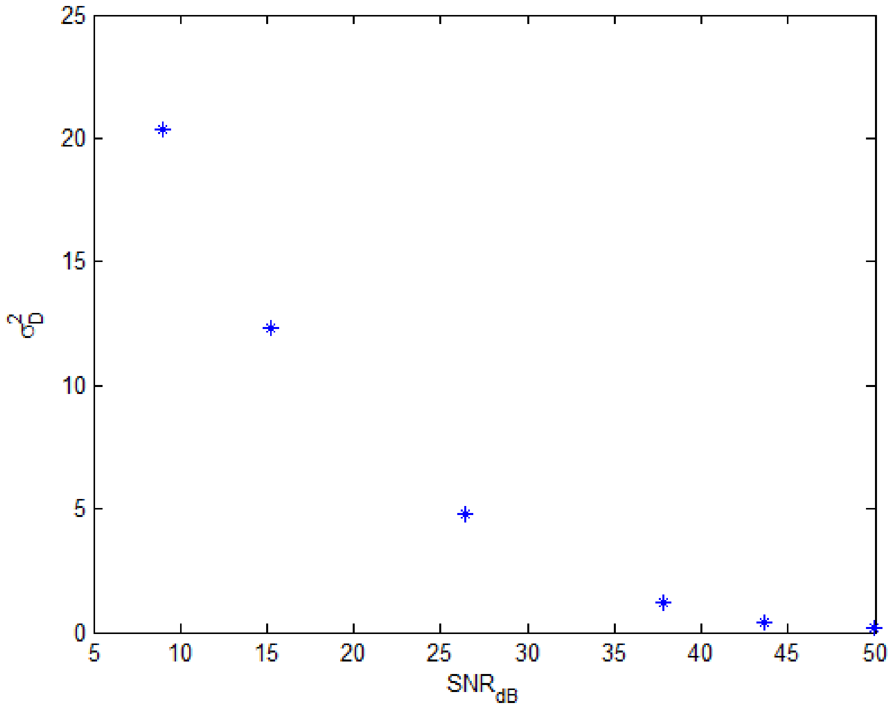


Fig. 4.14: σ_D^2 versus SNR_{dB}.

The behavior of the correlation-based delay estimation algorithm has been studied. It has been shown that the algorithm is unable to estimate the fractional portion of the delay between the signals traveling on the two paths to the microphones $s_1(t)$ and $s_2(t)$. Furthermore, the algorithm is sensitive to the accurate estimation of the beginning of the received signal burst. The estimator is biased and the variance of its estimates increases as the signal-to-noise ratio of the received burst decreases.

The method described in this chapter has two desirable qualities: 1) it is simple and straightforward to comprehend and 2) it does not depend on any assumptions about the nature of the signal such as its statistics. However, it is an important milestone in the journey towards solving the problem of estimating sub-sample delays in that it shows how echoes located at integral delays can be estimated without any *a priori* knowledge of the nature of the source signal.

5. The time-frequency ratio of mixtures (TFRM) algorithm

In this chapter, we will look at a time-frequency approach called Time-Frequency Ratio of Mixtures (TFRM) to resolve sub-sample delays. This method was originally proposed by Abrard and Deville [38] in the context of blind-source separation of signal mixtures. The idea in the TFRM method is to identify single-source areas, i.e. regions in time during which only one source is active, by taking ratios of estimated time-frequency distributions of the mixtures. The TFRM method necessitates no assumptions about the nature of the individual signals such as Gaussianity, independence, or linearity. The TF distributions need to be only slightly different for separation to be achieved. Abrard and Deville report that complete or partial separation is achieved depending on the number of signals N and the number of sets of observations P. The case $N = P = 2$ is considered in this chapter.

5.1 Two mixtures of two sources

5.1.1 Basic Overview

Let us consider the following linear combinations (mixtures) of two real-valued continuous-time signals:

$$\begin{aligned} m_1(t) &= s_1(t) + c_1 s_2(t) \\ m_2(t) &= s_1(t) + c_2 s_2(t) \end{aligned} \tag{5.1}$$

where the coefficients c_1 and c_2 are real, constant, and non-zero. The signal separation algorithm may be seen as a way to estimate c_1 and c_2 and then to perform successive signal cancellations. Using (5.1), we define:

$$\begin{aligned} y_1(t) &\triangleq m_1(t) - \frac{\hat{c}_1}{\hat{c}_2} m_2(t) \\ &= \left(1 - \frac{\hat{c}_1}{\hat{c}_2}\right) s_1(t) + \left(c_1 - \frac{\hat{c}_1}{\hat{c}_2} c_2\right) s_2(t) \end{aligned} \tag{5.2}$$

where $\hat{c}_1$ and $\hat{c}_2$ are estimates of c_1 and c_2. From $y_1(t)$ in (5.2), provided $\frac{\hat{c}_1}{\hat{c}_2} \cong \frac{c_1}{c_2}$, we have

$$\hat{s}_1(t) = \frac{y_1(t)}{1 - \frac{\hat{c}_1}{\hat{c}_2}} \tag{5.3}$$

where $\hat{s}_1(t)$ is an estimate of $s_1(t)$. Having estimated $s_1(t)$ it is possible to arrive at a scaled version of $s_2(t)$, the second signal, by subtracting $s_1(t)$ from either $m_1(t)$ or $m_2(t)$ in (5.1). Picking $m_1(t)$, we find,

$$\hat{c}_1\hat{s}_2(t) = m_1(t) - \hat{s}_1(t) \tag{5.4}$$

where $\hat{s}_2(t)$ is an estimate of $s_2(t)$. Although the signals transmitted are continuous-time signals, in actuality the mixtures $m_1(t)$ and $m_2(t)$ are sampled at the receiver before they are processed, and hence they are discrete-time. Replacing $m_1(t)$, $m_2(t)$, and $y_1(t)$ with $m_1[n]$, $m_2[n]$, and $y_1[n]$ respectively, (5.3) and (5.4) may be rewritten as:

$$\hat{s}_1[n] = \frac{y_1[n]}{1 - \frac{\hat{c}_1}{\hat{c}_2}} \tag{5.5}$$

and

$$\hat{c}_1\hat{s}_2[n] = m_1[n] - \hat{s}_1[n] \tag{5.6}$$

The localization problem of interest is one wherein a single source emits a signal that impinges on two microphones separated by some distance. Under appropriate circumstances, it may be practically justified to consider the sequence $s_2[n]$ to be simply a delayed version of $s_1[n]$. The argument goes as follows. The signal $s_2(t)$ does not exist as a second source signal but is in fact $s_1(t)$ passing through a separate path. In light of this approximation, (5.4) can be rewritten as:

$$\hat{c}_1\hat{s}_2(t) = \beta s_1(t-\tau) \tag{5.7}$$

and (5.6) as

$$\hat{c}_1\hat{s}_2[n] = \beta s_1[n-D] \tag{5.8}$$

$s_2[n]$ is thus a delayed and attenuated version of $s_1[n]$, as can be seen in (5.7) and (5.8).

Modeling the paths as linear and time-invariant filters, with impulse responses $h_1(t)$ and $h_2(t)$, we have,

$$\begin{aligned} m_1(t) &= s_1(t)*h_1(t)+c_1s_1(t)*h_2(t) \\ m_2(t) &= s_1(t)*h_1(t)+c_2s_1(t)*h_2(t) \end{aligned} \tag{5.9}$$

where $s_2(t) = s_1(t)*h_2(t)$. For now, throughout the remainder of the chapter, the assumption is made that there are no nearby reflecting surfaces and so, the experiments are conducted practically in free-space. When the free-space assumption is almost valid, the paths can be approximated by pure delay filters, leading to $h_1(t) = \delta(t)$ and $h_2(t) = \delta(t-\tau)$. Considering the delay between $s_1(t)$ and $s_2(t)$ to be τ, the equivalent discrete-time delay may be written in terms of the sampling frequency F_s and τ as

$$D_{true} = \tau F_s \tag{5.10}$$

Bearing this in mind, an iterative procedure is employed wherein the sequence $s_1[n]$ is successively scaled by a factor β and delayed by D samples (where D is generally not an integer). The following error function $e(\beta, D)$ may be constructed, based on (5.8),

$$e(\beta,D) = \left[\hat{c}_1\hat{s}_2[n] - \beta\hat{s}_1[n-D]\right]^T \left[\hat{c}_1\hat{s}_2[n] - \beta\hat{s}_1[n-D]\right] \tag{5.11}$$

The error function, in the absence of noise and when the two mixtures are not exactly identical, has a minimum at exactly $\beta = \hat{c}_1$ and $D = D_{true}$. The TFRM method is not always exact so, consistent with our notation, the delay estimate returned by the TFRM method is $\hat{D}$. It is important to distinguish between two kinds of microphone characterizations: the one that is closer to the source and the one that is farther from the source. The signal that reaches the microphone Rx_1 is designated as $s_1[n]$ and the signal that reaches the microphone Rx_2 is designated as $s_2[n]$. When Rx_1 is closer to the source, $s_1[n]$ travels a smaller distance, and thus for less time, than $s_2[n]$ and this results in two things: 1) $s_2[n]$ is delayed with respect to $s_1[n]$ and 2) $s_2[n]$ is attenuated compared to $s_1[n]$ by a factor c, where $c < 1$. When Rx_2 is closer to the source, $s_1[n]$ is delayed with respect to $s_2[n]$ and so the delay τ between $s_1[n]$ and $s_2[n]$ is negative. Since $s_1[n]$ is attenuated with respect to $s_2[n]$, the factor β is positive. The factor c takes two independent values c_1 and c_2, reflecting independent realizations of the experiment $m_1(t)$ and $m_2(t)$. For any two independent realizations of the recording experiment, the attenuation factor c for both the realizations can be either equal or not equal. If the attenuation factor c for the two mixtures is not the same, then it follows that one is less than the other. The lesser factor is taken as c_1 and the greater factor is chosen as c_2. In case the two factors are equal, in the case of no noise, the TFRM method, as will be shown in Section 5.4.1, fails. β is a factor that is used to search for c_1 and as long as $\tau > 0$, $\beta \in (0,1)$. In case $\tau < 0$, the two estimated signals could be swapped and the search is performed over the same interval for β. When $\tau > 0$, the upper and lower bound on β narrows the search since $\beta > 1$ is not of interest to us. When $\tau > 0$, however, the search is carried out over a larger interval where β is known to be positive and greater than 1. The value of the true shift D_{true}, measured in samples, relates to the separation between the microphones, the speed of sound in the medium in which the experiment is conducted, and the angle of arrival of the source signal with respect to the normal to the line connecting the microphones. The sampled mixtures $m_1[n]$ and $m_2[n]$ are finite-duration bursts, and their length, measured in samples, is the number of samples outside of which the signal is identically equal to zero and since the recording period is the same for the two mixtures, their length is denoted by $\mathrm{L}_{\mathrm{sum}}$. Figure 5.1 illustrates the parameter $\mathrm{L}_{\mathrm{sum}}$.

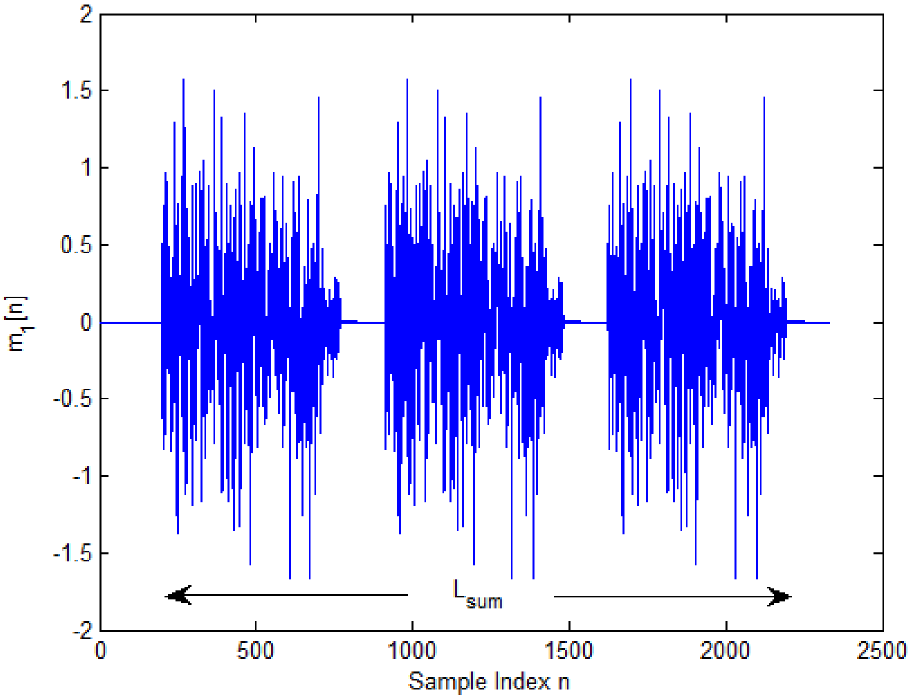


Fig. 5.1: Illustration of length of received mixture.

For a given distance of separation between the recording microphones, the maximum time-difference-of-arrival is limited by the time it takes sound to travel the distance of separation between the microphones. Therefore, the upper bound for D_{true} is known.

The performance of this algorithm depends on the accurate estimation of the ratio $\rho = \frac{\hat{c}_1}{\hat{c}_2}$. The individual constants c_1, c_2 themselves are not estimated but ρ is directly estimated and is sufficient to obtain a scaled version of $s_2[n]$ as shown in (5.6). The scaling factor c_1 is estimated via a search as shown in (5.11). The next section deals with the estimation of ρ.

5.2 Estimation of ρ

5.2.1 A preliminary temporal approach

A simple Blind-Source separation (BSS) method may be derived by applying the following principle: if some regions in time may be found during which only one of the mixing signals is active, then the ratio ρ may be calculated by simply dividing the samples of the two mixtures in those regions of time.

Consider any instant of time t_n when only the signal $s_2\left[n\right]$ is active. Then, from (5.1) we have

$$\begin{aligned} m_1\left[t_n\right] &= c_1 s_2\left[t_n\right] \\ m_2\left[t_n\right] &= c_2 s_2\left[t_n\right] \end{aligned} \tag{5.12}$$

and therefore $\rho = \dfrac{m_1\left[t_n\right]}{m_2\left[t_n\right]}$.

Unfortunately, such times t_n are difficult to determine in practical situations since usually both sources are simultaneously active. The temporal approach is restricted therefore to the very special case when each signal occurs alone in large enough time intervals. The temporal approach is susceptible to inaccuracy and in general it is difficult to isolate beforehand regions in which only a single signal is active. The next section looks at a time-frequency (TF) approach that does not require the sources to be temporally distinct over large time intervals.

5.2.2 Time-Frequency Analysis

Many powerful time-frequency (TF) methods have been developed but the simplest one is the short-time Fourier transform (STFT) [39]. Each mixture $m_i\left(t\right)$ is multiplied by a shifted real-valued window function, $w\left(t-t_0\right)$ centered at t_0, which produces the windowed signal $x\left(t,t_0\right) = m_i\left(t\right) w\left(t-t_0\right)$. The STFT of $m_i\left(t\right)$, which is centered at t_0, is

$$M_i\left(t_0,\Omega\right)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} m_i\left(t\right)w\left(t-t_0\right)e^{-j\Omega t}dt \tag{5.13}$$

$M_i\left(t_0,\Omega\right)$, for a given time t_0 and a given angular frequency Ω, is a single point in the time-frequency domain, the contribution of the windowed signal $x\left(t,t_0\right)$. Since computations and estimation are in the discrete-time domain, it would be more appropriate to rewrite the *discrete-time* STFT for discrete-time signals. Substituting $t=nT_s$, and restricting t_0 to integral multiples of the sampling period T_s, so that $t_0=mT_s$, yields:

$$M_i\left(mT_s,\Omega\right)=\frac{1}{\sqrt{2\pi}}\sum_{n=-\infty}^{\infty} m_i\left[nT_s\right]w\left[nT_s-mT_s\right]e^{-j\Omega nT_s}\Delta t \tag{5.14}$$

The variable ΩT_s can be written as $\frac{2\pi f}{f_s}$, and so is a continuous variable periodic with period 2π. The individual samples are separated by T_s so $\Delta t=T_s$. Replacing ΩT_s by the normalized angular frequency variable ω, and substituting $\Delta t=T_s$ we obtain,

$$M_i\left(m,\omega\right)=\frac{1}{\sqrt{2\pi}}\sum_{n=-\infty}^{\infty} m_i\left[n\right]w\left[n-m\right]e^{-j\omega n}T_s \tag{5.15}$$

Just as the discrete-time Fourier transform (DTFT) is continuous in the frequency domain and the discrete-time Fourier transform (DFT) is discrete in both the time and the frequency domain, by analogy of the relation governing the DFT and DTFT, the *discrete* STFT $M_i(m,\omega_k)$ can be rewritten as a DFT (or FFT) computation:

$$M_i\left(m,\omega_k\right)=M_i\left(m,\omega\right)\Big|_{\omega=\omega_k=\frac{2\pi}{F}k},k=0,1,...F-1 \tag{5.16}$$

where ω_k is a discrete angular frequency variable. The method described by Abrard and Deville [38] rests on three assumptions for correct operation, as listed next:

Assumption 1. (i) The constants $c_1,c_2\neq 0$

(ii) The power of each signal is non-negligible at least at some times t.

Assumption 1(i) is necessary to avoid a situation where both mixtures have contributions from only one signal. Physically, this could represent a scenario wherein a large obstacle is present in the path of one of the signals or one of the paths presents a highly variable attenuation characteristic leading to an almost complete attenuation of one of the signals. The corollary of (i) is that if a particular signal is active in the time-frequency window $\left(t_j, \omega_k\right)$ for a particular observation record, or realization, then it is also active in the same time-frequency window $\left(t_j, \omega_k\right)$ in all other realizations.

If Assumption 1(ii) were not true, then it would be difficult to distinguish between either of the signals and the ambient noise throughout the observation interval.

Assumption 2. For each signal $s_i\left(t\right)$, there exist adjacent time-frequency windows in the spectrograms of the mixtures $M_i\left(t_0, \omega\right)$, centered on time $t_0 = t_j$ and angular frequency $\omega = \omega_k$ where only one signal $s_i\left(t\right)$ occurs. The method is then based on the complex ratio $\alpha\left(t_j, \omega_k\right)$ of the discrete STFT of the mixtures defined as follows

$$\alpha\left(t_j, \omega_k\right) = \frac{M_1\left(t_j, \omega_k\right)}{M_2\left(t_j, \omega_k\right)} \tag{5.17}$$

Using (5.1), (5.17) may be rewritten as,

$$\alpha\left(t_j, \omega_k\right) = \frac{S_1\left(t_j, \omega_k\right) + c_1 S_2\left(t_j, \omega_k\right)}{S_1\left(t_j, \omega_k\right) + c_2 S_2\left(t_j, \omega_k\right)} \tag{5.18}$$

Therefore, if only one signal occurs in the TF window centered at $\left(t_j, \omega_k\right)$, then the ratio $\alpha\left(t_j, \omega_k\right)$ would simplify to either $\alpha\left(t_j, \omega_k\right) = 1$, in case only signal $s_1\left(t\right)$ is present, or $\alpha\left(t_j, \omega_k\right) = \frac{c_1}{c_2}$, in case only signal $s_2\left(t\right)$ is present. The situation where signals disappear, in some areas of the TF plane, is much more frequent than the case where they disappear at all frequencies during the entire observation time period *T*. The latter case corresponds to one of the signals being exactly equal to zero throughout, an unrealistic possibility.

Assumption 3. When both signals occur in a given set of adjacent TF windows, they vary such that $\alpha\left(t_j,\omega_k\right)$ does not take on the same value in all these windows. Specifically, the method does not work if the signals are identical or if one signal is a constant multiple of the other. It must be noted here that the echoes we are interested in isolating are located at a delay and so, the superimposing echo is approximately an attenuated and delayed version of the source signal.

If only one signal $s_1(t)$ or $s_2(t)$ occurs in several time-frequency windows adjacent in time, then $\alpha\left(t_j,\omega_k\right)$ would take the same value in all these time-frequency windows, whereas the ratio $\alpha\left(t_j,\omega_k\right)$ would take different values across these time-frequency windows if both sources were active, by Assumption 3. In order to exploit this phenomenon, the sample mean of $\alpha\left(t_j,\omega_k\right)$ is computed over both time as well as frequency, over all frequency components ω_k, $k=0,1,...F-1$ (F is the number of points of the discrete STFT computation) and centered around t_j and extending to $0.5W$ samples in time on either side of it. The sample mean is denoted by $\overline{\alpha(j)}$. The calculation of the sample mean is performed, after the spectrograms are calculated, by employing a window covering the entire frequency axis and encompassing *W* time samples. The window is shown shaded in grey in Fig. 5.2. The darker shade of grey indicates overlap between the windows and the lighter shade of grey is reserved for the non-overlapping portion. This window used for arriving at the sample mean $\overline{\alpha(j)}$ is shifted by an amount Δ in the time-domain to obtain the next value of $\overline{\alpha(j)}$ until the entire observation time interval *T* is covered. The center of the first window is at a distance of $0.5W$ time-samples from the origin and the center of the last window $0.5W$ time-samples short of *T*. An interval of size *W* time-samples is clipped from the time-interval *T* and so, $\overline{\alpha(j)}$ covers an interval of size *T-W* in steps of Δ. The expression for $\overline{\alpha(j)}$ may be written as:

$$\overline{\alpha(j)} = \frac{1}{(W+1)F}\sum_{x=-\frac{W}{2}}^{\frac{W}{2}}\sum_{i=0}^{F-1}\left|\alpha\left(t_j - x,\omega_i\right)\right|,\ j=\frac{W}{2},\frac{W}{2}+\Delta,\ldots\left\lfloor T-\frac{W}{2}\right\rfloor \tag{5.19}$$

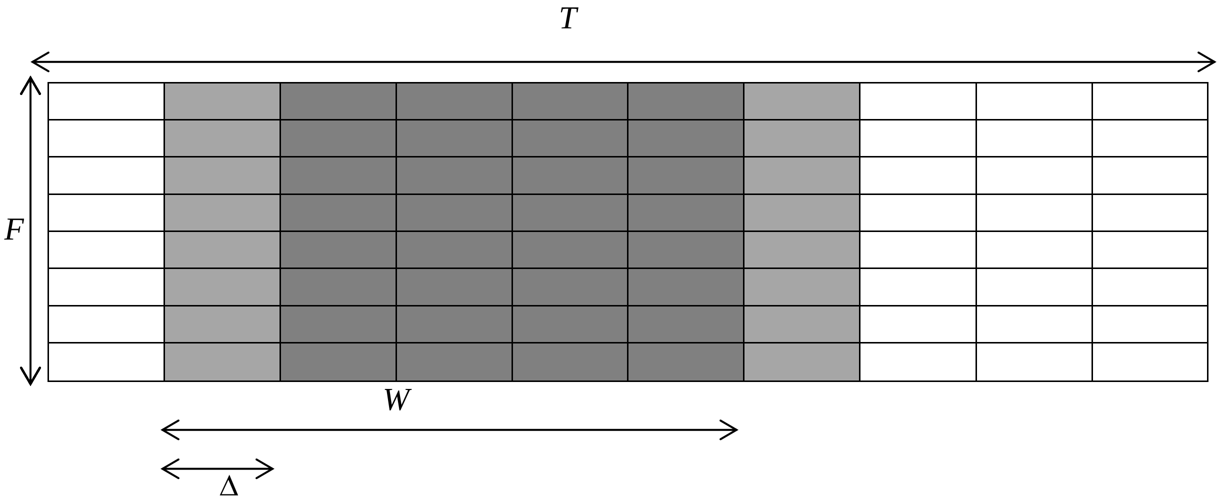


Fig. 5.2: Definition of the windowing process in the time-frequency domain.

