University of Southern Queensland

Faculty of Engineering & Surveying

Directional Microphone Array for Security Applications

A dissertation submitted by

Benjamin Jon Cobb

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Computer Systems)

Bachelor of Information Technology (Applied Computer Science)

Submitted: November, 2006

Abstract

This project investigates the potential for microphone arrays within a security setting, particularly through exploiting their ability to change their directivity pattern, thus, changing its direction of hearing.

In single microphone systems speech and other signals of interest can be severely degraded when recorded in real environments. Security and surveillance systems are no exception to this rule and therefore microphone arrays have the potential to supplement the range of listening devices already available.

This dissertation presents the work covered, which includes:

- The potential need for microphone arrays in security settings.
- The legal implications of using microphone arrays in surveillance.
- The concepts of beamforming and the theory behind three common beamformers, including the Delay and Sum beamformer, the Adaptive Frost beamformer and the Generalized Sidelobe Canceller beamformer.
- Tests conducted on the beamformers to determine their performance with different parameters and different types of data.

It was concluded that both the Frost beamformer and the Generalized Canceller both have the potential to be used in a security setting. The best performance was achieved with the Frost beamformer with an array size of eight microphones and a tapped-delay length of 131.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Benjamin Jon Cobb

Q1222246

Signature

Date

Acknowledgments

Firstly I would like to thank my supervisor, Dr. John Leis, for his support, advice and guidance not only throughout my project, but also throughout my studies here at USQ.

In addition, I would also like to thank Mr David Tilburey for his assistance and time setting up and gathering data for this project.

Finally, I would like to thank my family and friends for without their support, patience and understanding, I would not have achieved all that I have.

Benjamin Jon Cobb

University of Southern Queensland

November 2006

Dedicated to my grandma, Elvira Alice Cobb

27th of September 1924 - 31st August 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Microphone arrays, having been first developed over 20 years ago, have been the subject of much research (Brandstein & Ward 2001), but recent advances in digital signal processing have made them much more sophisticated (Kissell 2004). They have been used in a variety of settings including video conferencing, stage shows, car phones, hearing aids and in the home and office (Nelson & Schreck 1998). This project, however, focuses on the use of microphone arrays for security applications, in particular, surveillance.

1.1 Background Information

In single microphone systems, speech and other signals of interest can be severely degraded when recorded in real environments (Abad & Hernando 2004). When a microphone is placed at a distance from the desired signal source, other signals such as noise and reverberation can interfere with the recording of the desired signal. There are many types of signals that can be considered "noise". In the context of this project, noise is considered to be any non-desired signal of any nature that is picked up in addition to the desired signal. For example, in the context of recording human speakers, other speakers, door slams or air-conditioners can all be considered noise. Reverberation, on the other hand, is a product of the physical dimensions and material properties of the environment in which a signal is recorded. This is often the case in a controlled situation. In environments likely to be encountered in a security and surveillance role there may be many other forms of noise present including television, radio, wind and motor vehicles.

Microphone arrays are designed to take advantage of the fact that sound signals, both desired and undesired, generally originate from different points in space (Brandstein & Ward 2001). Using different methods of spatial filtering, microphone arrays have the ability to discriminate between the different signals and, according to Brandstein & Ward (2001, p. V), “allow users to roam unfettered in diverse environments while still providing a high quality speech signal and robustness against background noise, interfering sources, and reverberation effects”. Spatial filtering is achieved by using a microphone array, which samples a propagating wave spatially, in conjunction with a processor that is usually termed a “beamformer” (Van Veen & Buckley 1988). This is unlike conventional single microphone systems (Nelson & Schreck 1998), which require a microphone to be placed very close to the source to achieve the equivalent noise rejection ability (Ward, Williamson & Kennedy 1998).

Microphone arrays are usually constructed out of many omni-directional microphones and, combined with processing software, have several distinct advantages over typical directional and omni-directional microphones. They have the ability to change their directional sensitivity pattern, allowing the microphone array to steer or change the direction of its hearing, thus cancelling out unwanted noise. This ability is enhanced by the fact that it can do this in software and the physical array itself does not need to move. Finally, a microphone array has the ability to localise and track a sound source.

Due to these advantages, directional microphone arrays have the potential to be used in security contexts. The Macquarie Dictionary (2004) defines security (for the purpose of this research) as:

1. freedom from danger, risk, etc; safety.
2. freedom from worry or doubt; confidence.
3. something that secures or makes safe; a protection; a defence.

Based on this definition it can be concluded that a security application is any application where the device contributes to protecting a person's, organisation's or country's health and wellbeing, their assets or their intellectual property. This presents a wide variety of possible security applications for microphone arrays based around the array's abilities in signal improvement and source localisation.

"Directional Microphone Arrays for Security Applications", as a broad topic, includes both beamforming and source localisation. As there have been significant developments regarding microphone arrays for source localisation (a few of which will be presented later), this project will concentrate on the aspects of signal enhancement via the use of beamforming and its potential security applications, in particular microphone arrays for the purpose of surveillance.

1.2 The Need for Effective Surveillance

The world is a dynamic and dangerous place. The threat of terrorism is forever present in our everyday lives. Living in Australia does not mean that we are immune. The Government white paper entitled "Transnational Terrorism - The Threat to Australia" (Department of Foreign Affairs and Trade 2004, p. 66), released 15th of July 2004, states that:

> Australia is a terrorist target, both as a Western nation and in its own right. Intelligence confirms that we were a target before the September 11 attacks, and we are still a target. Our interests both at home and abroad are in the terrorists' sights.

This has been made very clear by statements made by Usama bin Laden, his deputy Ayman al-Zawahiri, and other al-Qai'da leaders, Abu Mus'ab al-Zarqawi in Iraq and Abu Bakar Baasyir in Indonesia, who all specifically mentioned Australia (Australian Security Intelligence Organisation 2005). The Australian Security Intelligence Organisation (2005, p. 15) states that "there has been at least one aborted, disrupted or actual terrorist attack against Australian interests every year since 2000." These include the 2002 Bali bombings, in which 202 people were killed including 88 Australians, the 2004 Jakarta embassy bombing, in which 11 Indonesians were killed and many more wounded (*The Age* 9 Sept. 2004), and the attacks on the Australian Security Detachment in Iraq in January 2005, in which several Australians were injured (*The Age* 27 Jan. 2005).

Less prominently, but still importantly, organised crime has a large impact on Australia. The report entitled "Policing Organised Crime", produced by the Australian Institute of Criminology (Irwin 2001, p. 2), defines organised crime in Australia as "a myriad of complex activity including illicit drug importation, manufacture and distribution, large-scale organised fraud, revenue evasion, money laundering, bribery, extortion and violence." It affects institutions such as the Stock Exchange or superannuation funds and seeks to corrupt public officials. The costs carried by such activities are enormous, both in terms of social and economic costs - where the costs may be in monetary terms, lost productivity, public health and welfare - and other social problems (Irwin 2001).

Irwin (2001) further states that the monetary cost of illicit drugs on the Australian community is estimated to be at least $1.7 billion annually and money laundering is estimated to be between $3-9 billion annually. The total cost of crime in Australia is estimated to be equivalent to about 4 percent of the Gross Domestic Product or $1000 per capita, per annum.

These are just some of the problems that need to be handled by authorities and, to tackle these problems, effective surveillance is required. Australia's Attorney-General, Mr Philip Ruddock, said in an address to parliament in March of 2004 (p. 1):

> Our police forces rely on a variety of tools to investigate, catch and prosecute criminal groups which are becoming ever more organised and sophisticated. One increasingly important tool is the use of surveillance devices which can range from a pair of binoculars, a tiny microphone or camera hidden in a suspect's vehicle, to a piece of software to capture the input of information to a computer.

In the report “Surveillance Devices Act 2004, Report for the year ending 30 June 2005”, (Commonwealth of Australia 2005, p. 6), the Australian Federal Police and the Australian Crime Commission reported that “the use of surveillance devices is an extremely valuable investigative tool.”.

These devices, however, are susceptible to noise interference, greatly reducing their performance and the value of the data recorded, as mentioned earlier.

1.3 The Need for Effective Security

Security is also becoming a major concern in today’s technological society. According to Yun (2002, p. 1), “Even in the current Information Technology (IT) age, identity authentication is very crucial.” Technology today allows us the means to perform many transactions where one-on-one personal contact is no longer necessary, making the act of confirming the identity of an individual difficult (Yun 2002).

Most current means of identification require using items such as cards, keys, Personal Identification Numbers (PINs) and passwords (Yun 2002). These, however, can be easily obtained. Credit card numbers, which can be used to authenticate users, and passwords are often used over the Internet and can be quite easily obtained by hackers using various means. PINs can often be obtained either because the owners recorded them somewhere so as not to forget them or through observation of the user using the PIN. The main issue with regards to identification, according to Yun (2002), is to identify an individual without the need of a complex system with which the user can misuse. One solution that is gaining popularity is Biometrics - one method being voice authentication. This method, however, requires a microphone which is susceptible to noise interference (Yun 2002), therefore a more effective solution is required, such as microphone arrays.

1.4 What is a Microphone Array

A microphone array is an array of microphones arranged in some pattern and attached to some form of Digital Signal Processor (DSP) to derive meaningful information. The three most commonly used patterns are (Naidu 2001):

Uniform Linear Array (ULA) The microphones are arranged in a straight line, shown in Figure 1.1.

Uniform Circular Array (UCA) The microphones are arranged in a circular array, shown in Figure 1.2.

Uniform Planar Array (UPA) The microphones are arranged in a matrix, shown in Figure 1.3.