The sample mean $\overline{\alpha(j)}$ is constant in those regions of time where only one signal is present and is highly variable when both signals are active. This comes from the fact that the individual signals $s_1(t)$ and $s_2(t)$ are not likely to be uniform DC-type signals but, in reality, information-bearing signals that exhibit a degree of randomness over time. It is also highly improbable that the signals exhibit any degree of periodicity in time. Therefore, even if it is assumed that $s_2(t)$ is a delayed version of $s_1(t)$, then it is reasonable to presume that the ratio $\alpha\left(t_j,\omega_k\right)$, as well as the sample mean $\overline{\alpha(j)}$, would not be exactly equal to unity across all the time frames during which both signals are active.

In order to accurately identify the single-signal-active regions, an approximate derivative of $\overline{\alpha(j)}$ is taken. The approximate derivative is defined as:

$$\Delta\overline{\alpha(j)} = \begin{cases} \overline{\alpha(j)} & j=1 \\ \overline{\alpha(j)}-\overline{\alpha(j-1)} & otherwise \end{cases} \tag{5.20}$$

In the single-signal-active regions, the derivative would evaluate or be close to 0. In those regions of time during which both sources are active, $\Delta\overline{\alpha(j)} \neq 0$.

5.3 Simulation of the TFRM method

5.3.1 The signal $s_1[n]$

The signal $s_1[n]$ was simulated in MATLAB as three identical OFDM bursts, each of length 512, with zeros in between. First, a bit stream of size 512 was created using MATLAB's *randsrc* function, which creates a random array of scalars, either -1 or 1, of the length specified by the user. This set of bits was passed through a BPSK modulator where the data bits 1 were mapped to $-\frac{1}{\sqrt{2}}$ and the data bits 1 were mapped to $\frac{1}{\sqrt{2}}$. A 512-point IFFT was then taken of the modulated bits to generate the OFDM burst. Two replicas of the created OFDM burst were appended to the first one, with 200 zeros interspersed between successive OFDM bursts; the result is the signal $s_1[n]$.

Figure 5.3 encapsulates the process of generating the OFDM burst.

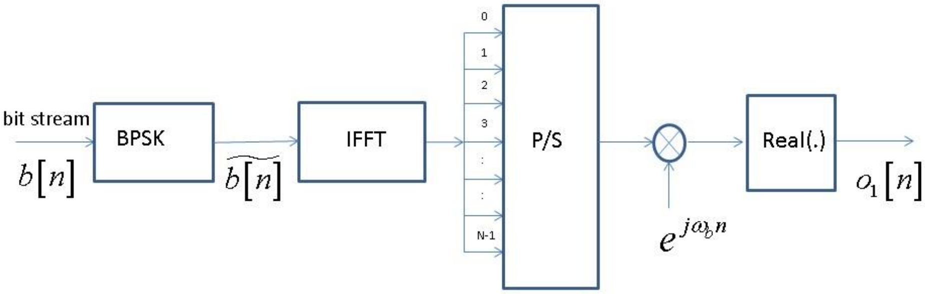


Fig. 5.3: Generation of OFDM burst.

In equation form, the above process may be expressed as

$$o_1[n] = \operatorname{Re}\left(\left(\frac{1}{N}\sum_{k=0}^{N-1} b[k]e^{\frac{j2\pi kn}{N}}\right)e^{j\omega_b n}\right), n = 0,1,\ldots N-1 \tag{5.21}$$

where $o_1[n]$ is a sequence of complex numbers and ω_b is the angular frequency to translate from baseband to bandpass. In order to make $o_1[n]$ realizable, the real part of the sequence is considered for the generation of $s_1[n]$.

The signal $s_1[n]$ is generated by appending two replicas of $o_1[n]$ to $o_1[n]$, with 200 zeros in between, as explained earlier. The angular modulating frequency $\omega_b = 1$ rad/s . A plot of $s_1[n]$ appears in Fig. 5.4.

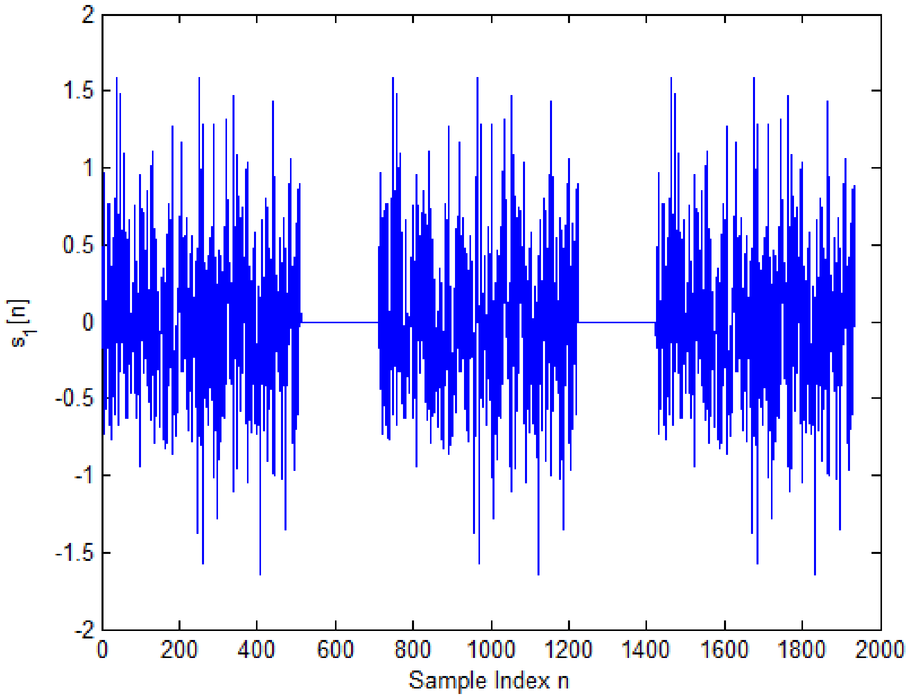


Fig. 5.4: Simulated OFDM source signal $s_1[n]$.

The 2^{18}-point FFT of the signal $s_1[n]$, consistent with the notation used so far, is denoted by $S_1(\omega_k)$ and shown in Fig. 5.5. The FFT was generated using MATLAB's *freqz* function which accepts as arguments the numerator and denominator polynomial coefficients of a filter and the desired size of the FFT to be used and generates the frequency response of the filter. To generate the FFT of the signal $s_1[n]$, the numerator polynomial was the signal itself, i.e. $s_1[n]$, the denominator polynomial was 1, and the FFT size was 2^{18}.

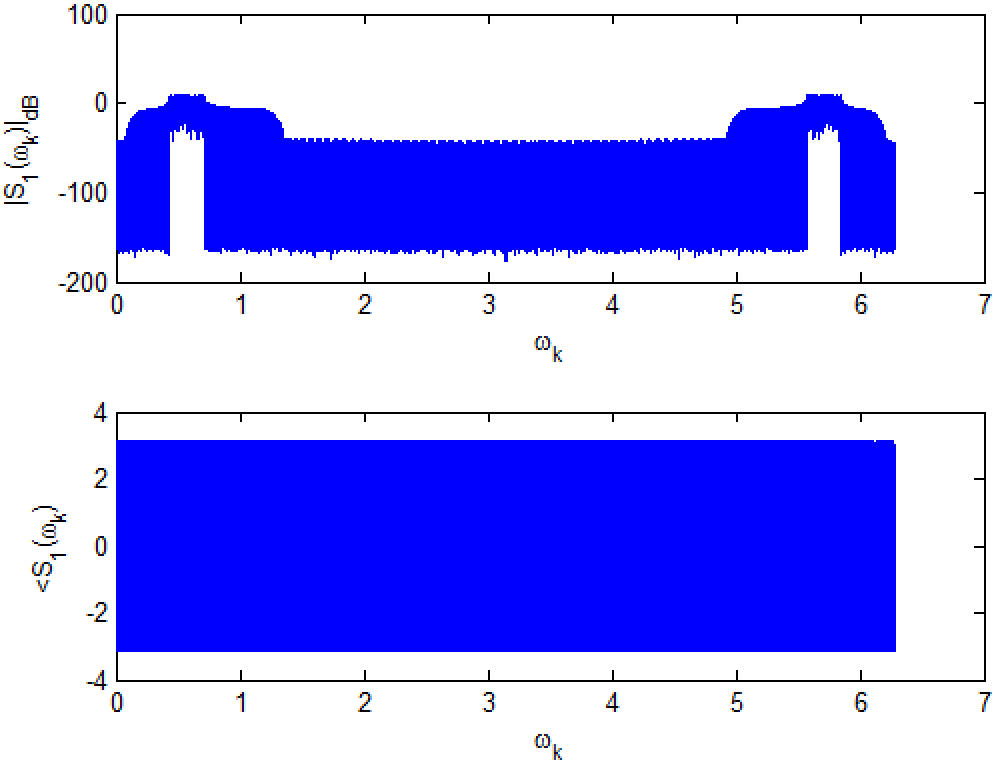
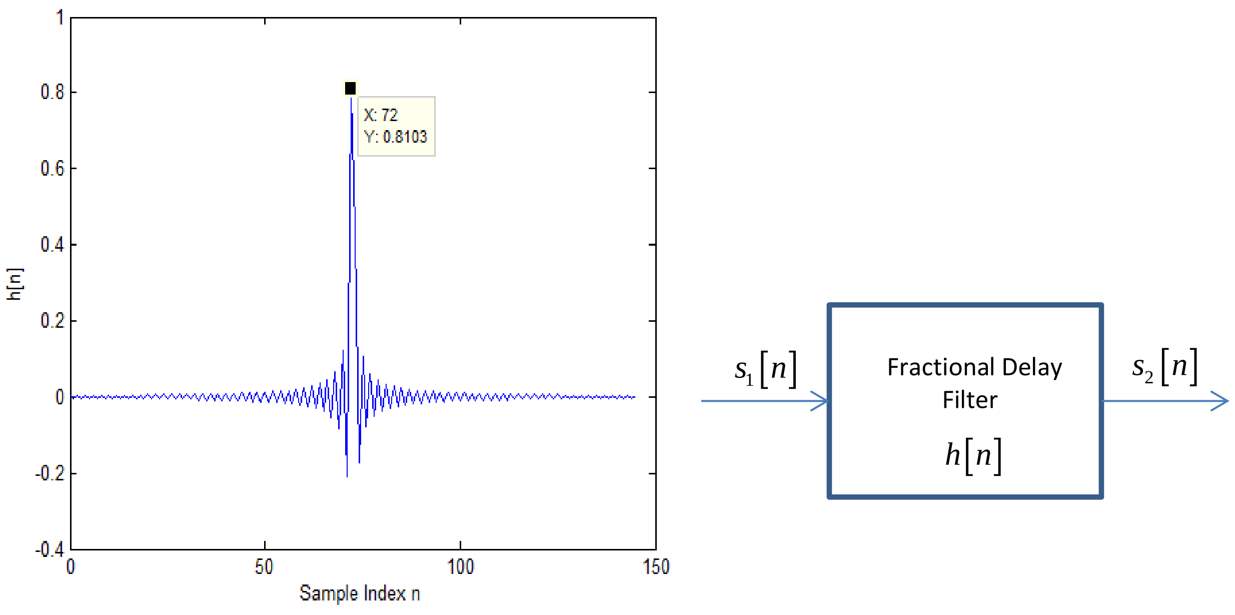


Fig. 5.5: Frequency domain representation of $s_1[n]$.

The signal has a passband extending from 0.4 to 0.7 radians. The two-sided frequency response is plotted in the above graph on the positive frequency axis and so the even magnitude response can be seen to repeat itself. Expressed in frequency terms, taking 2π radians to represent F_s, the sampling frequency which is 44.1 kHz, the lower edge of the passband is $\left(\left(\frac{0.4}{2\pi}\right)44.1 \text{ kHz}\right) = 2.8075$ kHz and the upper edge of the passband is 4.9131 kHz. The passband is squarely in the audible range, making it possible to hear transmissions during the conduct of experiments.

5.3.2 Generating the two mixtures

The signal $s_2[n]$ is generated by applying a delay to $s_1[n]$. The important aspect to be considered when applying a delay is that the delay must have both an integral component as well as a fractional component (in terms of sampling intervals). This is because the delay τ in the continuous time domain can be arbitrary and when the mixture is sampled, this arbitrary delay results in a discrete-time delay that necessarily possesses a fractional part. To simulate a discrete-time delay, it becomes necessary to interpolate between known sample values. The interpolated signal is then delayed by the requisite amount. The filter that performs the functions of both interpolation and delay is the *sinc* filter whose impulse response, the *sinc* function, is centered at the required delay. Fig. 5.6(a) shows the impulse response of the fractional delay filter and Fig. 5.6(b) shows the block diagram for generating a fractionally delayed version of $s_1[n]$. The true maximum of the *sinc* function is attained at $n = 72.35$ where $\text{sinc}[n-72.35] = 1$ but n takes on only integer values. At $n =$ 72, the *sinc* function attains 0.8.



(a) (b)

Fig. 5.6: Applying a fractional delay to the source signal.

Another way of generating a sub-sample delay is to apply a delay first to $s_1[n]$, summing it with un-delayed $s_1[n]$, and then to decimate the mixtures $m_i[n]$ by the required factor to generate the appropriate delay. For example, if a mixture consists of two signals that are separated by 500 samples, then decimation by a factor of 3 would result in a mixture with an echo at $\frac{500}{3} = 166.67$ sample delay. This method however requires the use of an anti-aliasing, low-pass filter before the decimation step to function properly.

In Section 5.6, it will be explained that in practice only one mixture $m_1[n]$ may be available for computation. As differences in the bursts that make up $m_1[n]$ will be used to find the delay, it makes sense to generate mixtures $m_1[n]$ and $m_2[n]$ with (near-) identical coefficients. The case for which results are shown is one with coefficients $c_1 = 0.88$ and $c_2 = 0.9$. The delay D_{true} applied to generate $s_2[n]$ was 72.35 samples. In equation form,

$$\begin{aligned} m_1[n] &= s_1[n] + 0.88 s_1[n - 72.35] \\ m_2[n] &= s_1[n] + 0.90 s_1[n - 72.35] \end{aligned} \tag{5.22}$$

The short-time Fourier transforms of both mixtures are estimated by making use of MATLAB’s *spectrogram* function. The windowing function $w[n]$ used is a Hanning window of size 32. Successive time-frequency points were obtained by shifting the window by 1 time sample.

In Fig. 5.7 the spectrograms of the two mixtures are shown. The sampling frequency used is $F_s = 1$ Hz (this can be thought of as a normalization). The spectrograms of the two mixtures are nearly identical as intuition suggests.

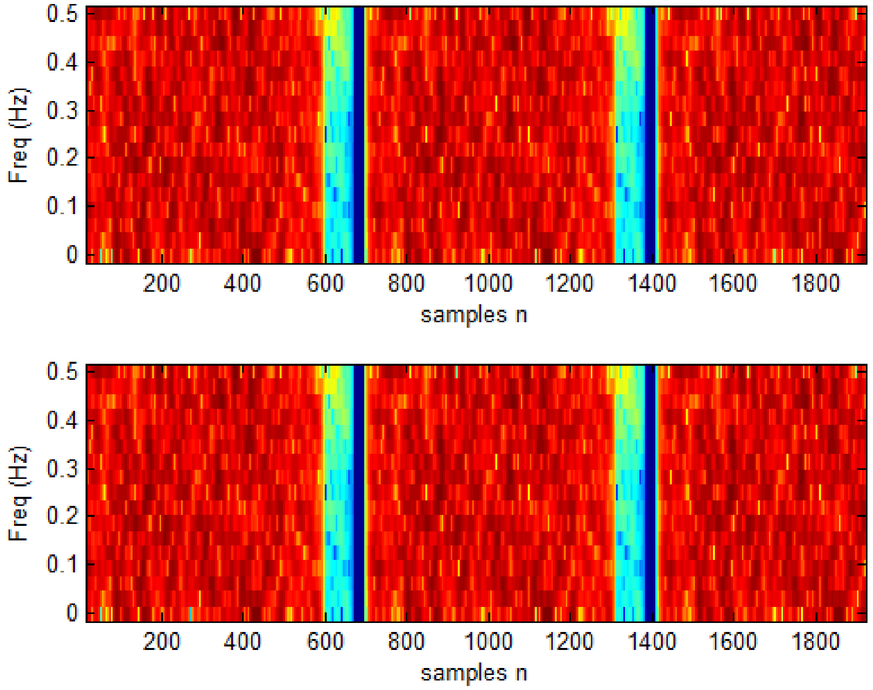


Fig. 5.7: Spectrograms of $m_1[n]$ (top) and $m_2[n]$ (bottom).

The plot near *n*=600 and *n*=1400 represents the spectrogram of the transition between successive bursts. The length of $m_i[n]$ in samples is the same as the length of either $s_1[n]$ or $s_2[n]$, which is 1936 samples, as shown in Fig. 5.4. Due to the length of the Hanning window and the overlap, the time variable *T*, defined in Fig. 5.2, of the spectrogram is 1904 (1936-32) samples long. The sample mean function $\overline{\alpha(j)}$ is generated according to (5.19) with $W=6$ and $\Delta=3$ and is shown in Fig. 5.8. The index j is the center of a window in the time-domain encompassing the entire frequency domain and its maximum value is $\left\lfloor T-\frac{W}{2}\right\rfloor=1901\text{ samples}$, as shown in (5.19). The increment factor Δ decides how many windows occur in the time interval for which the spectrogram is defined.

The derivative of $\overline{\alpha(j)}$, defined as $\Delta\overline{\alpha(j)}$ is useful in identifying the single-source regions and its extent in time is the same as that of $\overline{\alpha(j)}$. The derivative of $\overline{\alpha(j)}$ was found as explained in (5.20) and is shown in Fig. 5.9.

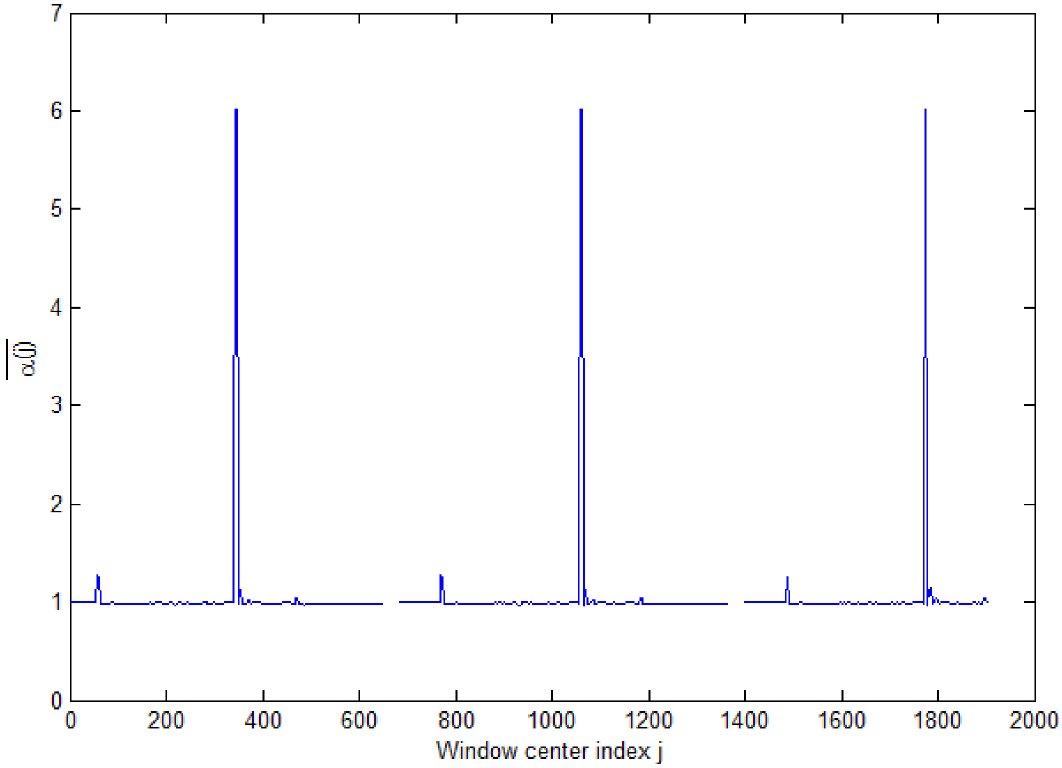


Fig. 5.8: Sample mean $\overline{\alpha(j)}$.

Near the beginning, for window center indices *j* of about 0 to 50, only the signal $s_1[n]$ is active and so the sample mean of the ratio of the spectrograms is 1. Later, near the end of the signal, beyond $j = 520$, when $s_1[n]$ has died down and only $s_2[n]$ is active, the sample mean becomes the ratio $\frac{c_1}{c_2} = \frac{0.88}{0.90} = 0.9778$. A zoomed version of $\overline{\alpha(j)}$, focusing only on the first burst is shown in Fig. 5.10 so as to give a clearer idea of the exact ratio.

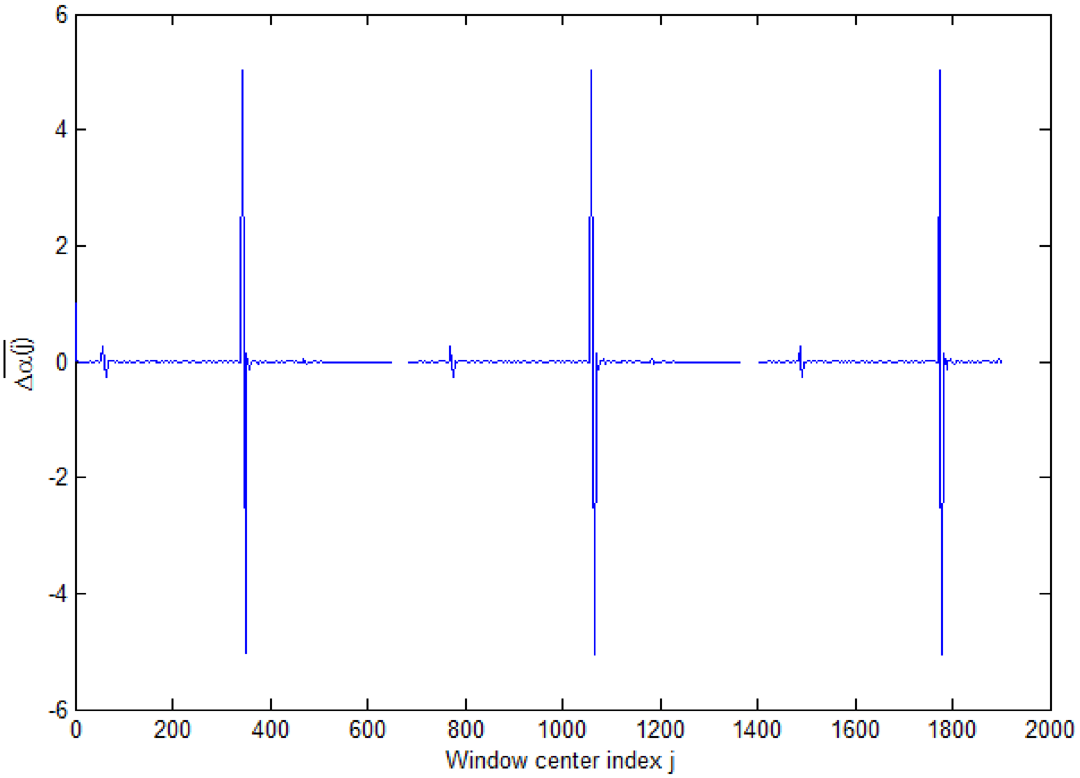


Fig. 5.9: Derivative of $\overline{\alpha(j)}$.