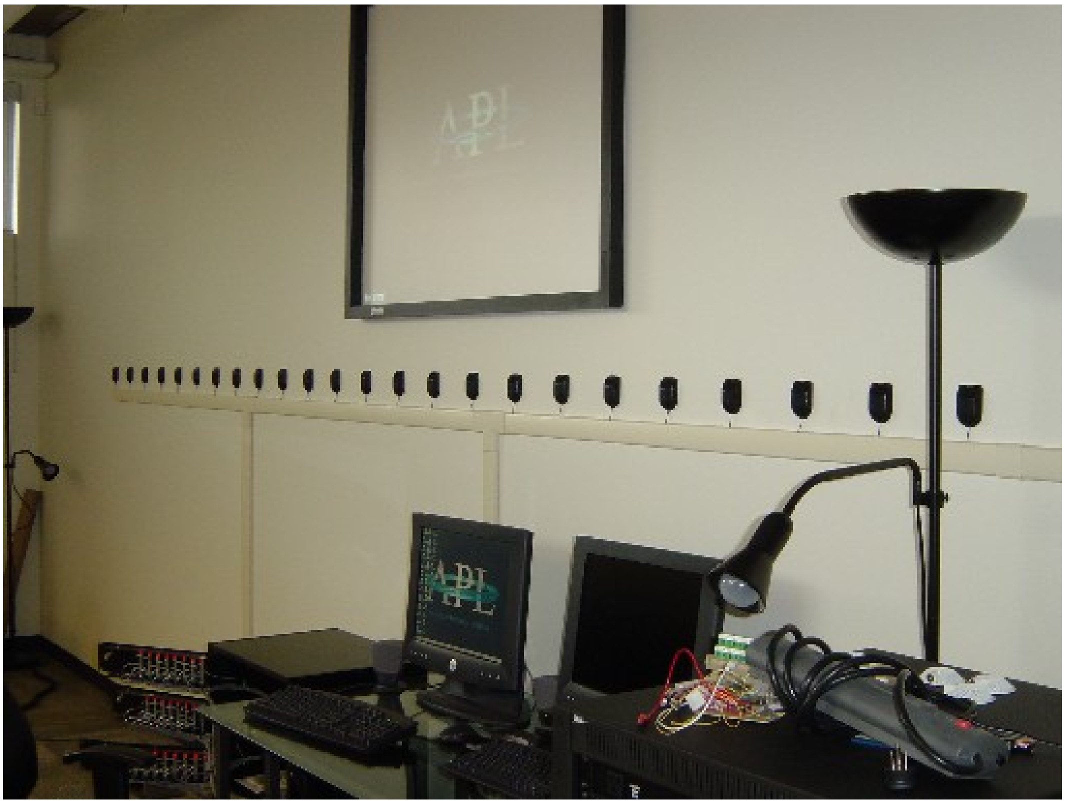

Figure 1.1: Microphones arranged in a ULA geometry (APL n.d.).

Each microphone in the array records a time-delayed version of the same basic waveform.

Figure 1.2: Microphones arranged in a UCA geometry (Hiyane & Iio 2000).

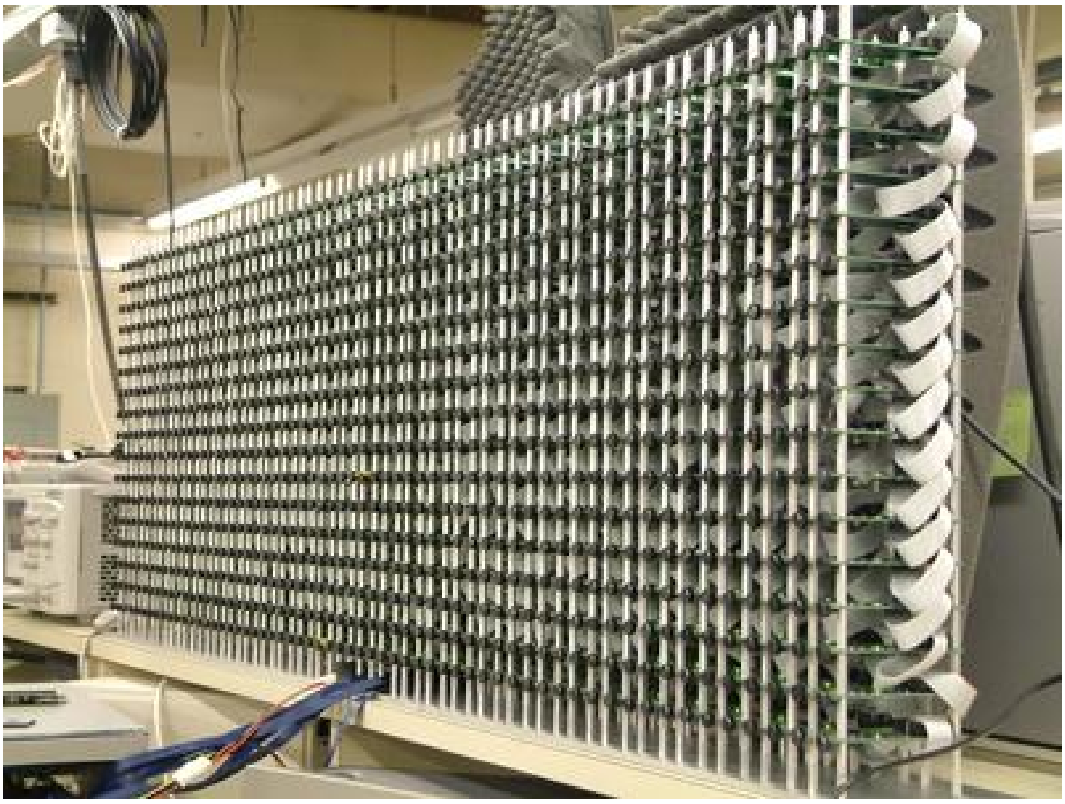

Figure 1.3: Microphones arranged in a UPA geometry (Weinstein et al. 2004).

1.5 Objectives of the Project

This dissertation examines the possible use of microphone arrays for information gathering, which is achieved through their ability to change their directional sensitivity pattern. Although there are many places where this capability can be used, it has the potential to make a significant impact in the area of surveillance. This application has many legal implications, therefore the legislation regarding listening devices is also covered. Three beamforming algorithms are then introduced and implemented to demonstrate the potential of microphone arrays.

The objectives of this project are:

1. Establish the potential need for directional microphone arrays in a security context and suggest examples of possible roles for which they can be used.
2. Investigate the legal implications of using microphone arrays in security applications.
3. Investigate the principles involved in creating a directional microphone array specifically designed for the purpose of beamforming.
4. Develop a software model of a directional microphone array in MATLAB^{TM} for the purposes of beamforming.
5. Evaluate the software model by varying the parameters of its operation and of the test sound files.
6. Set up a microphone array using off-the-shelf studio microphones and record test files which can be used to evaluate the microphone array models.

1.6 Overview of the Dissertation

This dissertation is organised as follows:

Chapter 2 Surveillance and Security This chapter discusses the importance of microphones in security and surveillance applications and the roles that they can perform. It also covers existing microphone array security and surveillance systems.

Chapter 3 Government Legislation and Ethics This chapter covers Government legislation that is relevant to microphone arrays when they are used in a security setting. Ethical issues relevant to engineers are also covered.

Chapter 4 Introduction to Beamforming This chapter introduces beamforming using the Delay and Sum beamformer as well as and several other important concepts such as Near-field and Far-field signals, spatial filtering, beamformer spatial response and narrowband and broadband beamforming.

Chapter 5 Broadband Beamforming This chapter presents several broadband beamformers including the Adaptive Frost beamformer and the Generalized Sidelobe Canceller.

Chapter 6 Beamformer Implementation This chapter presents the implementation of the microphone array model and three beamformers that were implemented in MATLAB^{TM}. In addition it describes the process used to set up a experimental microphone array for the purpose of recording data.

Chapter 7 Results and Discussion This chapter presents results obtained from the different beamformers implemented. These results were obtained by varying several microphone array and beamformer parameters such as number of microphones, length of tapped-delay line and distance between microphones.

Chapter 8 Conclusions and Further Work This chapter concludes the dissertation and presents recommendations for further work.

Chapter 2

Surveillance and Security

As stated in the first chapter, this project focuses on beamforming for the purposes of surveillance and more broader security applications. This is because the principles of applying microphone arrays to security applies to applications in other areas.

This chapter will discuss the importance of microphones in surveillance and then in wider security applications.

2.1 Microphones in Surveillance

Bugging, as defined by the Macquarie Dictionary (2004, p. 59) is “to install a bug in (a room etc),” where a bug is defined as “a microphone hidden in a room to tap conversation” (p. 59). However, by its very nature, it is difficult to tell how often surveillance is practiced by bugging, both legally and illegally, except when such devices are uncovered or Governments release statistics documenting their use.

One such case of illegal bugging occurred in 2005, when a listening device was placed outside of the Sydney mansion of Nicole Kidman, which the police believed was aimed at recording conversations between Ms Kidman and her bodyguards (CBS News 2005). Other reported incidents include the bugging of French, German and other nation’s offices at a European Union summit held in Brussels, 2003 (CBS News 2003), as well as the incident where the United States State Department was bugged by the Russian Spy Service (CBS News 2000).

Statistics released by the Commonwealth of Australia in the report entitled “Surveillance Devices Act 2004: Report for the year ending 30 June 2005” (Commonwealth of Australia 2005) give the number of surveillance devices issued to the Australian Federal Police and the Australian Crime Commission under the Surveillance Devices Act 2004, which came into effect on the 15th of December 2004. This report stated that, during this time, 86 warrants were issued for the use of listening devices and 102 warrants were issued for composite devices which provide more than one function, for example, a listening and tracking device. This is greater than the number of optical devices used, 19, the number of data, two, and the number of tracking devices, 47.

Bugs do have limitations, however. Advertised on a company website that sells surveillance and anti-surveillance devices (Endoacustica n.d.), are miniaturised bugs which, in their product description, state that the performance of the bug depends on environmental noise conditions. Devices can also be purchased specifically designed to drown out any speech that a planted bug is meant to hear. An example of this is given by Defense Devices (n.d.), another company that sells surveillance and anti-surveillance devices.