The derivative function in Fig. 5.9 merely confirms what was observed in Fig. 5.8. Where only one signal is active, $\overline{\alpha(j)}$ is constant and therefore, the derivative is exactly 0. For example, for $j \in (0,50)$, the derivative is zero, as it is at $j = 600$, signifying the regions where only $s_1[n]$ and $s_2[n]$ are active respectively.

Those indices where the derivative goes to zero are used to obtain the single-source regions and these indices are used in deriving from $\overline{\alpha(j)}$ the ratio $\rho = \frac{c_1}{c_2}$.

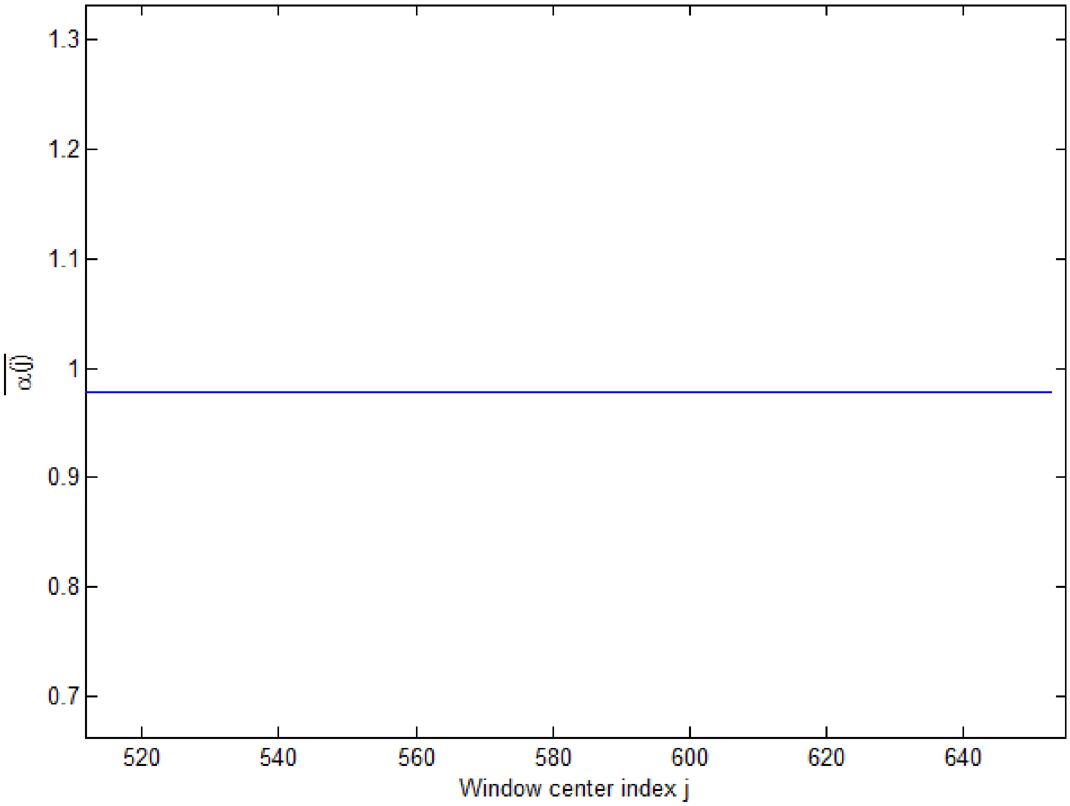


Fig. 5.10: Zoomed version of $\overline{\alpha(j)}$ focusing on the end of the first burst.

Between $j = 50$ and $j = 500$, there is no discernible pattern in the ratio; the pattern is highly variable without the maintenance of a constant level for any length of time. At around $j = 600$, as mentioned with reference to Fig. 5.7, the transition between successive bursts occurs and this is where it becomes possible to notice the effects of a single source. This is substantiated in Fig. 9 where $\Delta\overline{\alpha(j)}$ goes to 0 near $j = 600$ and in Fig. 5.10 where $\overline{\alpha(j)}$ is shown to maintain a constant level between $j = 520$ and $j = 620$.

A point in the later region around which $\Delta\overline{\alpha(j)}$ is zero or close to 0, i.e. $j = 600$, is taken as $\hat{\rho} = \frac{\hat{c}_1}{\hat{c}_2}$ and the method described in Section 5.2 is used to arrive at estimates for $s_1[n]$ and $s_2[n]$.

The estimated signals $\hat{s}_1[n]$ as well as $\hat{c}_1\hat{s}_2[n]$ are shown in Fig. 5.11.

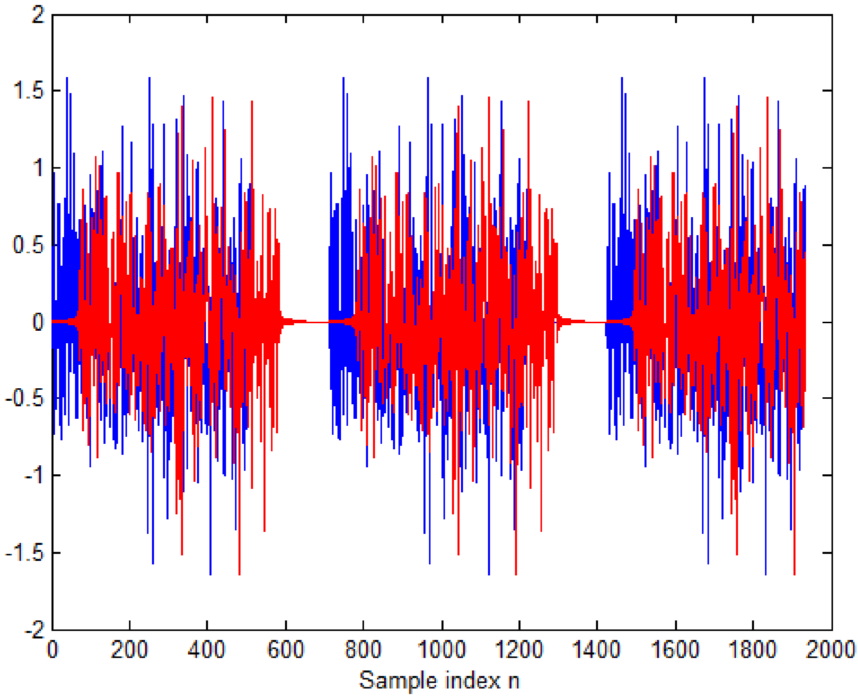


Fig. 5.11: $\hat{s}_1[n]$ (blue) and $\hat{c}_1\hat{s}_2[n]$ (red).

Note that $\hat{c}_1\hat{s}_2[n]$ is delayed with respect to $\hat{s}_1[n]$. In this case the scaling factor $c_1 = 0.88$ is close to 1 and so both signals appear to be similar.

The error function $e(\beta, D)$, defined in (5.11), is generated by storing the norm after each subtraction of $\hat{c}_1\hat{s}_2[n]$ and $\beta\hat{s}_1[n]$. The scaling factor β is varied from 0 to 1 in steps of 0.1. The delay parameter D is varied from 30 to 80 in steps of 0.01. The delay estimate $\hat{D}$ is returned by finding the minimum of the contour surface $e(\beta, D)$, shown in Fig. 5.12. The colorbar by the side of the contour plot exists to help visually identify the minimum.

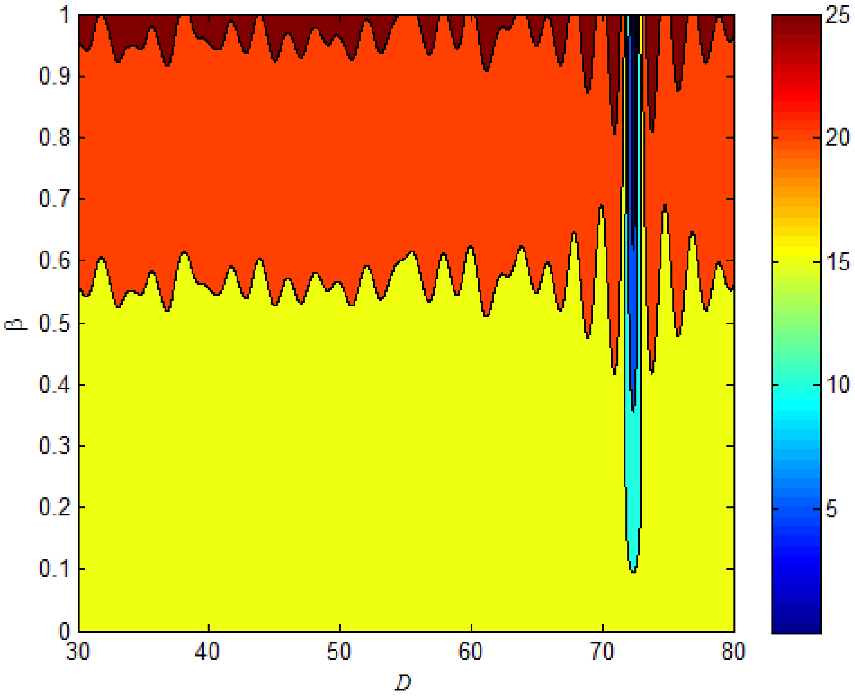


Fig. 5.12: Contour of $e(\beta, D)$.

The global minimum of the error contour $e(\beta, D)$ occurs at $\hat{D} = D = 72.35$, which is indeed the delay applied to $s_2[n]$, and at the scaling coefficient $\beta = 0.88$, which was indeed the scaling originally applied to the signal $s_2[n]$.

The TFRM method was simulated under no-noise conditions and with nearly identical mixtures. The usefulness of the sample mean function derives from the ease of locating single-signal-active regions and the level of $\overline{\alpha(j)}$ in the single-signal-active regions is used to estimate the ratio $\hat{\rho} = \frac{\hat{c}_1}{\hat{c}_2}$.

5.4 Equal Coefficients

5.4.1 The $\frac{c_1}{c_2} = 1$ case

When both c coefficients are equal, in the absence of any noise, the TFRM method fails. Consider

$$\begin{aligned} m_1[n] &= s_1[n] + c_1 s_2[n] \\ m_2[n] &= s_1[n] + c_1 s_2[n] \end{aligned} \tag{5.23}$$

In this case both mixtures are identical and, therefore, their time-frequency transforms are also identical. The ratio $\alpha(t_j, \omega_k)$ is then always equal to 1 and the windowed mean $\overline{\alpha(j)}$ equals 1 throughout. Multiplying $m_2[n]$ by 1 and subtracting from $m_1[n]$ results in 0, and nothing useful can be derived from that. The problem is ill-posed in this case.

5.4.2 With some noise

Consider the case when the coefficients c_1 and c_2 are equal and white Gaussian noise is added to each of the mixtures independently. Rewriting (5.23),

$$\begin{aligned} m_1[n] &= s_1[n] + c_1 s_2[n] + a_1[n] \\ m_2[n] &= s_1[n] + c_1 s_2[n] + a_2[n] \end{aligned} \tag{5.24}$$

where $a_1[n]$ and $a_2[n]$ are two sample functions of a zero-mean Gaussian random processes with variance σ_N^2.

It is useful to define another quantity, the signal-to-noise ratio (SNR), as follows

$$SNR_{dB} = 10\log_{10}\left(\frac{\mathrm{var}(m_1[n])}{\sigma_N^2}\right) = 10\log_{10}\left(\frac{\mathrm{var}(m_2[n])}{\sigma_N^2}\right) \tag{5.25}$$

The no-noise example with nearly identical coefficients (c_1=0.88, c_2=0.9) was repeated with the following modification: $c_1 = c_2 = 0.9$ and noise was varied in such a manner that SNR_{dB} went from 0 dB to 40 dB in steps of 5 dB. The delay applied to $s_1[n]$ was again 72.35 samples.

A plot of a sample function of the noise process is shown in Fig. 5.13. The variance of the mixtures without noise, i.e. the signal component was 0.3653. For an SNR=5 dB, the variance of the noise is $\sigma_N^2 = 0.3653E - 0.5 = 0.1155$. The variance of the noise function shown in Fig. 5.13 is 0.1155.

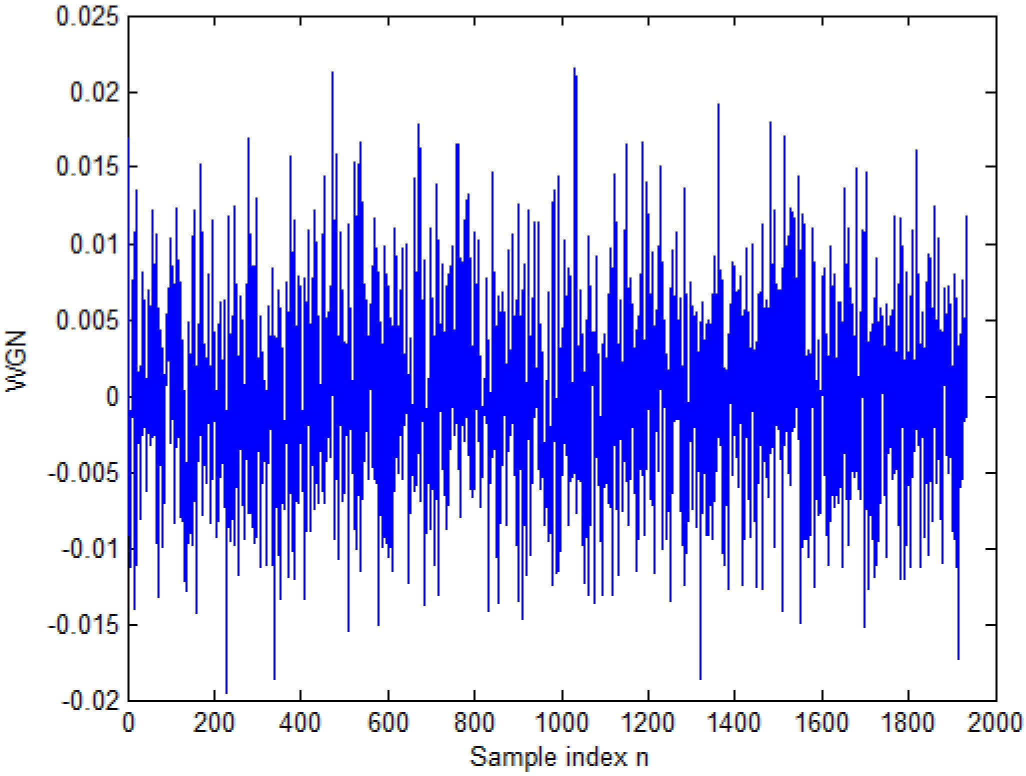


Fig. 5.13: White Gaussian noise with $\sigma_N^2 = 0.1155$.

As noted in Section 5.4.1, when the two mixtures are exactly identical, the TFRM method fails because it is impossible to eliminate either $s_1[n]$ or $s_2[n]$ when subtracting one mixture from the other results in identically 0 throughout.

Figure 5.14 shows the windowed mean function $\overline{\alpha(j)}$ for $SNR_{dB} = 5\text{ dB}$. Due to the high variance of the noise compared with the variance of the mixture, it is difficult to visually discern the single-source regions in $\overline{\alpha(j)}$. The accuracy of the delay estimate depends on the variance of the noise added.

Figure 5.15 shows the recovered signals $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$.

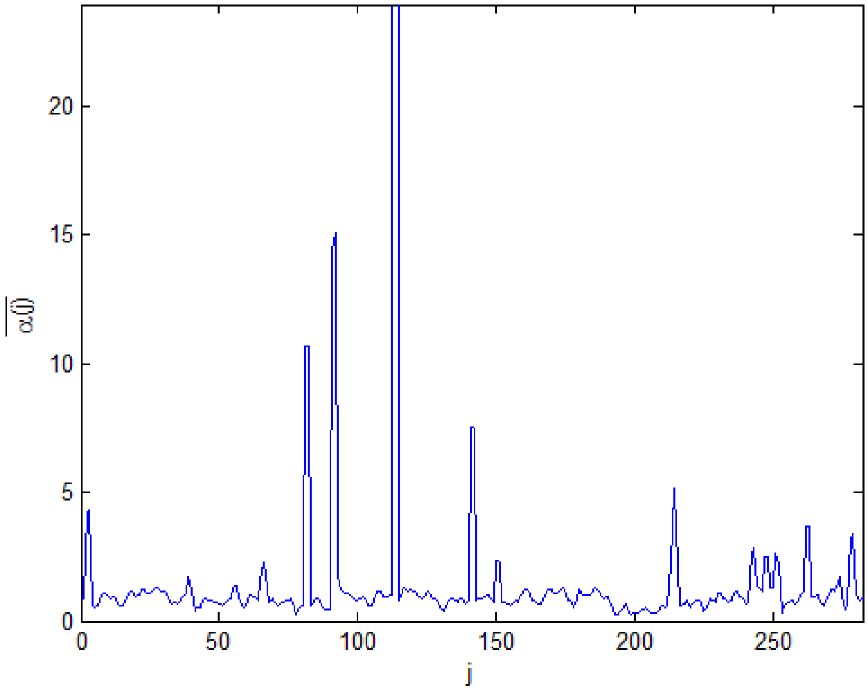


Fig. 5.14: $\overline{\alpha(j)}$ for $SNR_{dB} = 5\text{ dB}$.

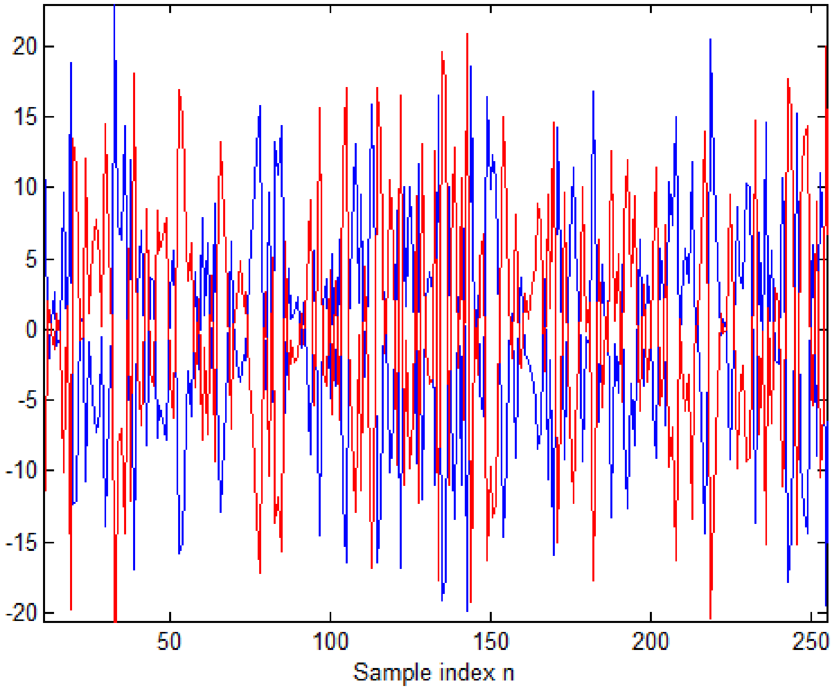


Fig. 5.15: $\hat{s}_1[n]$(blue) and $\hat{c}_1\hat{s}_2[n]$(red) from TFRM method.

These are the signals recovered from the algorithm and as such the harmful effects of noise can be seen. The signals are buried deep in noise and as such bear no resemblance to the original $s_1[n]$ in Fig. 5.4. Figure 5.16 shows the contour plot of the search for the true delay.

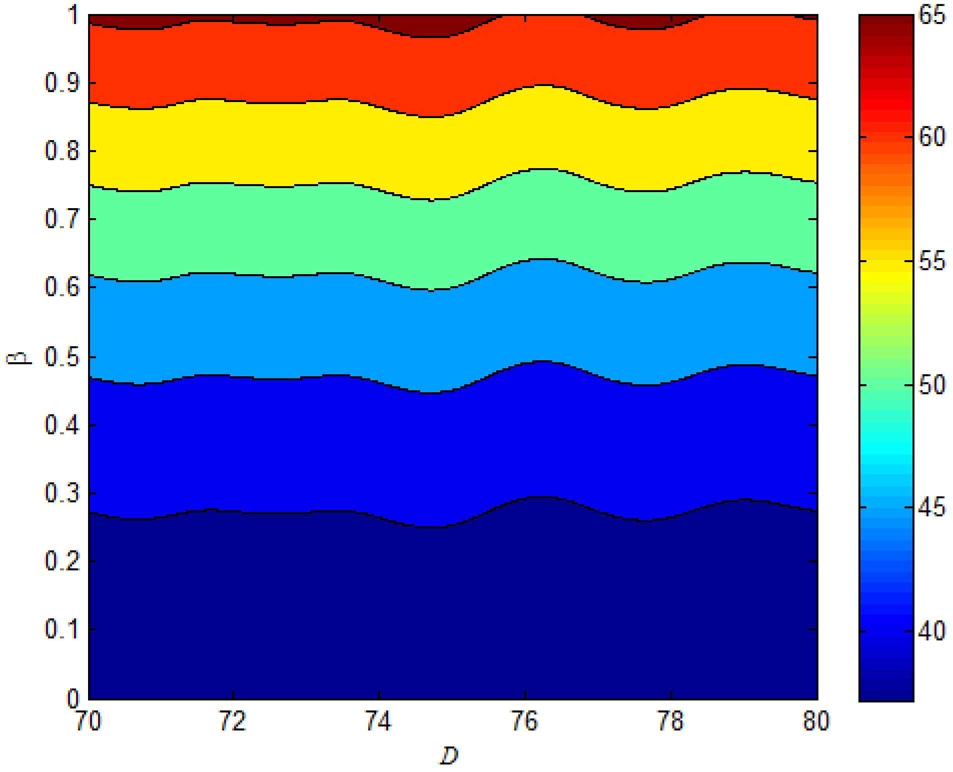


Fig. 5.16: Contour plot of $e(\beta, D)$.

Desirable operation of the algorithm corresponds to a global minimum being reached at the simulated delay D=72.35 and $\beta = c_1$=0.9. In the contour in Fig. 5.16, dark blue indicates a low level and red indicates a high level. At $D = 72.35$ and $\beta = 0.9$, the color of the plot is not dark blue and hence, the result is wrong. The minimum on the other hand is at a value of β closer to 0. This is better illustrated by the colorbar appearing beside the plot in Fig. 5.16. Since the global minimum is not at the right place, the result is wrong. The problem is that while the TFRM method requires some noise for its correct functioning, it fails when there is too much noise. This is the underlying paradox in the useful operation of the TFRM algorithm. *Some* bad is good, but too much is not.

Noise of lower variance was added so that $SNR_{dB} = 40$ dB. Figure 5.17 shows the $\overline{\alpha(j)}$ function for $SNR_{dB} = 40$ dB.

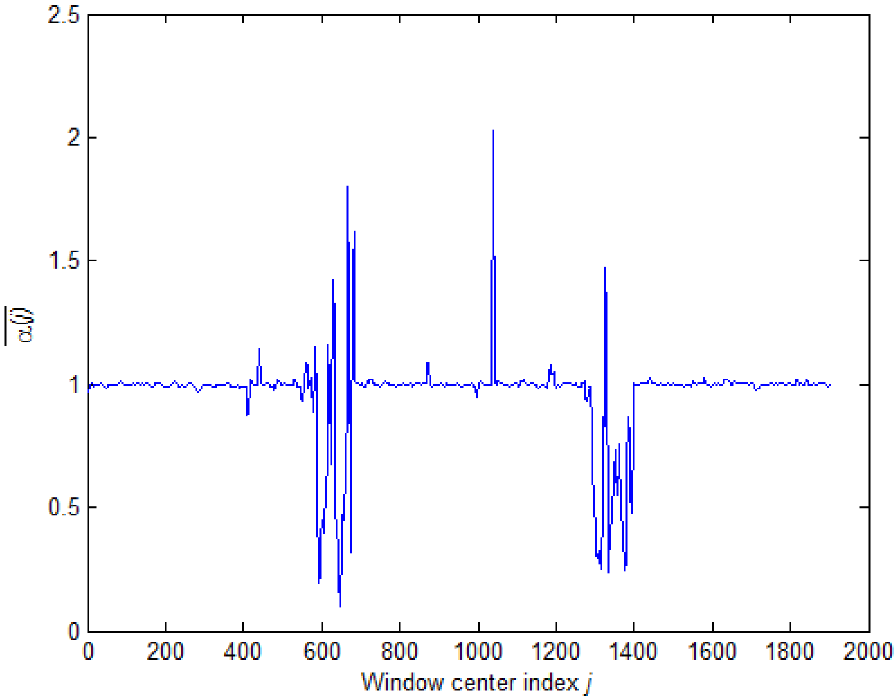


Fig. 5.17: $\overline{\alpha(j)}$ for $SNR_{dB} = 40$ dB.

As has been pointed out in the explanation for Fig. 5.7, the region around $j = 600$ is a transition region. Recall that that zeros were interspersed between successive OFDM bursts during the generation of $s_1[n]$ (Section 5.3.1, Fig. 5.4) and when there is noise added to the mixtures, the transition region contains noise rather than simply zeros. It is possible to identify this region in Fig. 5.17 by noting the high variability of $\overline{\alpha(j)}$ around the region $j = 600$. The value of $\overline{\alpha(j)}$ just before the transition region begins is taken as $\hat{\rho} = \frac{\hat{c}_1}{\hat{c}_2}$. Instead of visually isolating the transition region, the process is automated by means of using the derivative function $\Delta\overline{\alpha(j)}$ shown in Fig. 5.18.

The recovered signals $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$ are shown in Fig. 5.19. The contour plot of $e(\beta, D)$ is shown in Fig. 5.20. The contour plot was generated in the manner described in Section 5.1.

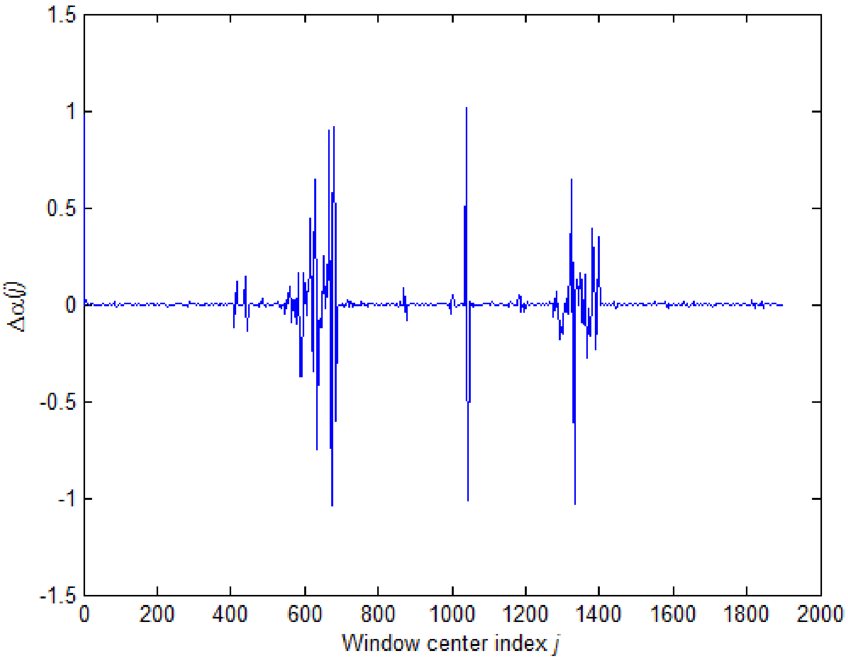


Fig. 5.18: $\Delta\overline{\alpha\left(j\right)}$ for $SNR_{dB}=40$ dB.

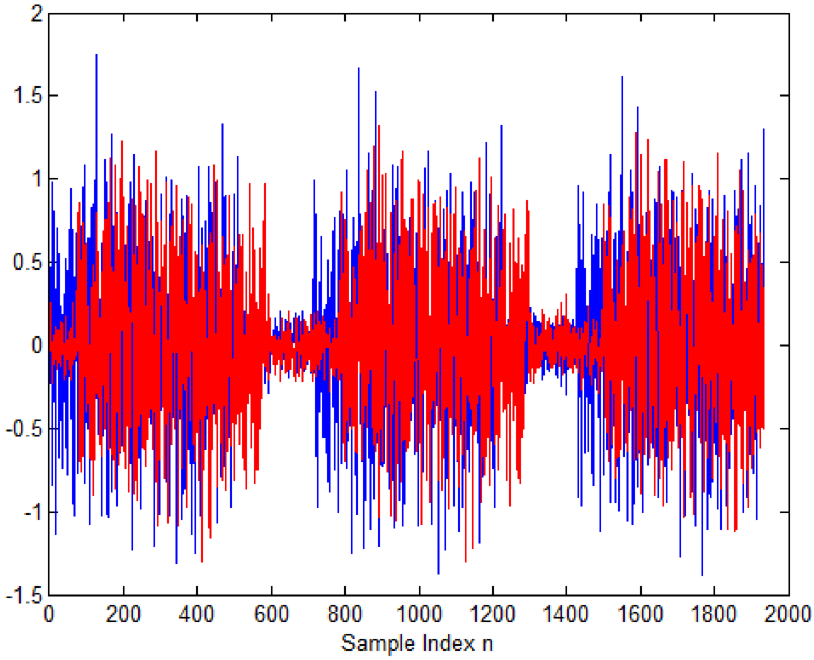


Fig. 5.19: $\hat{s}_1\left[n\right]$ (blue) and $\hat{c}_1\hat{s}_2\left[n\right]$ (red) from TFRM method for $SNR_{dB}=40$ dB.

The signals appear nearly identical as is expected since the scaling factor employed was 0.9 and $\hat{c}_1\hat{s}_2[n]$ is delayed with respect to $\hat{s}_1[n]$ as it is expected to be.

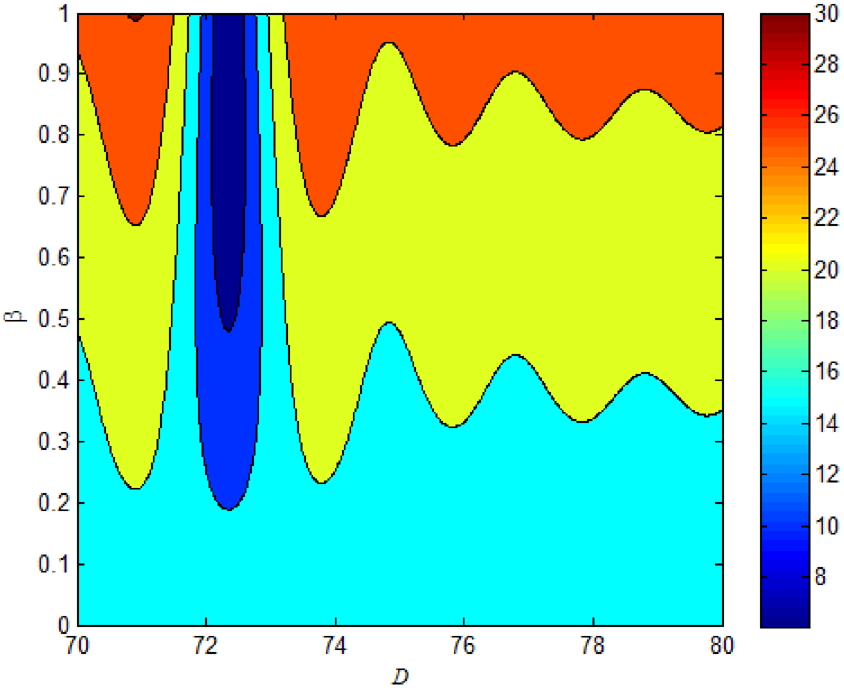


Fig. 5.20: Contour of $e(\beta, D)$ for $SNR_{dB} = 40$ dB.

The global minimum of the contour surface $e(\beta, D)$ in Fig. 5.19 appears at $\hat{D} = D = 72.34$ and $\beta = \hat{c}_1 = 0.9$, which is very close to the correct result. The slight offset is speculated to be due to the amount of noise added.

5.5 The Distribution of Estimates

5.5.1 The probability density function (pdf)

In the presence of noise, the delay estimate returned by the TFRM method is never the same with repeated experiments, even if the variance of the noise remains unchanged. In other words, it is more appropriate to refer to a distribution of estimates rather than to a single delay estimate.

An estimator $\hat{\theta}$ of a parameter θ from observations μ having a probability density function $f(\mu;\theta)$ is unbiased if:

$$E\left(\theta\right) = E(\theta) \tag{5.26}$$

Informally, the closer to the true mean that the estimates are, the “better” the estimator is. In the case of the TFRM method, the variance of the estimates returned depends on the variance of the noise added. As the amount of noise, measured in terms of the variance, decreases, the variance of the estimates approaches 0. In the extreme case of no noise, the estimator always returns the same estimate.

In this chapter, the estimator returns an estimate $\hat{D}$ from observations μ that contains the true delay D_{true} in its distribution. The distribution of the delay estimates is $f\left(\hat{D}\right)$.

Experimentally determined distributions of the delay estimates for different values of SNR_{dB} are shown in Fig. 5.21.

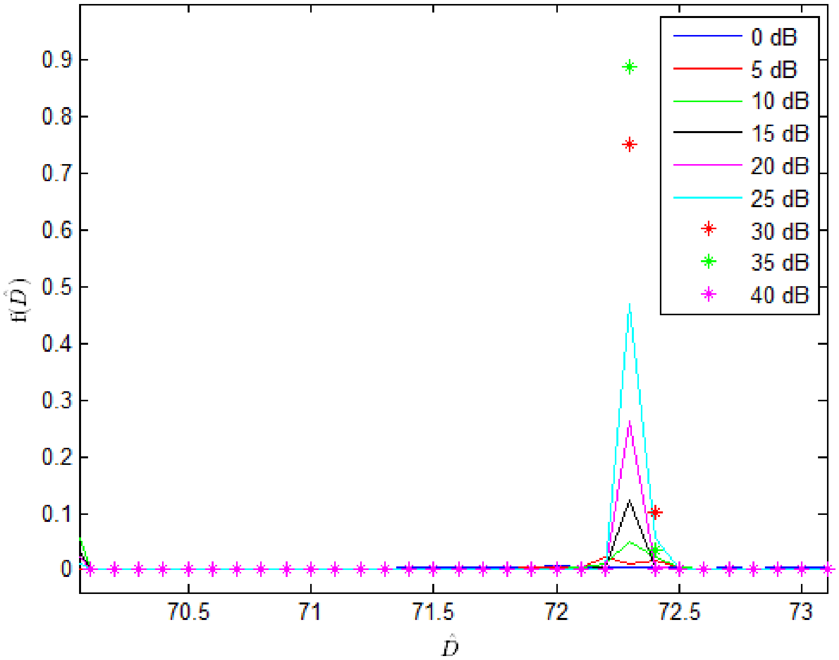


Fig. 21: The behavior of $f(\mu;D_{true})$ with respect to SNR_{dB}.

Observe that with an increase in SNR_{dB}, the probability density functions approach the true delay with diminishing variance, with most of the density functions displaying a peak at the desired mode. This is clearer in Fig. 5.22, where a zoomed version of Fig. 5.21 with the region between $\hat{D} = 72.1$ and $\hat{D} = 72.5$ is being magnified.

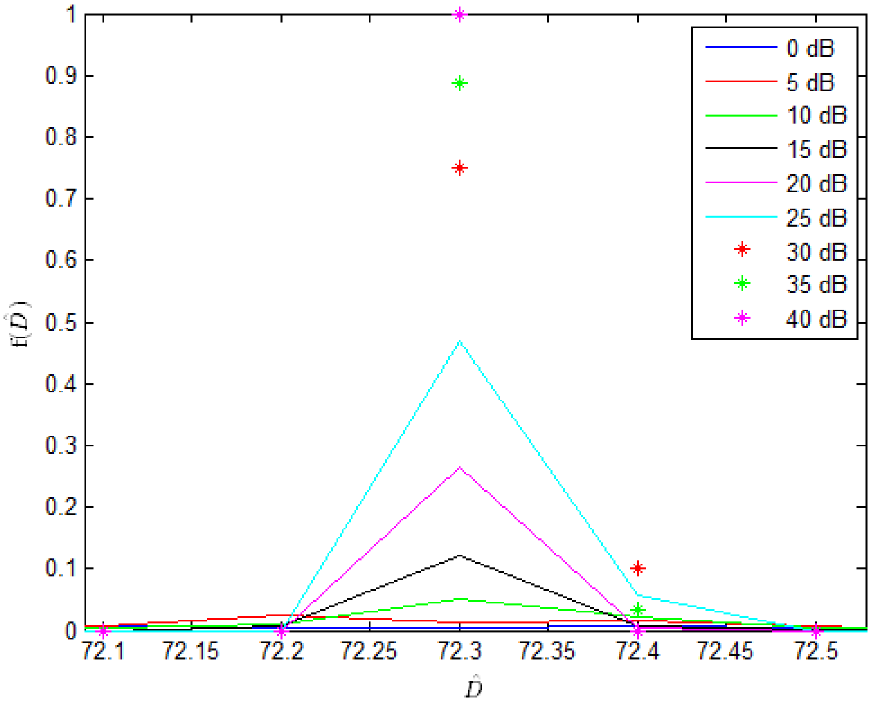


Fig. 5.22: Zoomed version of Fig. 5.20 focusing on $72.1 \leq \hat{D} \leq 72.5$.

The standard deviation of the estimates against the signal-to-noise ratio (SNR) is shown in Fig. 5.23. The formal way of expressing the relationship between the variance of an estimator σ_D^2 and the strength of the observations is by means of the Cramer-Rao bound, shown in the Appendix.

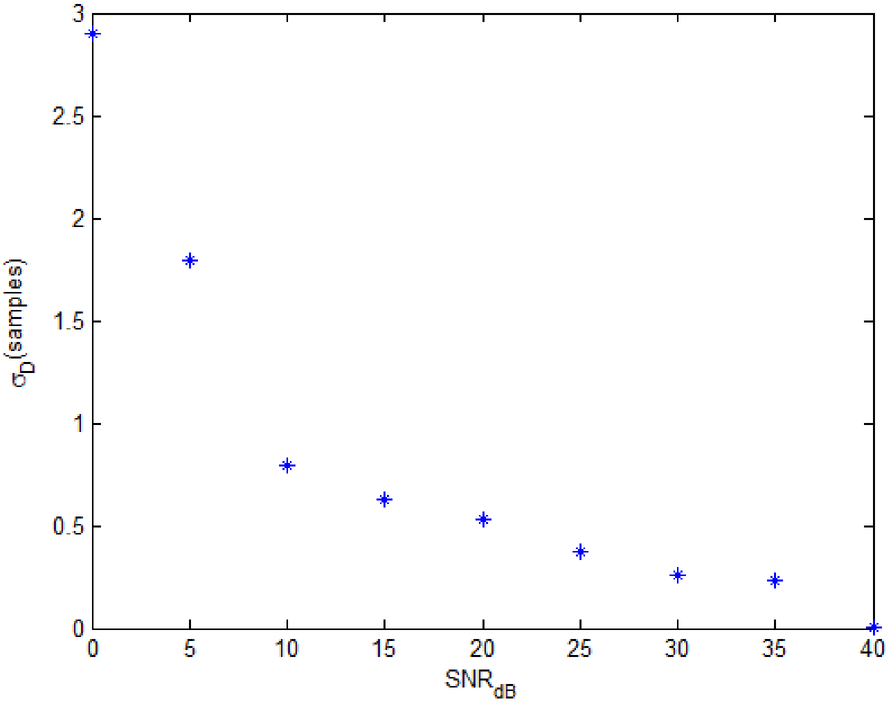


Fig. 5.23: σ_D versus SNR_{dB} (from simulation).

The search was carried out by varying the delay parameter from 70 through 80 in steps of 0.01 and hence the TFRM estimator returned delay estimates $\hat{D}$ that were upper-bounded by 80 and lower-bounded by 70. Estimates at the lower end of the search interval, i.e. between 70 and 70.1 are to be interpreted as being possibly very far from the true delay. If the search interval were extended to include 30, or 20 at the lower end, then those delay estimates $\hat{D}$ between 70 and 70.1 would instead be close to 30 and 20 respectively.

The TFRM estimator is said to have failed when the variance of the delay estimates is close to or comparable to the size of the range of delays in which the search is being performed. For example, in this case the range of delays is 10 samples long (80-70) and if the TFRM method returns delay estimates with a variance of 9 or 9.5, then the estimator can be classified as having failed. This is indeed the case in Fig. 5.22 at $SNR_{dB} = 0\text{ dB}$, where $\sigma_D = \sqrt{9} = 3\text{ samples}$.

5.6 Experimental Results

5.6.1 The experimental set-up

The TFRM method was executed on recordings collected in the DSPRL under room conditions. Figure 5. 24 shows a schematic of the experimental set-up.

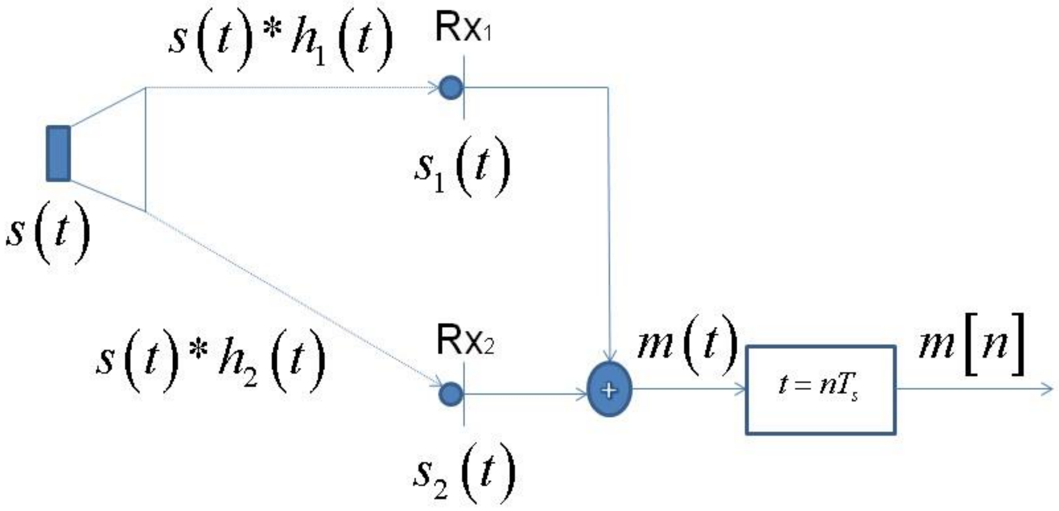


Fig. 5.24: Schematic of the experimental measurement set-up.

In order to eliminate feedback from the transmitting apparatus to the receiving apparatus in the soundcard, the transmission was done from a speaker connected to one laptop while the result of the summing junction $m(t)$ was sampled in another laptop. The sampling frequency used was $F_s = \frac{1}{T_s} = 44.1\text{ kHz}$.

The transmitted signal $s(t)$ consists of 4 OFDM bursts interspersed by 10000 zeros. The plot of signal $s(t)$ generated in MATLAB before transmission is shown in Fig. 5.25. The mixture $m(t)$ consists of a sum of the signals received along the paths to the two microphones. The sampled received $m(t)$ when one microphone was located at a distance of 0.36 m from the source and the other at a distance of 0.8 m from the source is shown in Fig. 5.26.

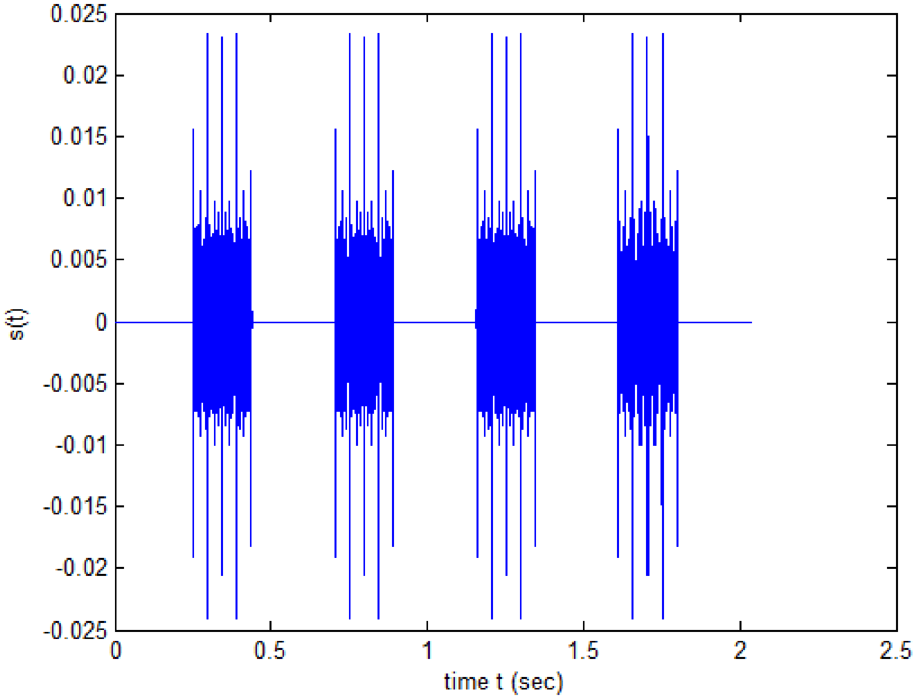


Fig. 5.25: Transmitted signal $s(t)$ generated in MATLAB.