Microphone arrays, with their ability to filter noise based on spatial locations, do not suffer from the above problems and therefore are ideal candidates to be used in this role.

2.2 Microphones in Security

Yun (2002, p. 2) tells us that “biometrics is the automated approach to authenticate the identity of a person using the individual’s unique physiological or behavioural characteristics such as fingerprint, face, voice, signature etc.” These systems have advantages over regular security systems in that one cannot forget these details and, because they are unique to a person they are, therefore more difficult for others to steal.

The process of setting up and using a biometric security system is shown in Figure 2.1. A sensor, or in the case of voice recognition/authentication, a microphone is used to capture the biometric data. A computer is then used to extract unique features which can be used to identify an individual and this is stored as a template.

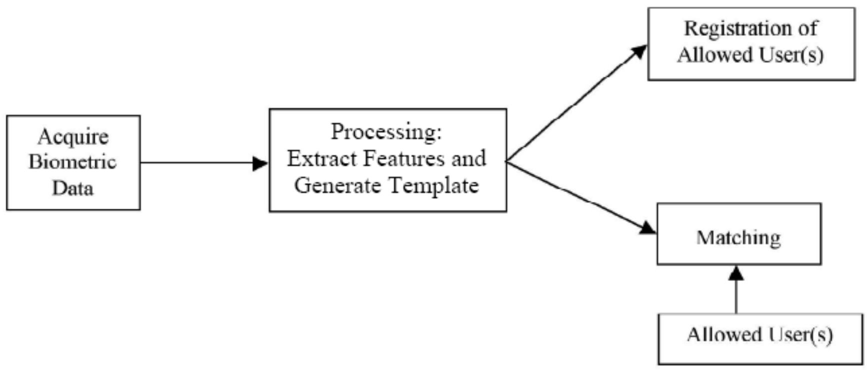

Figure 2.1: The process of using biometric technology (Yun 2002, p. 84).

There are two types of voice authentication - text dependent and text independent (Yun 2002). Text dependent requires the speaker to speak a passage of known text, while text independent uses unknown text. According to Yun (2002), the cost of voice authentication can be quite low and relatively easy to use. It does, however, suffer from several problems, in particular background noise in the environments in which it is used (Yun 2002).

A table comparing the different biometric systems is shown in Figure 2.2. "Universality" refers to how common the biometric is found with each person, "Uniqueness" is how well the biometric separates each person, "Permanence" is how consistent the biometric is over the life of the person, "Collectability" is how easy the biometric can be gathered, "Performance" indicates the achievable accuracy, "Acceptability" is how well the technology is accepted by society and "Circumvention" is how easy the technology is to fool.

McCowan, Pelecanos & Sridharan (n.d.) make the link between voice authentication and microphone arrays, stating,

Accurate speaker recognition can be an integral part of many security appli-

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

Figure 2.2: Comparison of various biometric technologies (Yun 2002, p. 91).

> cations, controlling access to information, property and finances... In such applications, a microphone array capable of enhancing the desired speech from a known location offers a means of meeting the requirements for hands-free operation and robustness to noise conditions.

These applications are essentially a section of biometrics, i.e. speaker identification.

By using microphone arrays, the amount of noise entering into the system could be greatly reduced, allowing the system to perform at a higher accuracy or in high noise conditions without the need for the microphone to be very close to the speaker's mouth. Such applications could include voice identification for ATMs or access to buildings, vehicles and computers.

2.3 Currently Deployed Systems

Microphone arrays are a versatile platform from which many different forms of surveillance and security tasks can be carried out. As such, they have already found uses in a variety of settings, as demonstrated by the following cases.

2.3.1 SENTRI

Chicago police are implementing a new system capable of recognising the sound of a gunshot, determining the location of the source and then turning a mounted camera in that direction (Reichgott 2005). It then makes a 911 call to authorities who can then take control of the camera to monitor the scene and dispatch officers.

The system comprises of two elements. The first is a Smart Sensor Enabled Neural Threat Recognition and Identification system (or SENTRI system) and the second is a movable camera. The SENTRI system uses four microphones, see Figure 2.3, and using sophisticated signal processing, it first determines whether a sound that is received is a gunshot and not a car backfiring or a siren and then, using localisation algorithms, determines the direction of the source.

Figure 2.3: The SENTRI or Smart Sensor Enabled Neural Threat Recognition system attached to a phone pole in Chicago (Reichgott 2005, p. 1).

According to Reichgott (2005), the system appears to be working, with the Chicago officials crediting the SENTRI system with helping to lower violent crime rates. At the time of publication, Chicago officials had installed 30 units equipped with video cameras on phone poles in high crime areas with plans to install many more. In addition, the sheriff's department in Los Angeles County, and police in San Francisco and Philadelphia, plan to run their own pilot and test programs with the new technology.