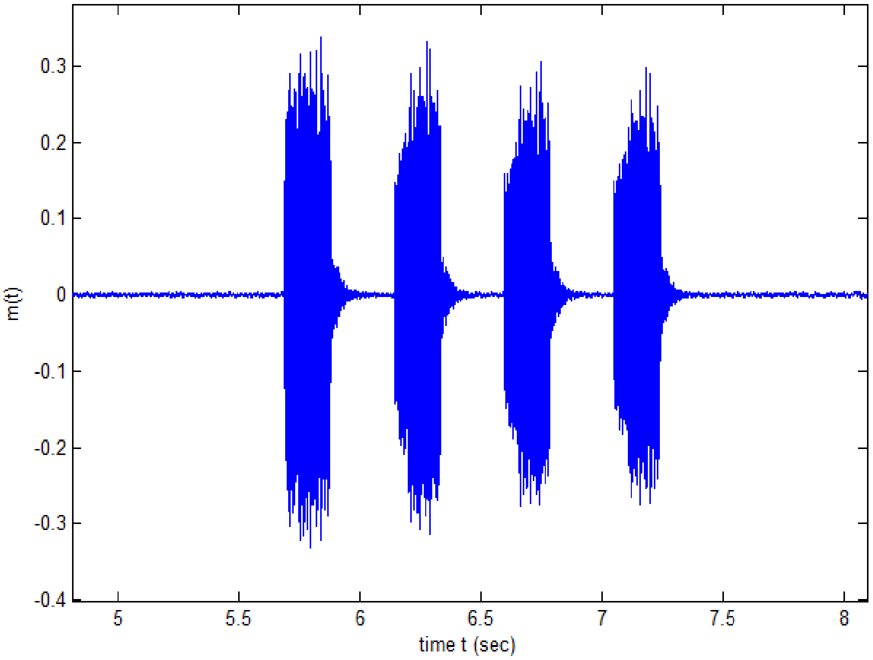


Fig. 5.26: Sampled received mixture $m(t)$.

The received signal lasts longer than the transmitted signal, which is an effect of the reverberation in the room. Even though the transmitted signal stops abruptly, it takes some time before the transmitted signal decays to 0. The recording apparatus must take this into account in setting the time for which it records the *mic* input. The recording was performed for a duration that was 7 times larger than the duration of the transmitted signal. Noise was recorded by running the recording routine and not running the transmission routine and shown in Fig. 5.27. The reason for recording the noise-only signal is because it is necessary to obtain the variance of the noise function so as to be able to calculate the ambient SNR.

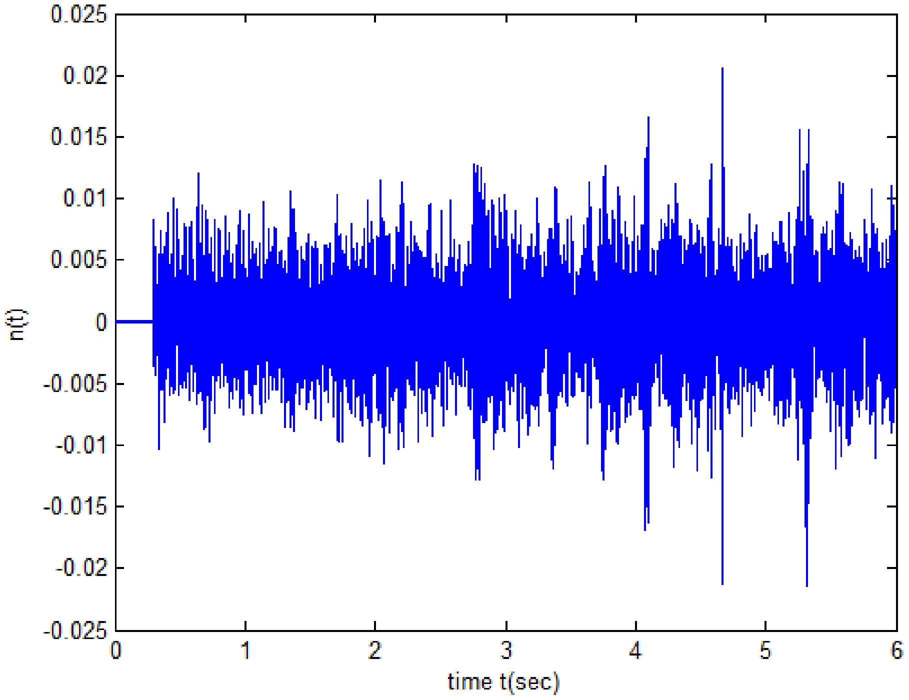


Fig. 5.27: Room noise $n(t)$.

The signal-to-noise ratio was calculated as a ratio of the variances of the signal received and that of the noise. Four distinct pulses may be identified in Fig. 5.26 and the variance of the signal was calculated by isolating these pulses and taking the mean of the variances of these 4 pulses. The variance of the pulses were $8.7980 \text{ x } 10^{-4}$, $7.8167 \text{ x } 10^{-4}$, $10 \text{ x } 10^{-4}$, and $11 \text{ x } 10^{-4}$. The average variance taken over these four pulses is $9.3847 \text{ x } 10^{-4}$. The variance of the noise in Fig. 5.26 is $0.18156 \text{ x } 10^{-6}$. The ambient SNR can now be calculated according to (5.25) as:

$$SNR_{dB} = 10\log_{10}\left(\frac{9.3847 \text{ x } 10^{-4}}{0.18156 \text{ x } 10^{-6}}\right) = 37.1340 \text{ dB} \qquad (5.27)$$

5.6.2 The TFRM method on a single mixture

5.6.2.1 Case 1: $\hat{D}_{theory} = 57.13$

In the derivation and theoretical simulation of the algorithm, it was assumed that two distinct mixtures were available to perform the delay estimation. Apart from conducting two separate recordings, there is no way to obtain $m_1[n]$ and $m_2[n]$. Instead the individual bursts that make up the received signal are seen as mixtures that contain both the signals mixed in some ratio with noise added to the mixtures. Seen in this light, there are 4 mixtures in the received signal shown in Fig. 5.26. This problem is therefore identical to the example treated in Section 5.4.2. Figure 5.28 shows the received mixture with the individual mixtures indicated.

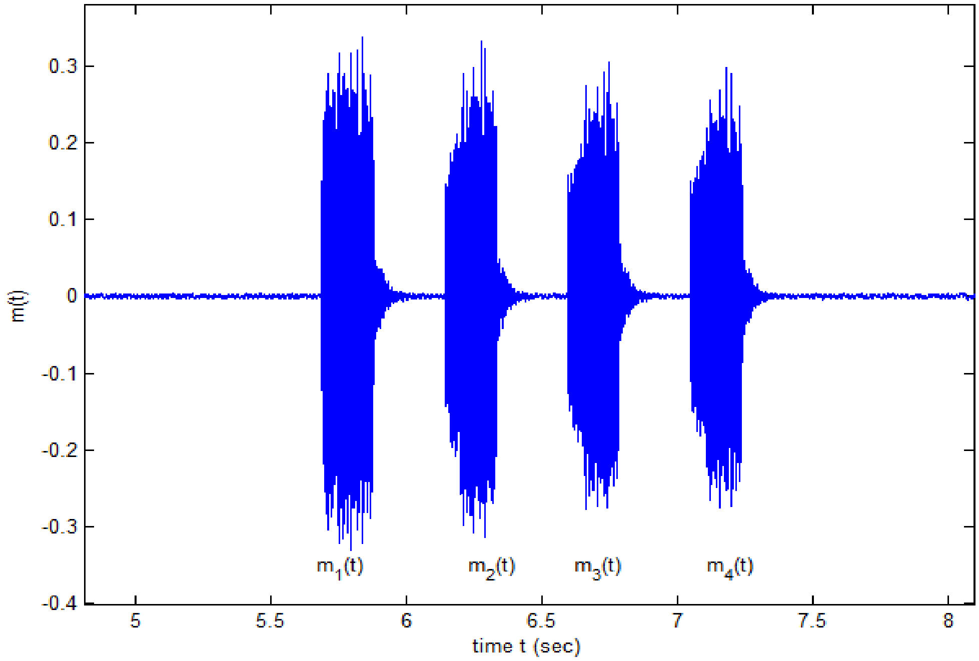


Fig. 5.28: Individual mixtures indicated.

From each recording, it is possible to identify four pulses and six unique pairs of mixtures and therefore six delay estimates may be obtained from each recording. One possible grouping could be one mixture consisting of the first two pulses and the other mixture consisting of the remaining two pulses. Yet another grouping would have $m_1[n]$ and $m_3[n]$ as one pair and $m_2[n]$ and $m_4[n]$ as another. In the remainder of this chapter, $m_1[n]$ and $m_2[n]$ are treated as one pair and the other pair consists of $m_3[n]$ and $m_4[n]$.

The recording experiment was conducted with one microphone placed at a distance of 36 cm from the speaker and the other placed at a distance of 80 cm from the speaker. The extra time taken by $s[n]$ to reach the farther microphone is, assuming the speed of sound in air is 341.2 m/s,

$$t_{extra} = \frac{0.80 \text{ m} - 0.36 \text{ m}}{341.2 \text{ m/s}} = 0.00129 \text{ s} \tag{5.28}$$

At a sampling frequency $F_s = 44.1\,\text{kHz},$ this translates to a delay of

$$\hat{D}_{theory} = 0.00129 * 44.1e3 = 56.89 \text{ samples} \tag{5.29}$$

The duration of each pulse in the MATLAB-generated burst shown in Fig. 5.25 before transmission is 0.25 s. An allowance of 0.15 s was made for the reverberation of the room and the individual mixtures were isolated by employing a time-window of 0.4 s width in time. With a sampling frequency of 44.1 kHz, this means a window consisting of 17640 time-samples. The start of any of the mixtures $m_1(t)$, $m_2(t)$, $m_3(t)$ or $m_4(t)$ may be identified by a steep transition. The window is chosen such that it incorporates this transition and starts before it, so that there is noise included in the window as well. With the spectrogram being generated with a Hanning window of size 32 with an overlap of 1 time-sample, the extent in time of the spectrogram of the mixtures is 17640-32=17608. The mean function $\overline{\alpha(j)}$ was generated with the same parameters, viz. W=6 and $\Delta = 3$. Figure 5.29 shows a zoomed version of $\overline{\alpha(j)}$.

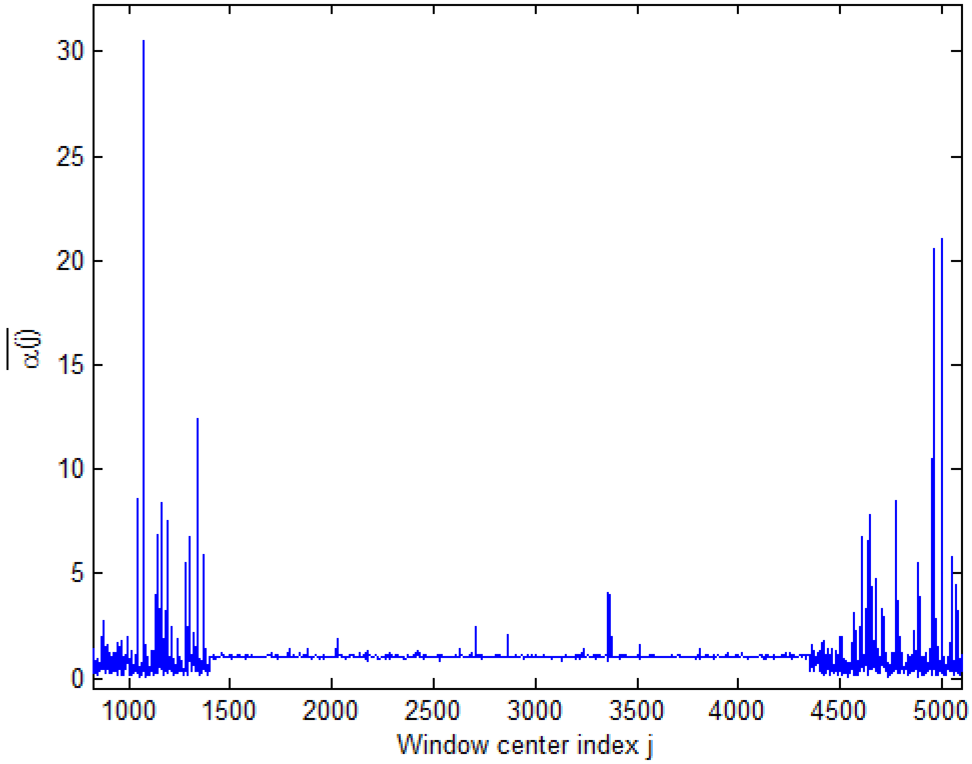


Fig. 5.29: Mean function $\overline{\alpha(j)}$ for mixtures $m_1[n]$ and $m_2[n]$.

Two trends are visible in Fig. 5.29: high variability at the beginning and end and relatively low variability in the middle. The high variability at the two ends ($1400 \le j \le 1450$ and $4500 \le j \le 5000$) of the figure arises out of the ratio of the spectrograms of the intervals in the windowed pulses where only noise was present. The low variability in the middle ($1500 \le j \le 4000$) is due to the ratio of the spectrograms of those regions in $m_1[n]$ and $m_2[n]$ where either $s_1[n]$ or $s_2[n]$ or both $s_1[n]$ and $s_2[n]$ are active.

The value of $\overline{\alpha(j)}$ immediately before the noisy region, which in this case is around $j = 4450$, is taken as $\hat{\rho}$. The recovered signals $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$ on the two paths are shown in Fig. 5.30. A zoomed-in version is shown in Fig. 5.31, in order to aid a physical inspection and verification of the results of the TFRM method.

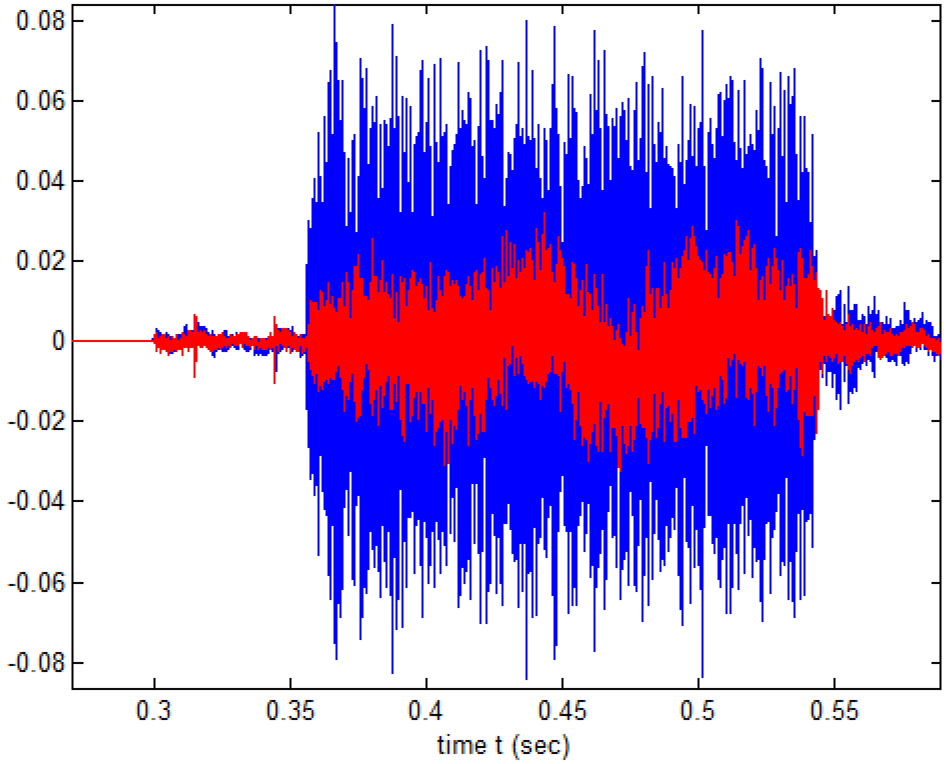


Fig. 5.30: Estimates $\hat{s}_1[n]$ (blue) and $\hat{c}_1\hat{s}_2[n]$ (red).

Using $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$, a search is performed for a better delay estimate. The contour of the search function is shown in Fig. 5.32. The minimum in this case was found to be at $\hat{D} = 56.56$. The delay $\hat{\tau}$ can be theoretically found as

$$\hat{\tau} = \frac{\hat{D}}{F_s} = \left(\frac{56.56}{44.1}\right)10^{-3} = 1.2825 \text{ ms} \tag{5.30}$$

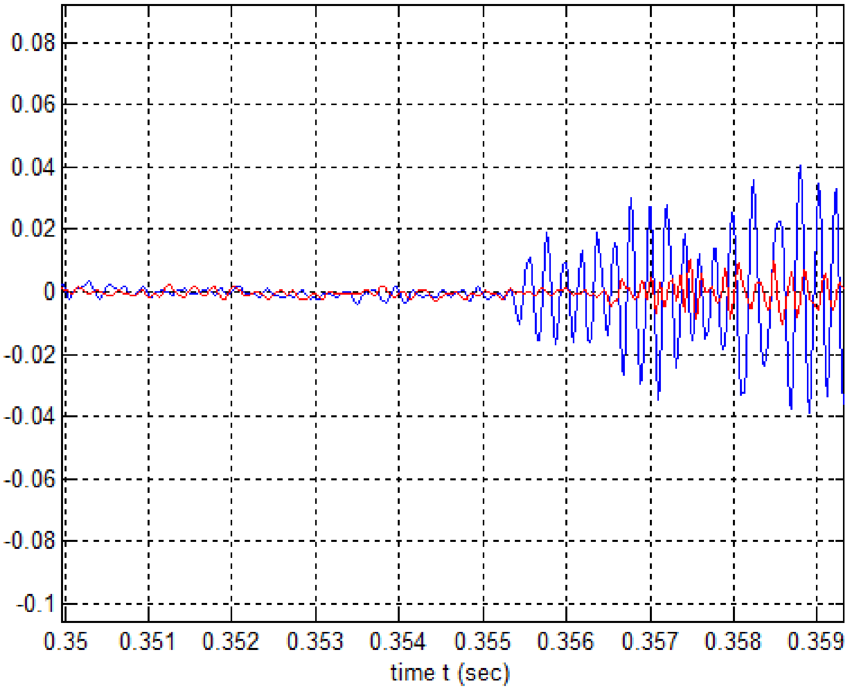


Fig. 5.31: Depiction of the delay between $\hat{s}_1(t)$ and $\hat{c}_1\hat{s}_2(t)$ for $m_1(t)$ and $m_2(t)$.

In Fig. 5.31 above, $\hat{s}_1(t)$ can be seen to start at t = 0.3552 s and $\hat{c}_1\hat{s}_2(t)$ can be seen to start at around t = 0.3565 s. The delay between the two signals is $\hat{\tau} = 0.0013$ s or 1.3 ms. $\hat{\tau}$ is an estimate for the true delay between the two signals. Translating this into samples, we obtain

$$\hat{D} = \hat{\tau} F_s = 0.0013 * 44.1 * 1e3 = 57.33 \text{ samples} \tag{5.31}$$

The delay estimate obtained from (5.28) is less than the visual estimate of 1.3 ms and this could be because of an error in isolating exactly when a signal begins. The onset of a signal is hard to determine precisely by simply a visual examination of the signal bursts.

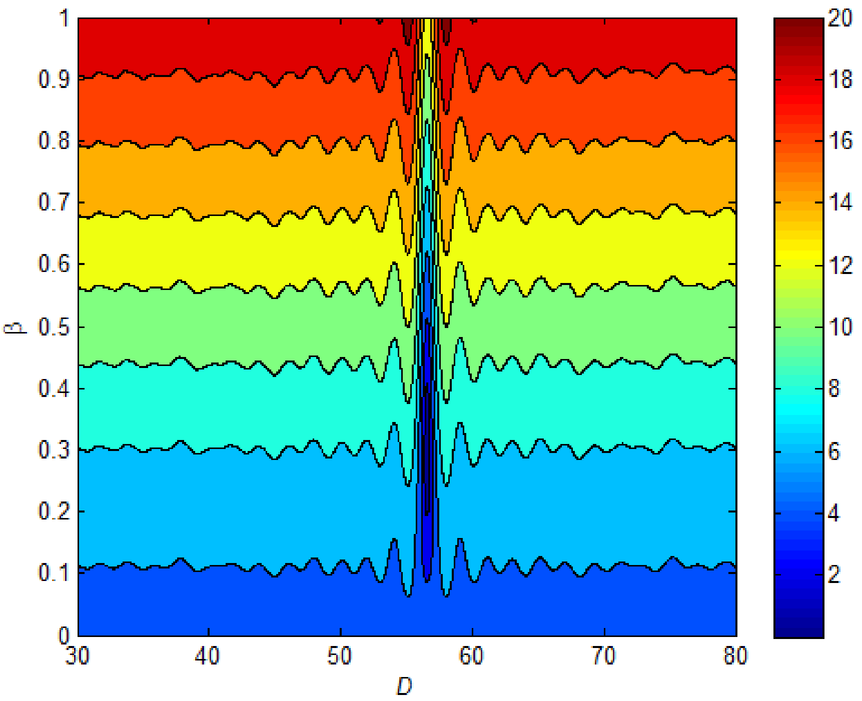


Fig. 5.32: Contour of error surface $e(\beta, D)$.

The delay obtained by a visual inspection of the two estimated signals in Fig. 5.31 is 57.33 samples, and the minimum of the contour surface $e(\beta, D)$ was found to be at $D = 56.56$ samples. The theoretical delay calculated by measuring distances and using physics was $\hat{D}_{theory} = 56.89$ samples.

The procedure above was also employed on the mixtures $m_3[n]$ and $m_4[n]$ and another delay estimate obtained. The procedure on this recording was repeated for ten other recordings at the same inter-microphone separations, i.e. $\hat{D}_{theory} = 56.89$, and a total of twenty delay estimates was obtained. These twenty delay estimates, with the variance indicated, are shown in Fig. 5.33.

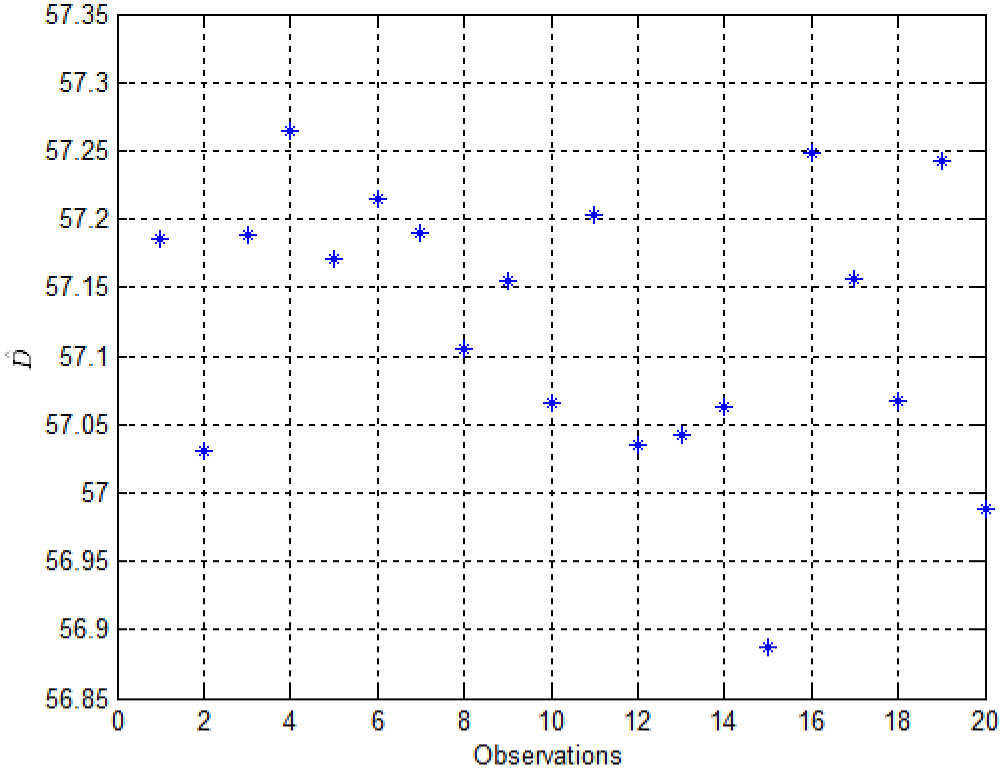


Fig. 5.33: Delay estimates for $D_{true} = 57.13$, $\sigma_D^2 = 0.0825$ at $SNR_{dB} = 37.1340$ dB.