2.3.2 Boomerang

A second similar system was developed by BBN Technologies for use by U.S. troops in Iraq (BBN Technologies 2006). The system, called "Boomerang", was designed to protect vehicles and troops from sniper and small arms fire. The system uses a microphone array mounted onto a vehicle, see Figure 2.4, and is capable of detecting arms fire from both the supersonic shock waves of the projectile and sound waves from the muzzle blast and then localise the fire so that the soldiers can either return fire or retreat to a safer location.

Figure 2.4: The boomerang system mounted onto a High Mobility Multipurpose Wheeled Vehicle (HMMWV) (BBN Technologies 2006, p. 1).

2.3.3 Ears in the sky

Another security application of microphone arrays is that of "aerial audio surveillance". According to Cermin (2004) "The military, security and law enforcement communities recently have become interested in collecting acoustic data from airborne platforms, particularly from unmanned aerial vehicles (UAVs)." Cermin (2004) states that airborne platforms have a number of advantages over ground-based sensors:

- The system can be deployed quickly and easily to collect data over large areas.
- The system allows intelligence data to be collected from a safe distance.

The system that Cermin (2004) describes is an extension of existing two-to-four microphone array systems that are currently used. The new system incorporates many more microphones to help in reducing wind noise and increase the accuracy of the system and extends the array in two dimensions so that a direction can be found to the source. Typical applications of this type of system include locating gun-fire, vehicles and other targets based on their acoustic signatures.

2.3.4 Audio, Video and Robotic Surveillance System

Menegatti et al. (n.d.) presented a paper on a multi-element surveillance system comprising of audio, video and mobile agents to reveal and track the presence of an intruder. The system is designed for use in multiple rooms where the room contents are dynamic, for example, the storage warehouse of a shipping company. The system is designed to adapt to the changing environment, particularly in a situation where one or more of its sensors are blocked due to piled-up objects.

The system first uses a static omni-directional video camera to detect a moving object. It then communicates the position of the moving object to several microphone arrays positioned around the room (Menegatti et al. n.d.). The arrays use source localisation to locate the intruder and then begin tracking him or her. The location estimates from the vision system and the microphone arrays are then combined to provide a more

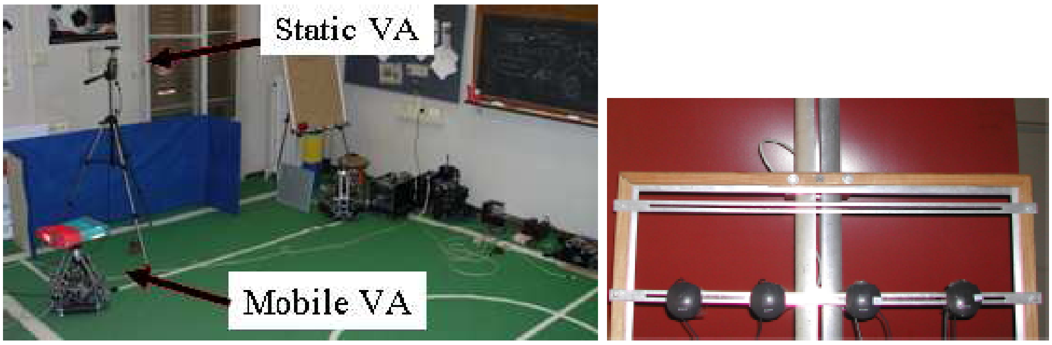

Figure 2.5: The different sensors used. The left image shows both the static and mobile vision agent and the image on the right shows one of the microphone arrays (Menegatti et al. n.d., p. 2).

accurate estimate which can then be sent to the mobile vision agent which approaches the intruder to gain a close-up image that is sent to the monitoring station to determine whether the intruder is really a threat. Figure 2.5 shows the three different types of elements used in the system.

2.4 Chapter Summary

Surveillance microphones, or bugs, are one popular way of gathering intelligence through the use of surveillance devices. They do, however, suffer from the problem of environmental noise as well as devices specifically designed to hinder their performance. Microphone arrays offer a solution to this problem, particularly in high noise environments.

Biometrics, in particular voice authentication, offer a user friendly method of improving the security of accessing various resources but, like bugs, they also suffer the problem of not coping with environmental noise.

Source localisation is a capability provided by microphone arrays that can also be used in a security environment and several systems have been developed or are currently being researched. Those presented were:

- The Smart Sensor Enabled Neural Threat Recognition and Identification (SENTRI) system developed to help lower crime rates in high crime areas by detecting the sound and location of gun shots and reporting them to authorities.
- The Boomerang system to detect the direction and range of gun fire directed at military personnel.
- The airborne audio surveillance system designed to locate gun-fire, vehicles and other targets based on their acoustic signatures.
- The multi-element surveillance system in which several microphone arrays play an integral part in locating and tracking an intruder in a dynamic environment.

Chapter 3

Government Legislation and Ethics

One area of particular interest, in relation to the use of microphone arrays, is that of surveillance. However, there is a considerable amount of State, Territory and Federal Government legislation regarding the use of listening devices. This legislation limits their uses, how they can be used and what needs to be in place for them to be used. In addition, there are ethical issues, regarding the recording of individuals without their consent, that need to be considered.