The variance of the delay estimates shown above is 0.0825 $samples^2$ and the theoretical variance of the delay estimates returned by the TFRM method at $SNR_{dB} = 35$ dB as obtained by squaring the standard deviation is 0.0625 $samples^2$ (Fig. 5.23).

5.6.2.2 Case 2 $\hat{D}_{theory} = 123.48$

The inter-microphone separation was increased so that the microphone that was formerly 80 cm distant from the source was now at a distance of 131 cm from the source. The nearer microphone's distance was kept unchanged. The extra time that $s_2(t)$ travels is now:

$$t_{extra} = \frac{1.31 \text{ m} - 0.36 \text{ m}}{341.2 \text{ m/s}} = 0.0028 \text{ s} \qquad (5.32)$$

When converted into samples, the extra time becomes:

$$\hat{D}_{theory} = t_{extra} * F_s = 0.0028 * 44.1e3 = 123.48 \text{ samples} \quad (5.33)$$

The procedure performed for Case 1 presented in Section 5.6.2.1 was repeated for Case 2.

Figure 5. 34 shows the received signal consisting of four mixtures. The individual mixtures were isolated as explained in Section 5.6.2.1, i.e. by identifying the start of the mixtures $m_1[n]$, $m_2[n]$, $m_3[n]$, and $m_4[n]$ and by appropriately choosing the width of the time-window viz. 0.4 s.

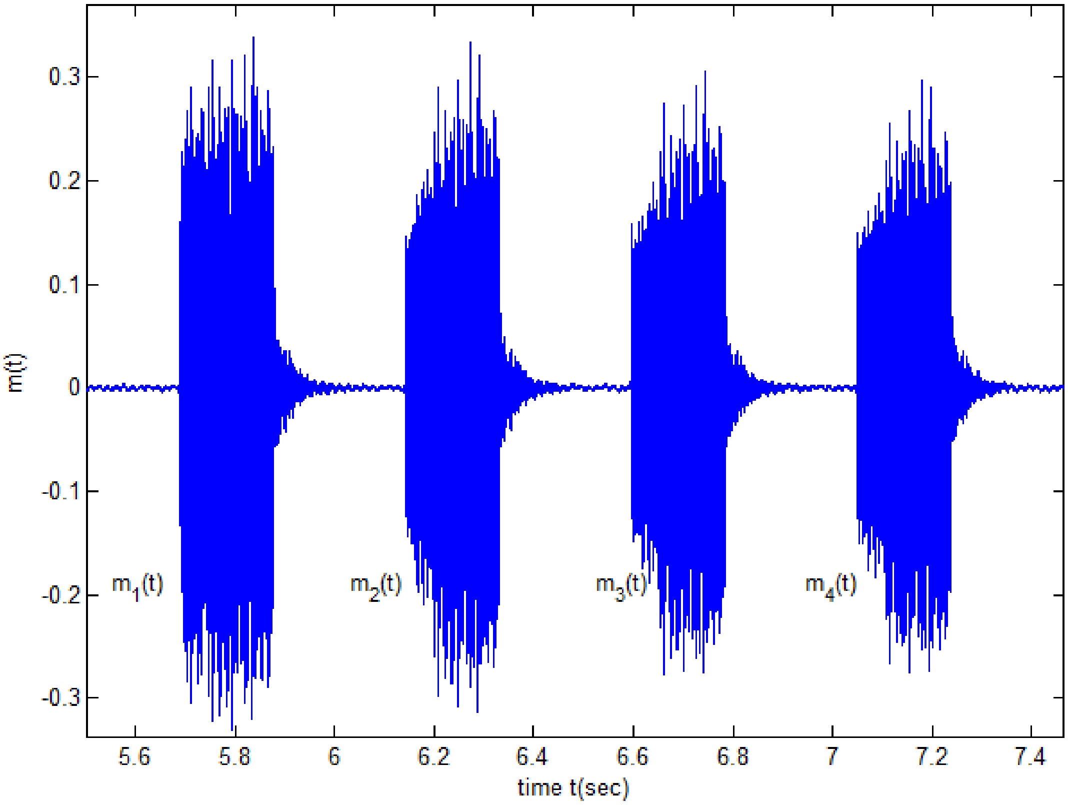


Fig. 5.34: Received mixture $m(t)$ for $\hat{D}_{theory} = 123.48$.

The mixtures $m_1[n]$ and $m_2[n]$ were chosen for the delay estimation. The extent of $\overline{\alpha(j)}$ is the same as in Case 1 since the values of all parameters are identical to those in Section 5.6.2.1. A zoomed version of $\overline{\alpha(j)}$ is shown in Fig. 5.35.

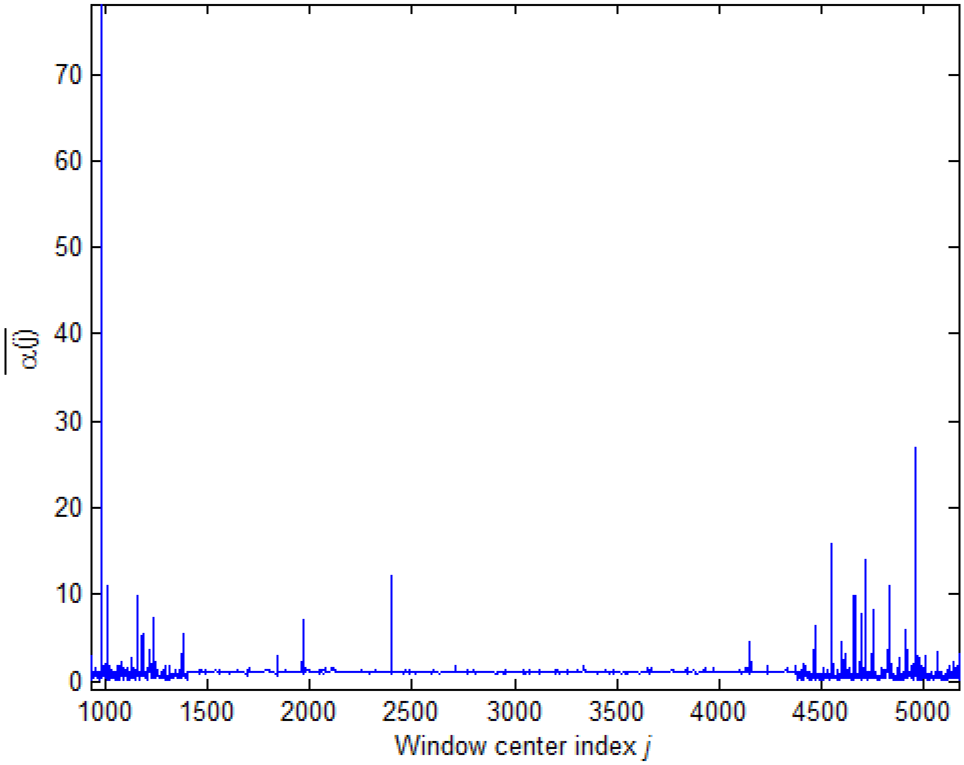


Fig. 5.35: $\overline{\alpha(j)}$ for $\hat{D}_{theory} = 123.48$.

As before, there is comparatively high variability at the beginning and end of the section of $\overline{\alpha(j)}$ shown in Fig. 5.35. This stems from the same reasons mentioned in the explanation of Fig. 5.29, i.e. noise at the beginning and end of the windowed pulse. The two signals reside and interact in the region $1500 \leq j \leq 4450$. In the above example, $\Delta\overline{\alpha(j)}$ assumes high values towards the beginning and end of the window center indices for which $\overline{\alpha(j)}$ is defined. The region of interest is identified by the relatively low value of the derivative $\Delta\overline{\alpha(j)}$ in that region. The value of $\overline{\alpha(j)}$ before it assumes the value of the ratio of the spectrograms of the noisy regions in the original mixtures, i.e. $j = 4200$ is taken as $\hat{\rho}$. $\overline{\alpha(j)}$ exhibits greater variability in the region for which $j \in (1500, 4300)$ where both the signals are present. The signal $s_2[n]$ travels a greater distance (131 cm) than before (80 cm). Consequently $s_2[n]$ undergoes greater attenuation and is therefore harder to distinguish from the noise. Figure 5.36 gives a clearer idea of the weakness of $s_2[n]$ where, as before, both $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$ are shown.

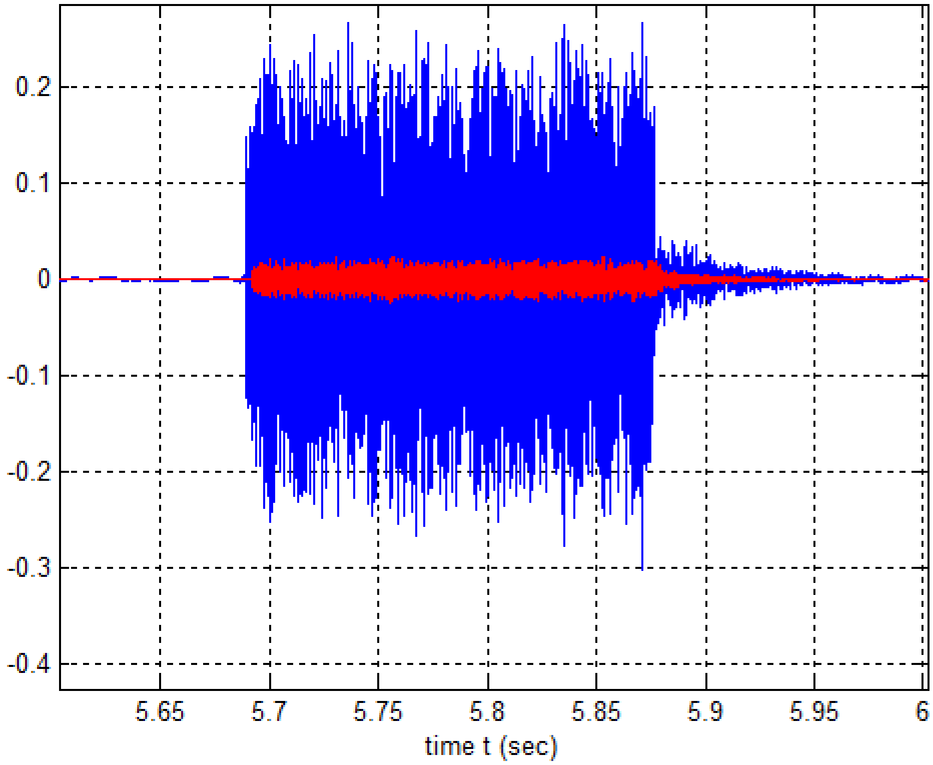


Fig. 5.36: $\hat{s}_1[n]$ (blue) and $\hat{c}_1\hat{s}_2[n]$ (red) estimated from the TFRM method.

A zoomed-in version of the above figure is shown in Fig. 5.37 to give a clearer picture of the delay between the estimated signals $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$.

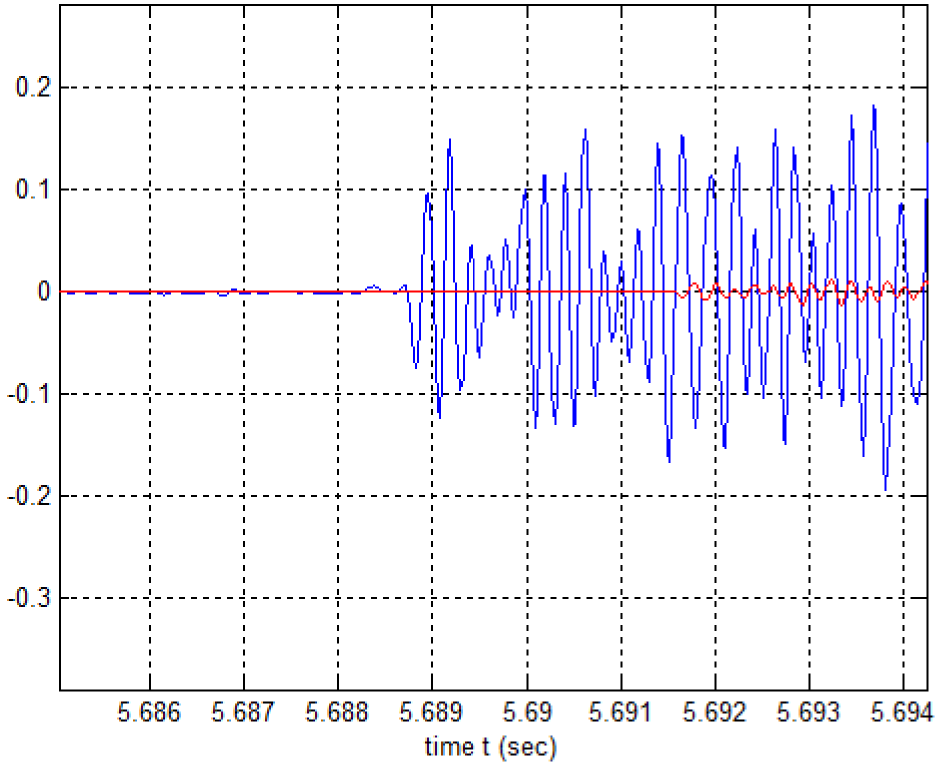


Fig. 5.37: Zoomed version of Fig. 5.30 showing the delay.

The signal $\hat{s}_1[n]$ is observed to start at approximately $t = 5.6885$ s while the signal $\hat{c}_1\hat{s}_2[n]$ starts at approximately $t = 5.6915$ s. The start is identified by the first significant deviation from 0 and seen this way, the delay between the two is about 0.003 s. With the delay estimate obtained by visual inspection, it is possible to calculate the delay in samples as:

$$\hat{D} = \hat{\tau} F_s = 0.003 * 44.1 * 1e3 = 132.3 \text{ samples} \tag{5.34}$$

With these two estimates for $\hat{s}_1[n]$ and $\hat{c}_1\hat{s}_2[n]$ the search for the true delay was performed and the result is shown in Fig. 5.38.

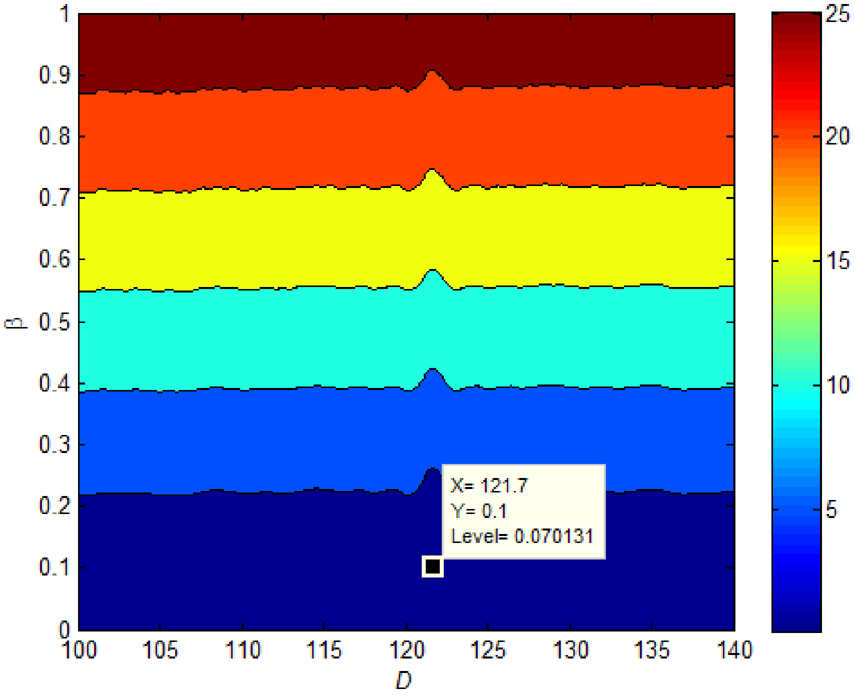


Fig. 5.38: Contour plot of $e(\beta, D)$.

The search for the true delay reveals that the global minimum is estimated to occur at $\hat{D} = 121.7\text{ samples}$, which is not exactly equal to $\hat{D}_{theory} = 123.48\text{ samples}$. The difference can be attributed to inaccuracies in estimation arising out of noise in the environment corrupting the signal.

As the microphone farther from the source becomes more and more distant, the signal $s_2[n]$ becomes more and more diminished, i.e. its variance will be closer and closer to, or even below, that of the noise. Two estimates from the TFRM method were collected for each recording by pairing up $\{m_1[n], m_2[n]\}$ and $\{m_3[n], m_4[n]\}$. The position of the farther microphone with respect to the source was varied while the nearer microphone was kept fixed at a distance of 36 cm from the source. The distances of the farther microphone from the source were 70 cm, 76 cm, 80 cm, 131 cm, and 151 cm. Ten experiments were conducted at each of the distances of the farther microphone and the resulting twenty delay estimates were stored in a matrix for the respective distance. Then the variance of the delay estimates was estimated for each of the distances and the behavior of the variance of the delay estimates with respect to distance is shown in Fig. 5.39.

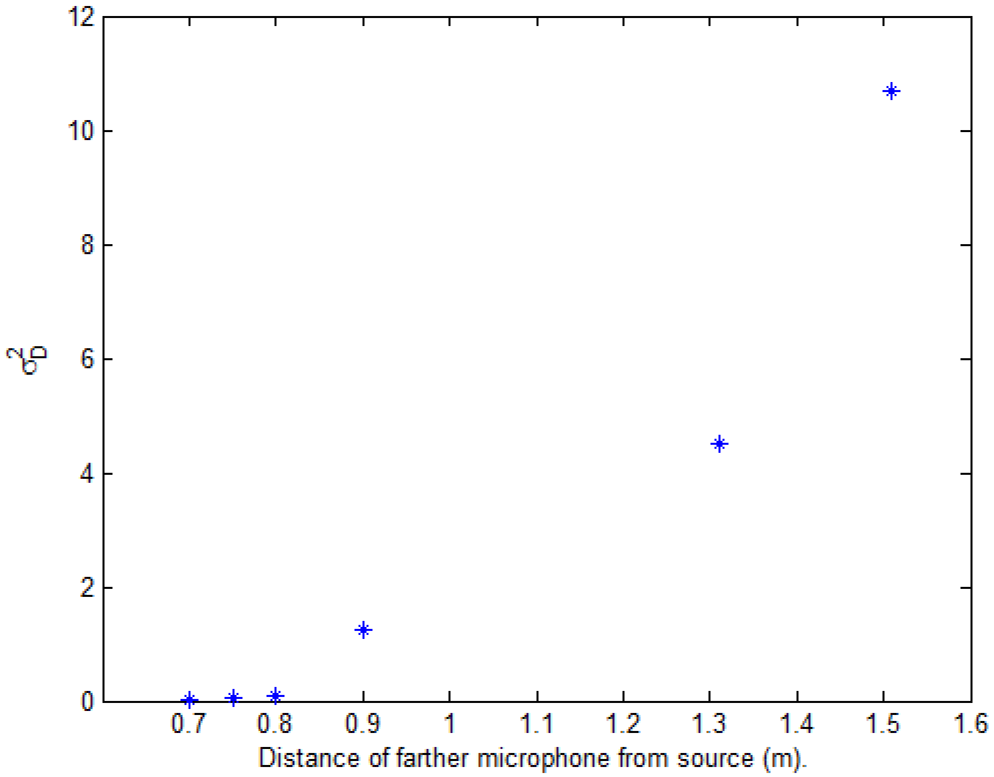


Fig. 5.39: σ_D^2 versus distance to the source for the farther microphone.

The variance shows a non-linear increase with increasing separation between the farther microphone and the source. As the distance from the farther microphone to the source is increased, the signal $s_2(t)$ gets weaker due to greater attenuation. With greater attenuation of $s_2(t)$, it becomes more difficult for the TFRM algorithm to distinguish between $s_2(t)$ and noise and so the reliability of the estimator diminishes. The lack of reliability at greater distances from the source of the farther microphone manifests itself in an increase in the variance of the delay estimates. At the extreme case of a separation of 151 cm between the microphone that is being moved and the source, the TFRM method fails to find the right delay with any sort of reliability and returns delay estimates with a variance of nearly 10. The search for the true delay takes place in an interval 10 samples in size in steps of 0.01. The test for the correct functioning of the estimator is that the range of the search must be large enough to contain estimates that fall within 3 times the standard deviation. With a search interval of size 10, this means:

$$\begin{aligned} &3\sigma \leq 10 \\ &\sigma \leq \frac{10}{3} \end{aligned} \tag{5.35}$$

Squaring (5.34) we obtain:

$$\sigma^2 \leq 11.01 \tag{5.36}$$

Based on this criterion, the TFRM estimator fails when the second microphone is placed at a distance of 151 cm from the source.

The dependence of the variance of the delay estimates on the distance of the farther microphone from the source can also be expressed as a dependence between variance and SNR_{dB} since a larger distance between the source and the farther microphone means a smaller signal level. The distances 0.7 m, 0.75 m, 0.8 m, 0.9 m, 1.3 m, 1.5 m are converted into equivalent SNR_{dB} and the σ_D^2 is plotted versus SNR_{dB}. This is plotted along with the theoretical estimates for the variance shown in Fig. 5.23.

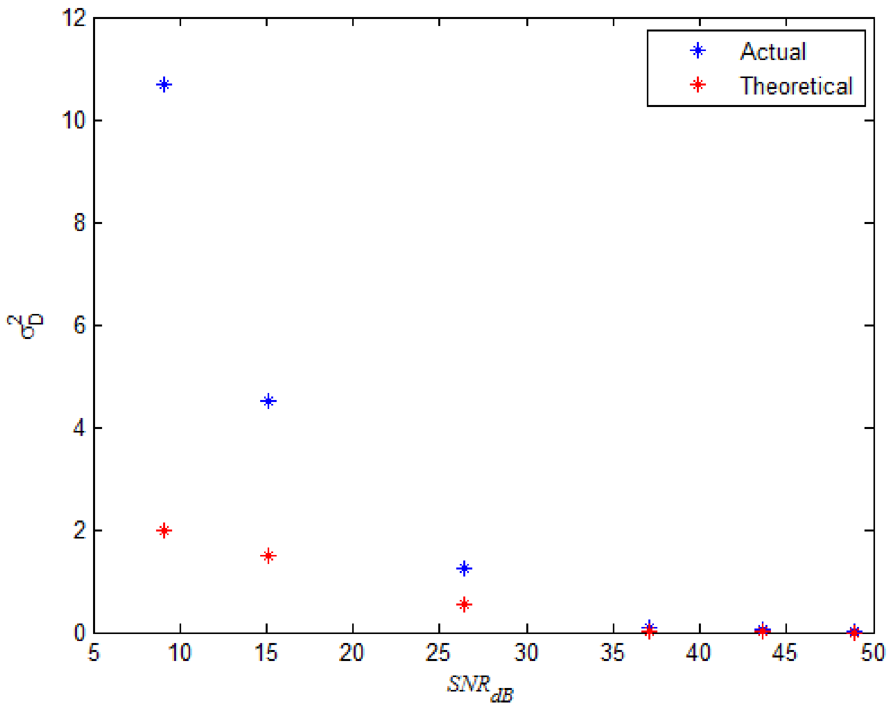


Fig. 5.40: σ_D^2 versus SNR_{dB}.

It can be seen that that the difference between the theoretically predicted variance estimates and the practically achieved variance estimates narrows as SNR_{dB} increases. The explanation of what is observed in Fig. 5.40 is that the TFRM estimator gets better as the strength of the received mixture increases. Up to a distance of 0.9 m between the source and the farther microphone, there is practically no difference between the theoretically predicted and practically derived results. When the distance between microphone 2 and the source is 0.9 m or, equivalently, when $SNR_{dB} = 26.43\text{ dB}$, some deviation from the theoretically predicted variance is observed and this gap widens as the farther microphone is taken farther and farther away from the source until failure is reached at $SNR_{dB} = 9.056\text{ dB}$.

In simulated and controlled conditions, it is possible to achieve acceptable performance for the entire range of SNR_{dB} from 0 dB to 40 dB. However, reality is different. The TFRM estimator fails when $SNR_{dB} = 9.056\text{ dB}$ and is far from the theoretically predicted variance up to

$SNR_{dB} = 26.43\text{ dB}$. Performance close to what is predicted by theory is only achieved at $SNR_{dB} \geq 35\text{ dB}$.

5.7. The Performance of the TFRM estimator

In Section 5.6, the dependence between the variance of the delay estimates returned by the TFRM estimator and the signal-to-noise ratio of the received mixture was studied. An unbiased estimator is better than a biased one, because at least in the mean, the unbiased estimator is accurate. An estimator that returns estimates of low variance is more desirable than an estimator of high variance. Is it possible to design an estimator of arbitrarily low variance? How does the variability (or the variance of its estimates) of an estimator depend on the strength of the received signal? The answer to these questions is intimately linked to the Cramer-Rao bound.

The Cramer-Rao bound is named for Calyampudi Radhakrishna Rao and Harald Cramer, the original discoverers of the relationship. The theoretical lower bound on the variance of a delay estimator has been derived by Mark Fowler of Binghamton University (email id: mfowler@binghamton.edu) in connection with EECE 522: Estimation Theory and has been reproduced in the Appendix [40]. The Cramer-Rao bound, in its original form, states that variance of an estimator is lower bounded by the inverse of the Fisher information [41]. An estimator that achieves this lower bound is said to be fully efficient. A fully efficient estimator achieves the lowest possible mean-square error among all unbiased estimators and, hence, such an estimator is called the minimum variance unbiased (MVU) estimator.

The formula derived for the Cramer-Rao bound in the Appendix was used to obtain the Cramer-Rao bounds for the variance of the estimator at the signal-to-noise ratios considered in previous experiments. The bounds are shown along with the actual variances of the estimates at those signal-to-noise ratios in Fig. 5.41.

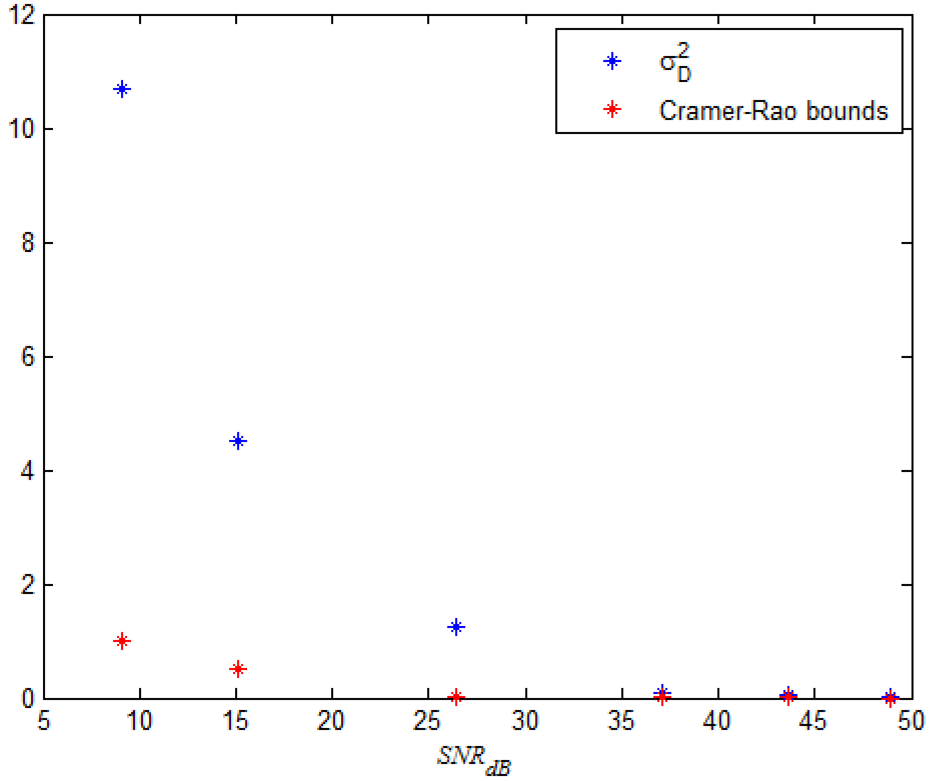


Fig. 5.41: Noise performance of the TFRM estimator.

The Cramer-Rao lower bound decreases with increasing SNR. This is obvious from a direct inspection of the formula for the Cramer-Rao bound derived in the Appendix: the theoretically predicted lower bound for the variance of an estimator is inversely proportional to the SNR of the signal. The actual variance of the estimator is much higher than the theoretically predicted variance at lower values of SNR_{dB} but this discrepancy decreases with increasing SNR_{dB}, with the estimator practically becoming fully efficient for $SNR_{dB} \geq 35\text{ dB}$.

5.8. Application to Source Localization

The TFRM estimator provides an estimate of the delay between echoes contained in a received burst. The shift, measured in samples, between the signals received by the two microphones is a measure of the difference in the distances traveled by those signals in reaching the two sensors.

With knowledge of how much longer than the path from source to Rx_1 the path from source to Rx_2 is, along with information about the precise location of the microphones, it is possible to arrive at a locus for the source. The source is a point which moves such that the difference of its distances from two fixed points, in this case the microphones, is a constant. The locus defines a hyperbola.

5.8.1 Locus for $\hat{D}_{theory} = 57.13$

The two microphones were placed at a distance of 0.56 m from each other. The position of the microphone Rx_2 is arbitrarily chosen as the origin and Rx_1 is arbitrarily chosen as $(0, 0.56)$. The TFRM method estimated the delay to be 56.56 samples. This delay estimate is equivalent to 1.3 ms. In 1.3 ms, sound travels a distance of 0.44 m through air, at a speed of 341.2 m/s. Applying the distance formula, and denoting the location of the source as (x, y), we can derive the locus for the source as:

$$\sqrt{x^2+y^2}-\sqrt{x^2+(y-0.56)^2}=0.44 \tag{5.37}$$

Transposing and squaring both sides,

$$\begin{aligned}
&x^2+y^2=0.1936+x^2+(y-0.56)^2+0.88\left(\sqrt{x^2+(y-0.56)^2}\right)\\
&0=0.1936+0.56^2-1.12y+0.88\left(\sqrt{x^2+(y-0.56)^2}\right)\\
&1.12y-0.5072=0.88\left(\sqrt{x^2+(y-0.56)^2}\right)\\
&0.48y^2-0.2688y+0.0144=0.7744x^2\\
&(y-0.28)^2-0.0484=1.6133x^2
\end{aligned} \tag{5.38}$$

There are two possible expressions for y, namely:

$$\begin{aligned}
y&=\sqrt{1.6133x^2+0.0484}+0.28\\
y&=-\sqrt{1.6133x^2+0.0484}+0.28
\end{aligned} \tag{5.39}$$

Both the curves are shown in Fig. 5.42, with the individual microphones indicated:

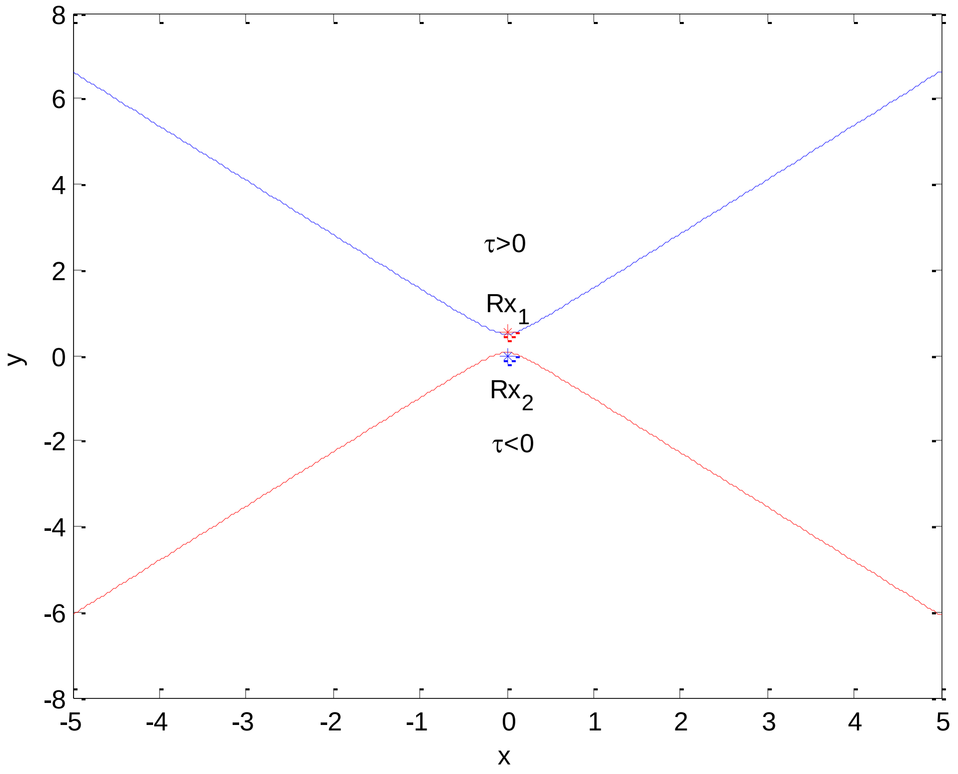


Fig. 5.42: Locus of the source.

The locus is a hyperbola intersecting the line connecting the two microphones. The TFRM method fails when the microphones are equidistant from the source because such an arrangement leads to identical signals being received at both microphones, i.e. with zero delay difference. Therefore the line $y = 0.28$ constitutes a blind region for the TFRM estimator, but not for other estimators capable of producing $\hat{\tau} = 0$.

The estimation of the signals $s_1(t)$ and $s_2(t)$ allows a determination to be made whether τ is positive or negative. A positive τ, according to the convention used in this thesis, implies that $s_2(t)$ is delayed with respect to $s_1(t)$ and our findings show that τ is indeed positive. With this knowledge, the source may be located in a narrower space, in the region labeled in Fig. 5.42 as $\tau > 0$. Figure 5.43 shows the locus of the source for only $\tau > 0$.

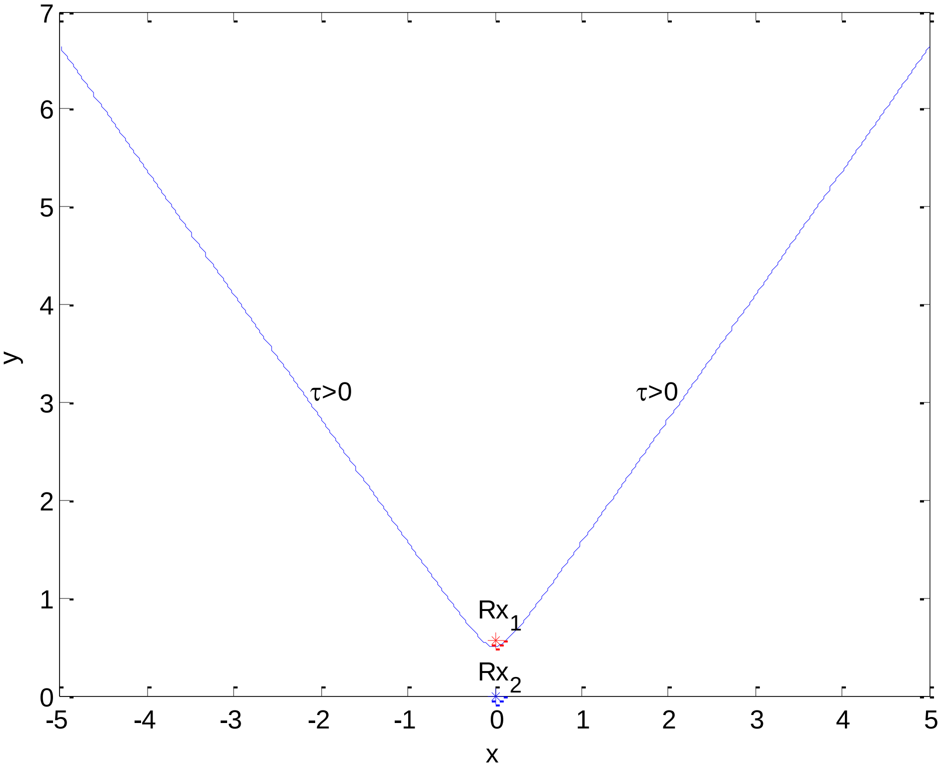
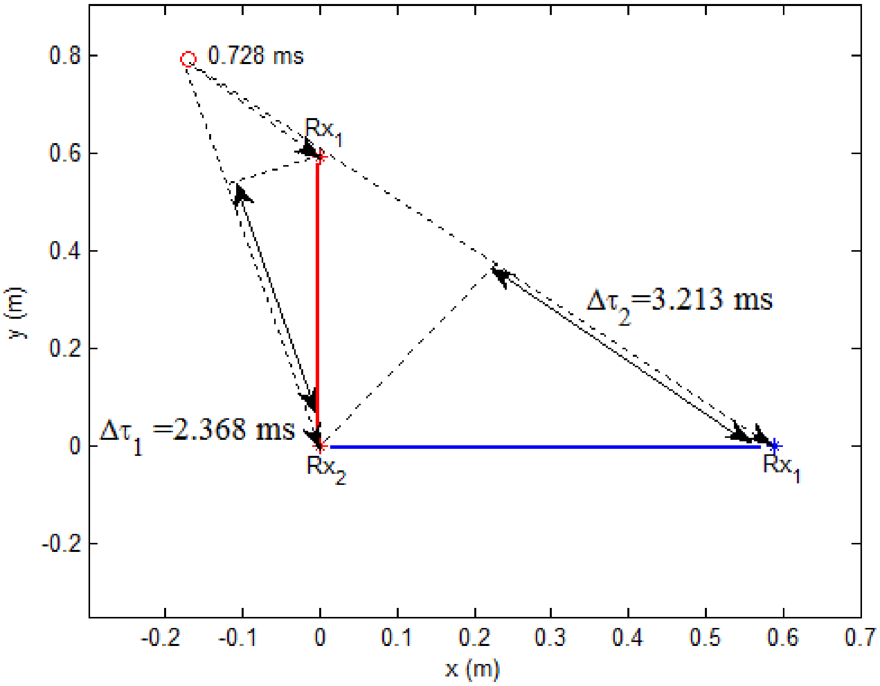


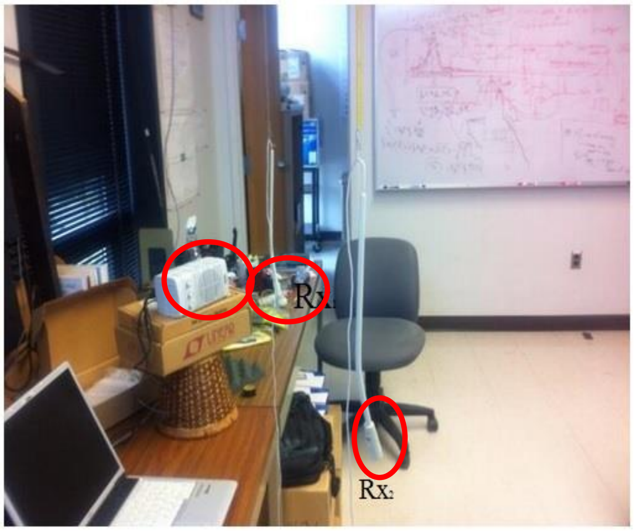
Fig. 5.43: Locus of the source for $\tau > 0$.

The choice of coordinate system used in the derivation of the locus places the source in a plane different from the x-y plane of Fig. 5.42 and Fig. 5.43. The source resides on a solid surface obtained by rotating the locus shown in Fig. 5.43 about the y-axis. Rather than stating that the source lies somewhere on one branch of a hyperbola, it is more correct to state that the source lies on a hyperboloid whose projection on the 2D-plane is the hyperbola in Fig. 5.43.

As seen in Fig. 5.43, there is potentially an infinite number of possible locations for the source. By moving either or both microphones to a different location, another locus could be obtained and the loci for the two arrangements of the microphones would produce a finite list of possible locations for the source. To have a convenient reference plane, the level of the speaker is raised in order to bring it on a plane parallel to the floor and containing the microphones Rx_1 and Rx_2. The microphone Rx_2 is kept fixed and the microphone Rx_1 is moved to two separate locations from the location in Fig. 5.43. So, apart from the locus in Fig. 5.43, two other loci are obtained corresponding to the two locations of Rx_1. Rx_1 is moved first from (0, 0.59 m) to (0.59 m, 0) and then to (0.15 m, 0.85 m). A snapshot of these two arrangements coupled with a graph showing their locations is shown Fig. 5.44 and Fig. 5.45.

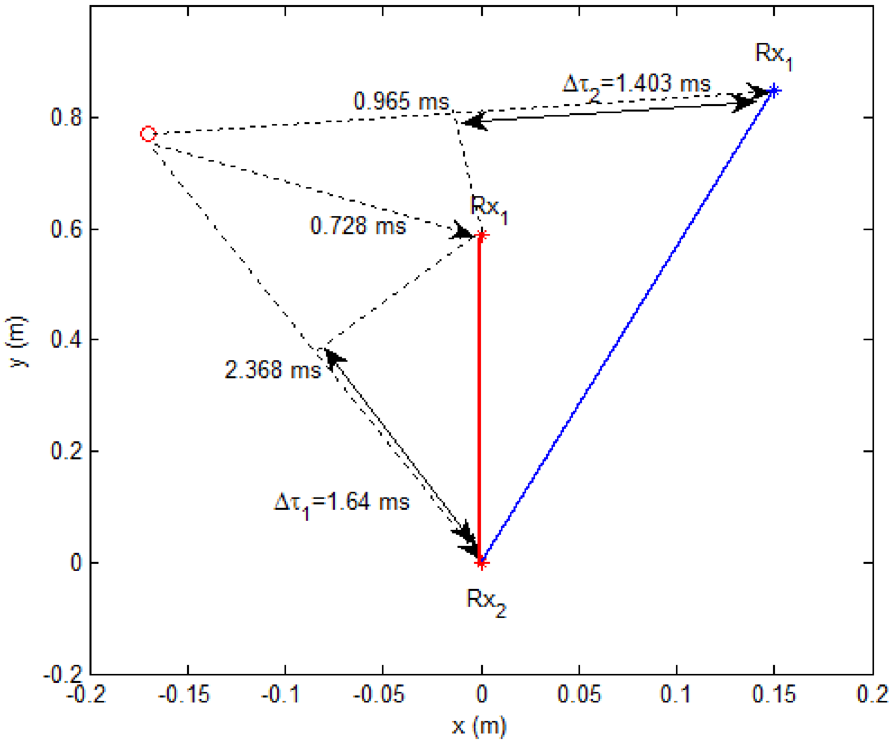


(a)

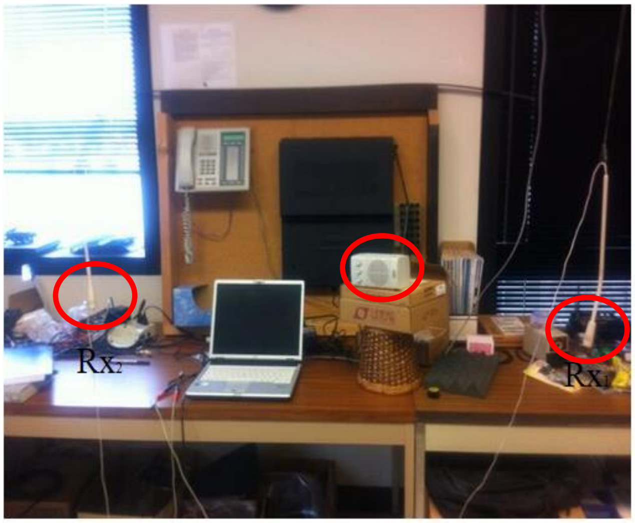


(b)

Fig. 5.44: (a) Theoretical delays for arrangement 1; (b) Image of arrangement 1.



(a)



(b)

Fig. 5.45: (a) Theoretical delays for arrangement 2; (b) Image of arrangement 2.

The estimated time-difference-of-arrivals for arrangement 1 were $\Delta\hat{\tau}_1 = 1.581\,\text{ms}$ and $\Delta\hat{\tau}_2 = -0.866\,\text{ms}$. The estimated time-difference-of-arrivals for arrangement 2 were $\Delta\hat{\tau}_1 = 1.581\,\text{ms}$ and $\Delta\hat{\tau}_2 = 1.502\,\text{ms}$. The loci obtained for the arrangement in Fig. 5.44 are shown in Fig. 5.46 and the loci obtained for arrangement 2 are shown in Fig. 5.47. Zoomed versions of Fig. 5.46 and Fig. 5.47 are shown in Fig. 5.48 and Fig. 5.49.

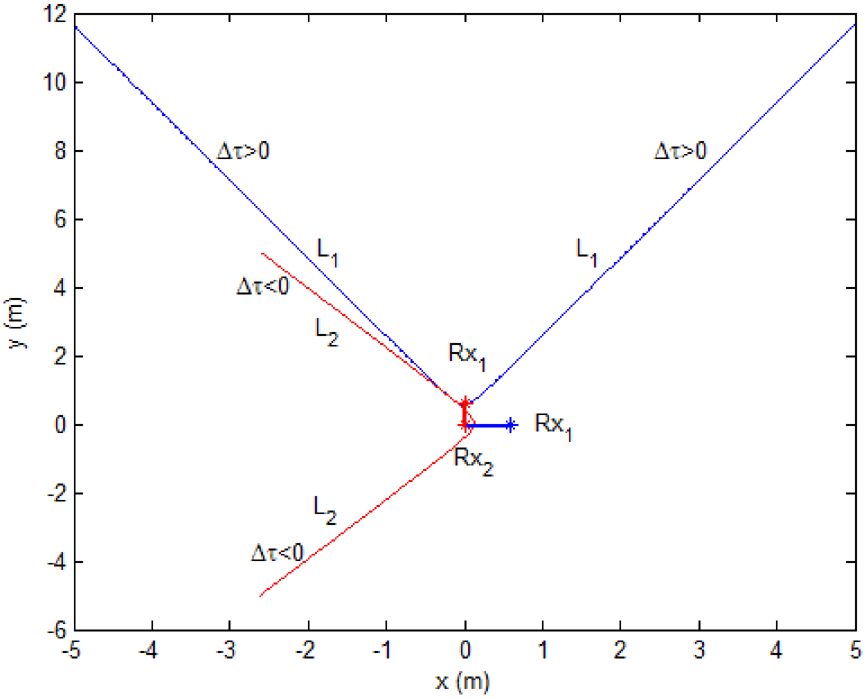


Fig. 5.46: Loci obtained for arrangement 1.