In the context of examining the following legislation, and for the purposes of this project, a microphone array will be considered a form of listening device.

3.1 Government Legislation

Each State and Territory has at least one piece of legislation relating to the use of listening and surveillance devices. Some of the significant State and Territory legislation can be seen in Table 3.1.

There also exists Commonwealth legislation which governs the use of listening devices for government bodies that do not fall within a particular state and to establish pro-

Act	**State**
Listening Devices Act 1984	NSW
Surveillance Devices Act 1999	VIC
Invasion of Privacy Act 1971	QLD
Police Powers and Responsibilities Act 1997	QLD
Surveillance Devices Act 1998	WA
Listening and Surveillance Devices Act 1972	SA
Listening Devices Act 1991	TAS
Surveillance Devices Act 2000	NT
Listening Devices Act 1992	ACT

Table 3.1: State and Territory legislation relating to the use of listening devices. Here only principle Acts are shown - amendments are not included.

cedures for obtaining warrants for the use of listening devices for those organisations. Significant Commonwealth legislation can been seen in Table 3.2.

Act
Surveillance Devices Act 2004
Australian Security Intelligence Organisation Act 1979

Table 3.2: Federal legislation relating to the use of listening devices. Here only principle Acts are shown - amendments are not included.

All Government legislation that has been encountered follows the same general theme with regards to listening devices, although the wording can vary considerably and time frames for events can change between the different Acts. Therefore, the Queensland Government legislation will be presented here as it provides a good background in what is expected and it is most relevant to the context of this project.

3.2 Restriction on Listening Devices

Queensland has two Government Acts that relate to the use of listening devices. They are the "Invasion of Privacy Act 1971" and the "Police Powers and Responsibilities Act 1997". The first act applies to both private individuals and Government agencies while the second act applies to the Queensland Police Force.

Before looking at the legislation regarding listening devices, it is first important to define what, according to the *Invasion of Privacy Act 1971*, the term "private conversation" means:

> Private conversation means any words spoken by one person to another person in circumstances that indicate that those persons desire the words to be heard or listened to only themselves or that indicate that either of those persons desires the words to be heard or listened to only by themselves and some other person, but does not include words spoken by one person to another person in circumstances in which either of those persons ought reasonably to expect the words may be overheard, recorded, monitored or listened to by some other person, not being a person who has consent, express or implied, of either of those persons to do so.

Using the Invasion of Privacy Act definition of private conversation, a person is guilty of an offence under section 43 of this Act if he or she uses a listening device to record, monitor or overhear a private conversation. There are, however, exceptions to this statement. These are:

- When the person using the listening device is a part of the conversation.
- When the overhearing is unintentional through the use of a telephone.
- When an officer in the employment of the Commonwealth in relation to a customs authorised warrant is using the device.
- When a person employed by the Commonwealth is using the device in the interests of security under an Act passed by Parliament of the Commonwealth relating to

security of the Commonwealth.

- When the device is used by a police officer or another person under the provision of an Act authorising its use.

In the instance that the person using the listening device is a part of the conversation, that person is guilty of an offence under section 45 of the Invasion of Privacy Act 1971 if that person publishes, to any person, the content of the recorded conversation. This, however, does not apply in the following cases:

- When it is communicated to a member of the private conversation or that person has the express of implied consent of all other parties.
- When it is communicated during the course of legal proceedings.
- When the communication is not more than reasonably necessary and in the public interest, part of the duty of the person making the communication, or for the protection of the lawful interests of that person.
- When the recording is communicated to a person who has, on reasonable grounds, an interest in the private conversation.
- When the person using the listening device is doing so under a warrant or under the provision of an Act authorising its use.

3.3 Warrants

For authorities to legally use a listening device, a surveillance warrant is required. The procedures for obtaining such a warrant, and the powers associated with such a warrant, are set out in the *Police Powers and Responsibilities Act 1997*. Under this act, a surveillance device is a listening device, a visual surveillance device, a tracking device or any combination of the three mentioned. These devices then belong to one of two classes depending on how they are installed. A class 'A' device is one that is installed in a private place or on a suspect's person, without their consent, or in a public place. This does not include visual surveillance devices installed in a public place. A class 'B' device is a tracking device installed in a vehicle and is, therefore, not relevant to this project.

Application for a class A device must be made to a Supreme Court judge by a police officer of minimum rank of inspector and the offence must be a serious indictable offence. A serious indictable offence is defined by the Police Powers and Responsibilities Act (p. 106) as:

- serious risk to, or actual loss of, a persons life;
- serious risk of, or actual, serious injury to a person;
- serious damage to property in circumstances endangering the safety of any person;
- serious fraud;
- serious loss of revenue to the State;
- official corruption;
- serious theft;
- money laundering;
- conduct related to prostitution or SP bookmaking;
- child abuse, including child pornography;
- a drug offence punishable by at least 20 years imprisonment.