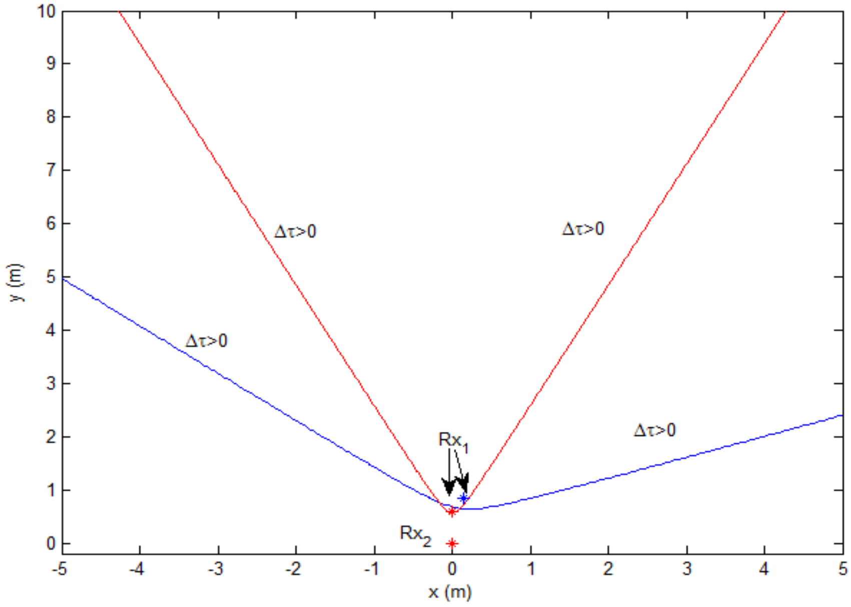


Fig. 5.47: Loci obtained for arrangement 2.

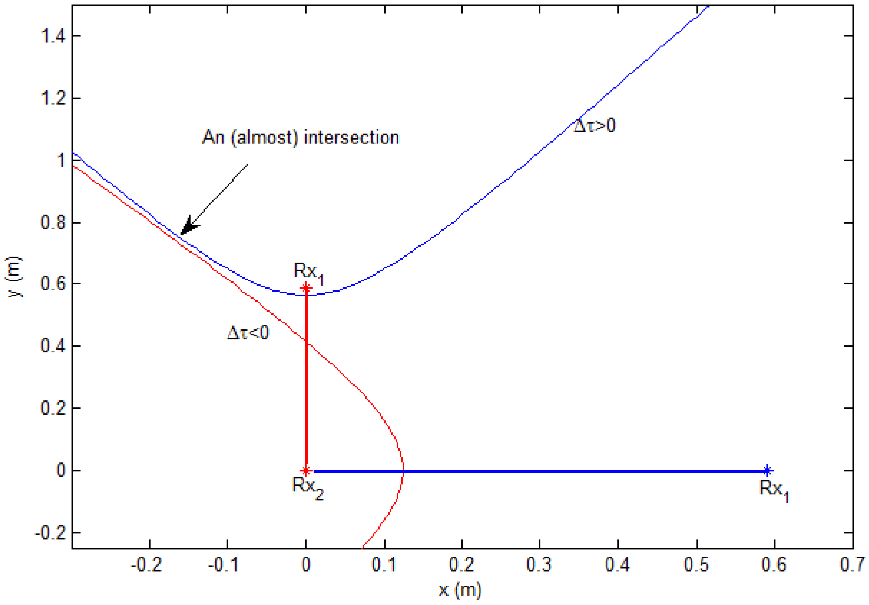


Fig. 5.48: Zoomed view of Fig. 5.46

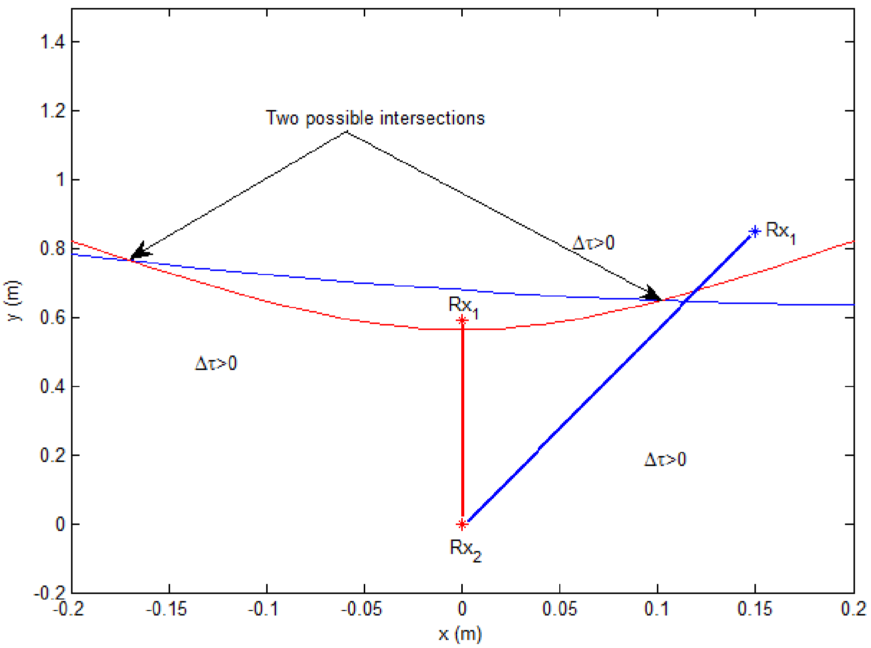


Fig. 5.49: Zoomed view of Fig. 5.47.

From Fig. 5.48, it is clear that while the red curve approaches the blue curve closely, it does not exactly intersect. The two loci in Fig. 5.48 are sensitive to slight errors in the estimation of the time-delay by the TFRM algorithm, compounded by the geometry of the arrangement. In Fig. 5.49, the second position of the microphone Rx_1 is closer to the source than the corresponding position in arrangement 1 and this plays a part in making the TFRM delay estimate more accurate. In addition, the geometry of the arrangement now causes the branches of the loci indicated in red and blue to intersect at a more acute angle.

6. Conclusions and Future Work

An algorithm to identify single-signal active regions is applied to the problem of delay estimation to help in source localization. Instead of relying on the output from multiple sensors or multiple versions of the transmitted signal, the algorithm accepts as input a single mixture, containing echoes of the transmitted signal. The sampling rate employed at the receiver is not high enough for the delay, measured in samples, to consist of only an integer component and a negligible fractional part. The proposed algorithm is successful in estimating the delay to a tenth of the sampling interval, where the sampling interval is $22.67\ \mu s$.

The proposed algorithm is termed the time-frequency ratio of mixtures (TFRM) method. In the main, the method aims to eliminate the superimposed signals in the received mixture by using two independent instantiations of the mixture. Prior to the cancellation of the individual signals, the ratio in which they combine must be estimated. The ratio is estimated in the time-frequency domain by first defining the ratio of spectrograms of the two instantiations of the mixture. The ratio function exhibits high variability during those regions in time when both the signals mix and relatively low variability in single-source regions. In order to smooth out local variations, a windowed sample mean of the mean function is defined and analyzed for low-variability regions. Subsequent to the estimation of the ratio, estimates for the signals received at the microphones are determined by source separation. With the knowledge of the estimates for signals received at the microphones, a search for the true delay is performed.

The TFRM algorithm has a structural deficiency: it fails when the mixtures are exactly identical. This is because the mean function is exactly one throughout and is then useless in providing information about the location of the echoes. However, the mixtures need only be slightly different for the successful functioning of the algorithm. Consequently, a blind spot of the TFRM algorithm is the perpendicular bisector of the line joining the sensors. When the speaker is equidistant from the sensors, the signals received on each of the sensors are identical and synchronous with respect to each other. In the absence of any significant delay between the signals received at the sensors, source localization using the proposed method becomes impossible..

The noise performance, measured by the dependence of the variance of the estimates returned by the TFRM time-delay estimator on the strength of the observations, is better in simulation than in actuality. Regarding the signal-to-noise ratio, it must be noted that the only measurable signal-to-noise ratio is the combined SNR of the received mixture, consisting of the sum of the signals received at the microphones and noise. The combined SNR is definitely influenced by the strength of the signals at each of the sensors but is predominantly influenced by the strength of the stronger signal. Signal-to-noise ratio must be interpreted here as the signal-to-noise ratio of the received mixture. The deviation in performance from the theoretical is higher at lower values of SNR and practically negligible at values of SNR greater than 40 dB. The variance of the delay estimator obtained in experiments conducted in the DSPRL is compared to the Cramer-Rao bound. We found that at lower values of signal-to-noise ratio the estimator performed much worse than the Cramer-Rao inequality predicted, while at higher values of signal-to-noise ratio the performance closely matched the theoretical Cramer-Rao bound.

The delay estimates obtained are ultimately used to localize the source. It is important to note that only a locus for the source rather than an exact location can be achieved with a single sensor as described here. Changing the location of the pair of sensors to obtain another locus could provide an intersection point with which to pinpoint the source. Arriving at a locus for the source involved using the locations of the individual sensors.

The TFRM algorithm does not use any statistical information about the nature of the signal emitted by the source. Although illustrated only for OFDM signals, it is in theory applicable to any kind of signal. The property of not being dependent on *a priori* information about the source signal expands the applicability of the algorithm and can help locating any manner of transmitter.

In connection with the chirplet signal algorithm, it was mentioned that a single kernel could not be derived for an OFDM burst. An interesting direction of research, not pursued in the present work, would be the possibility of using multiple kernels to identify echoes in a data-bearing OFDM signal.

The TFRM algorithm failed at $SNR_{dB} = 9.056\text{ dB}$ and does not come close to the Cramer-Rao bound until *SNR* is about 36 dB. In terms of real-life scenarios, the algorithm would need to execute in a relatively calm and peaceful environment to give acceptable performance. The aim is to find a method of which the performance approaches or is close to the Cramer-Rao bound at lower values of the combined signal-to-noise ratio.

The performance of the TFRM estimator was observed to be best when the microphones were close to the speaker and there were relatively few reflecting surfaces in the vicinity of the sensors, thereby reducing multipath. As one of the microphones was placed farther and farther away from the source, the estimator became progressively worse. The deterioration in performance is speculated to be because of the diminishing level of the signal received at one of the sensors as well as the possibility of multiple components, apart from just the line-of-sight component, superposing at the farther microphone. Quantifying the effects of noise and multipath on the performance of the estimator is a question not addressed in this work but one that would go some distance in informing the estimator's applicability to high-multipath scenarios.

The algorithm may find application in an underwater acoustic OFDM environment, although that has not been evaluated here. Employment of an underwater source localization system based on the proposed algorithm would be an important contribution to the state of the art.

APPENDIX: CRAMER-RAO BOUND FOR TDE

This derivation is taken from Prof. Mark Fowler’s slides for his course EECE 522: Estimation Theory [40]. Consider a mixture

$$m(t)=s_1(t)+c_1s_1(t-\tau)+w(t) \tag{A.1}$$

where $w(t)$ is white Gaussian noise and c_1 is a constant between 0 and 1. In reality, the receiver processes discrete-time signals so sampling of (A.1) leads to:

$$m[n]=s_1(nT_s)+c_1s(nT_s-\tau)+w(nT_s) \tag{A.2}$$

where sampling is performed every T_s seconds. Let us assume that $m[n]$ has L_{sum} number of non-zero samples starting at $n=n_0$. Assuming an observation period of length N samples, $m[n]$ can be defined piecewise as:

$$m[n]=\begin{Bmatrix} w[n] & 0\le n\le n_0-1 \\ s_1[nT_s]+c_1s_1[nT_s-\tau]+w[n] & n_0\le n\le L_{sum}-1 \\ w[n] & L_{sum}\le n\le N-1 \end{Bmatrix} \tag{A.3}$$

According to the standard definition of the Cramer-Rao lower bound:

$$\operatorname{var}(\hat{\tau}_0)\ge\frac{\sigma^2}{\sum_{n=0}^{N-1}\left(\frac{\partial m[n]}{\partial\tau_0}\right)^2} \tag{A.4}$$

σ^2 is the variance of the noise sequence and $\sigma^2=BN_0$, with B being the bandwidth of the noise and $\hat{\tau}_0$ is the delay estimate returned by the estimator. To simplify the derivation, it is assumed that sampling takes place at the Nyquist rate, i.e. at twice the bandwidth. In other words, $T_s=1/2B$.

Ignoring the noise terms, (A.4) may be rewritten as:

$$\operatorname{var}\left(\hat{\tau}_0\right) \geq \frac{\sigma^2}{\sum_{n=n_0}^{n_0+L_{sum}-1}\left(\frac{\partial m(t)}{\partial t}\right)^2_{t=nT_s}} \tag{A.5}$$

Assuming a high-sampling frequency relative to the bandwidth of the received signal, or equivalently, a low sampling period compared to the duration of the signal, the summation in (A.5) can be replaced by an integral so (A.5) is equivalent to:

$$\operatorname{var}\left(\hat{\tau}_0\right) \geq \sigma^2\left(\frac{1}{T_s}\int_0^{T_s}\left(\frac{\partial m(t)}{\partial t}\right)^2 dt\right)^{-1} = \left(N_0/2\right)\left(\int_0^{T_s}\left(\frac{\partial m(t)}{\partial t}\right)^2 dt\right)^{-1} = \left(\frac{E_m}{N_0/2}\frac{\int_0^{T_s}\left(\frac{\partial m(t)}{\partial t}\right)^2 dt}{E_m}\right)^{-1} \tag{A.6}$$

Using the Fourier transform theorem relating the derivative of a signal to its Fourier transform, the integral in the denominator of (A.6) can be substituted by an integral in the frequency domain:

$$\operatorname{var}\left(\hat{\tau}_0\right) \geq \left(\frac{E_m}{N_0/2}\frac{\int_{-\infty}^{\infty}\left(2\pi f\right)^2 |M\left(f\right)|^2\, df}{E_m}\right)^{-1} \tag{A.7}$$

where $M\left(f\right)$ is the Fourier transform of the mixture $m(t)$. Parseval's theorem can be used to express the energy of the signal as an integral of the squared magnitude of the Fourier transform, and so we have

$$\operatorname{var}\left(\hat{\tau}_0\right) \geq \frac{1}{\frac{E_m}{N_0/2}\frac{\int_{-\infty}^{\infty}\left(2\pi f\right)^2 |M\left(f\right)|^2\, df}{\int_{-\infty}^{\infty}|M\left(f\right)|^2\, df}} \tag{A.8}$$

The term $\frac{E_m}{N_0/2}$ is a type of "SNR" and the term that multiplies it has dimensions of Hz^2. A root-mean square measure of bandwidth of the mixture can be defined as:

$$B_{rms} = \sqrt{\frac{\int\limits_{-\infty}^{\infty} (2\pi f)^2 \, |M(f)|^2 \, df}{\int\limits_{-\infty}^{\infty} |M(f)|^2 \, df}} \tag{A.9}$$

Formally, the Cramer-Rao lower bound for a delay estimator is:

$$\operatorname{var}(\hat{\tau}_0) \geq \frac{1}{SNR \text{ x } B_{rms}^2} (\sec^2) \tag{A.10}$$

References

[1] Y. P. Shen, *et al.*, "Time delay estimation for source localization of vibroarthrographic signals from human knee joints," in *Engineering in Medicine and Biology Society, 1993. Proceedings of the 15th Annual International Conference of the IEEE*, 1993, pp. 381-382.

[2] Y. Mahieux, G. Le Tourneur and A. Saliou, "A microphone array for multimedia workstations," *JOURNAL OF THE AUDIO ENGINEERING SOCIETY,* vol. 44, pp. 365-372, 1996.

[3] S. M. Kuo and J. Chen, "Multiple-Microphone Acoustic Echo Cancellation System with the Partial Adaptive Process," *Digital Signal Processing,* vol. 3, pp. 54-63, 1993.

[4] W.-X. Du, "Earthquake relocation using cross-correlation time delay estimates verified with the bispectrum method," *Bulletin of the Seismological Society of America,* vol. 94, pp. 856-866, 2004.

[5] B. Hodgkinson, D. Shyu, and K. Mohseni, "Acoustic source localization system using A linear arrangement of receivers for small unmanned underwater vehicles," in *Oceans, 2012*, 2012, pp. 1-7.

[6] B. Ferreira, Anibal Matos, and Nuno Cruz, "Optimal positioning of autonomous marine vehicles for underwater acoustic source localization using TOA measurements," in *Underwater Technology Symposium (UT), 2013 IEEE International*, 2013, pp. 1-7.

[7] O. Postolache, M.D. Pereira, and P. Girao, "Intelligent Distributed Virtual System for Underwater Acoustic Source Localization and Sounds Classification," in *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop on*, 2007, pp. 132-135.

[8] A. Pourmohammad and S. M. Ahadi, "TDE-ILD-based 2D half plane real time high accuracy sound source localization using only two microphones and source counting," in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, 2010, pp. V1-566-V1-572.

[9] Y. Huang, J. Benesty, and G.W. Elko , "Microphone Arrays for Video Camera Steering," in *Acoustic Signal Processing for Telecommunication*. vol. 551, S. Gay and J. Benesty, Eds., ed: Springer US, 2000, pp. 239-259.

[10] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 780-785, 1997.

[11] M. S. Brandstein, J. E. Adcock, and H.F Silverman, "A closed-form location estimator for use with room environment microphone arrays," *Speech and Audio Processing, IEEE Transactions on,* vol. 5, pp. 45-50, 1997.

[12] E. Lleida, J. Fernandez, and E. Masgrau, "Robust continuous speech recognition system based on a microphone array," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 1998, pp. 241-244 vol.1.

[13] P. Julian, "A comparative study of sound localization algorithms for energy aware sensor network nodes," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 51, pp. 640-648, 2004.

[14] J. Benesty, J. Chen, and Y. Huang, "Time-delay estimation via linear interpolation and cross correlation," *Speech and Audio Processing, IEEE Transactions on,* vol. 12, pp. 509-519, 2004.

[15] H. Meyr, "Delay-Lock Tracking of Stochastic Signals," *Communications, IEEE Transactions on,* vol. 24, pp. 331-339, 1976.

[16] J. C. Hassab and R. Boucher, "Optimum estimation of time delay by a generalized correlator," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 27, pp. 373-380, 1979.

[17] S. Stein, "Algorithms for ambiguity function processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 29, pp. 588-599, 1981.

[18] C. Jiang , X. Wang, and C. Liu, "Time-Delay Estimation for single-frequency pulse signals based on differential Maximum Likelihood estimation method and DTFT," in *Communications, Circuits and Systems, 2008. ICCCAS 2008. International Conference on*, 2008, pp. 934-936.

[19] H. Saarnisaari, "ML time delay estimation in a multipath channel," in *Spread Spectrum Techniques and Applications Proceedings, 1996., IEEE 4th International Symposium on*, 1996, pp. 1007-1011 vol.3.

[20] A. Satish and R. L. Kashyap, "Multiple target tracking using maximum likelihood principle," *Signal Processing, IEEE Transactions on,* vol. 43, pp. 1677-1695, 1995.

[21] B.P. Bogert, M.J.R. Healy, and J.W. Tukey, "The Quefrequency Analysis of Time Series for Echoes: Cepstrum, Pseudoautocovariance, Cross-Cepstrum and Saphe Cracking," *Proceedings of the Symposium on Time Series Analysis (M.Rosenblatt,Ed.) ,* Chapter 15, pp. 209-243, 1963.

[22] A. V. Oppenheim and R. W. Schafer, "Discrete-time Signal Processing," *Prentice Hall Signal Processing Series ,* Englewood Cliffs, 1989.

[23] P. Stoica and K. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 38, pp. 1132-1143, 1990.

[24] P. Stoica and K. C. Sharman, "Novel eigenanalysis method for direction estimation," *Radar and Signal Processing, IEE Proceedings F,* vol. 137, pp. 19-26, 1990.

[25] S. M. Kay, "Modern Spectral Estimation: Theory and Application," *Ed. Alan V.Oppenheim, Prentice-Hall Signal Processing Series,* Englewood Cliffs, 1988.

[26] C. W. Therrien, "Discrete Random Signals and Statistical Signal Processing," *Ed. Alan V.Oppenheim, Prentice-Hall Signal Processing Series,* Englewood Cliffs, 1992.

[27] T. Ikuma, "Model-Based Identification of POTS Local Loops for DSL Connectivity Prediction," *ECE, Virginia Tech,* Blacksburg, 2001.

[28] Y. Bresler and A. Macovski, "Exact Maximum Likelihood Estimation of Superimposed Exponential Signals in Noise," *Proceedings to IEEE ICASSP-85,* pp. 1824-1827, 1985.

[29] R. Demirli and J. Saniie, "Model-based estimation pursuit for sparse decomposition of ultrasonic echoes," *Signal Processing, IET,* vol. 6, pp. 313-325, 2012.

[30] D.Gabor, “Theory of communications,” *J.Inst.Elec.Eng.,* vol,93, pp.429457, 1946

[31] L. Yinghui and J. E. Michaels, "Numerical implementation of matching pursuit for the analysis of complex ultrasonic signals," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* vol. 55, pp. 173-182, 2008.

[32] H. Jin-Chul, K. H. Sun and Y. Y. Kim, "Waveguide damage detection by the matching pursuit approach employing the dispersion-based chirp functions," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* vol. 53, pp. 592-605, 2006.

[33] G. Cardoso and J. Saniie, "Ultrasonic data compression via parameter estimation," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* vol. 52, pp. 313-325, 2005.

[34] L. Yufeng, R. Demirli, G. Cardoso, J. Saniie, "A successive parameter estimation algorithm for chirplet signal decomposition," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* vol. 53, pp. 2121-2131, 2006.

[35] G. C. Carter, "Coherence and time delay estimation," *Proceedings of the IEEE,* vol. 75, pp. 236-255, 1987.

[36] J. P. Ianniello, "High-resolution multipath time delay estimation for broad-band random signals," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 36, pp. 320-327, 1988.

[37] K. Varma*, et al.*, "Robust TDE-based DOA estimation for compact audio arrays," in *Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2002*, 2002, pp. 214-218.

[38] F. Abrard and Y. Deville, "A time-frequency blind signal separation method applicable to underdetermined mixtures of dependent sources," *Signal Processing, vol.85,* pp. 1389-1403, 2005.

[39] L. Cohen, "Time-Frequency Analysis," *Prentice-Hall PTR,* Englewood Cliffs, NJ, 1995.

[40] M.Fowler, “EECE 522:Estimation Theory,” *Binghamton University,* SUNY, NY http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_08%20Ch_3%20CRLB%20Examples%20in%20Book.pdf

[41] http://en.wikipedia.org/wiki/Cram%C3%A9r%E2%80%93Rao_bound, "Cramer-Rao bound.”