According to the Police Powers and Responsibilities Act, a judge, when considering issuing a warrant, must consider:

- The nature and the seriousness of the suspected offence;
- The likely extent of interference with the privacy of the suspect or other occupants of the place;
- The extent to which issuing the warrant would help prevent, detect or provide evidence of the offence;

- The benefits derived from the issue of any previous surveillance warrants in relation to the suspect;
- The extent to which conventional means of investigation have been used;
- How much the use of conventional means of investigation would be likely to help in the investigation of the offence;
- How much the use of conventional means of investigation would prejudice the investigation of the offence because of delay or for another reason.

3.4 Absence of Warrant

In the event that there is a risk of serious injury to a person and a surveillance device may lower the risk, and there is an insufficient amount of time to obtain a warrant, the Police Powers and Responsibilities Act provides a police officer of the rank of inspector the power to authorise the use of a device, provided he or she applies for approval from a Supreme Court judge within two working days.

3.5 Powers given by Warrants

A surveillance device warrant given under the Police Powers and Responsibilities Act gives certain powers to police officers. These are:

- The power to enter a specified place covertly or through subterfuge to install, maintain, replace or remove a listening device;
- The power to intercept and record conversations even though it may constitute an offence under the Invasion of Privacy Act;
- The power to take electricity for using a surveillance device;
- The power to use reasonable force to install, maintain, replace or remove a surveillance device;

- The power to use one or more surveillance devices, whether the same or a different kind, in the same place;
- The power to pass through, over, under or along a place to the place where the surveillance device is to be used.

3.6 Implications of Legislation

Due to the above-mentioned legislation, there are several implications of the use of microphones for security applications. As stated in the legislation, to use a listening device to record a private conversation the person must either be a part of the conversation or hold a warrant authorising its use. Because of the fact that a microphone array is classed as a class A device, it can only be used for gathering information on serious indictable offences listed above.

A warrant authorising the use of a listening device also has a limited life. This means that a microphone array to be used as a surveillance device needs to be portable so that it can be installed covertly and subsequently removed upon the expiry of the warrant. This would also mean that any microphone array to be designed for surveillance applications would need to be concealed, both itself and its, supporting components.

3.7 Legal Considerations of this Project

The Invasion of Privacy Act also prohibits the advertisement, through any form of media, or public exhibition, of listening devices. Therefore, this project and dissertation is written to fulfil the academic requirements for the completion of my degree. It seeks to establish a proof of concept for the use of microphone arrays for the purpose of security applications. It does not seek to endorse or promote the construction or use of microphone arrays or any other form of listening device for the purpose of recording words, sounds or signals that an individual is not privy to.

3.8 Ethical Considerations

There is a considerable degree of ethical responsibility associated with this project and care will be taken to ensure that what are generally accepted as current ethical standards are not breached.

Although the purpose of this project is to investigate microphone arrays for the purpose of security applications, such as detecting unlawful and often immoral behaviour of individuals, the privacy of those individuals, as well as other individuals, has the potential to be compromised.

As a student member of the Institution of Engineers Australia, I am committed and obliged to apply and uphold the cardinal principles of the Code of Ethics. In particular, the tenets of the code which apply to this project are (The Institution of Engineers Australia 2000):

1 members shall at all times place their responsibility for the welfare, health and safety of the community before their responsibility to sectional or private interests, or to other members;

2 members shall act in order to merit the trust of the community and membership in the honour, integrity and dignity of the members and the profession;

4 members shall act with fairness, honesty and in good faith towards all in the community, including clients, employers and colleagues;

5 members shall apply their skill and knowledge in the interests of their employer or client for whom they shall act as faithful agents or advisors, without compromising the welfare, health and safety of the community.

6 members shall take all reasonable steps to inform themselves, their clients and employers and the community of the social and environmental consequences of the actions and projects in which they are involved.

In applying these principles, I will not seek to invade other people's privacy by recording their conversations to which I am a party or otherwise, without all parties' consent.

3.9 Chapter Summary

There is a considerable amount of Government legislation relating to and regulating the use of listening devices and this has a considerable impact on how a microphone array might be used as a listening device. Two current pieces of legislation relating to the use of listening devices in Queensland are the Invasion of Privacy Act 1971 and the Police Powers and Responsibilities Act 1997.

Current legislation requires that a person does not use a listening device to record, overhear or monitor a private conversation unless that person is a party to that conversation or holds a warrant authorising such an action. Obtaining a warrant requires that the offence be a serious indictable offence. In the event that there is not enough time to obtain a warrant for the use of a listening device, a police inspector may authorise the use of the device provided approval is sought within two working days.

Due to this legislation, restrictions on using a microphone array as a surveillance device, require it to be portable so that it can be installed and then removed when the warrant authorising its use expires. It also requires that the device can be concealable to make it an effective covert surveillance device.

Chapter 4

Introduction to Beamforming

Brandstein & Ward (2001, p. 3) describe beamforming as " one of the simplest and most robust means of spatial filtering, i.e., discriminating between signals based on the physical locations of the signal sources". The term "beamformer" is therefore given to any processor that performs spatial filtering when the spatial sampling is discrete (Van Veen & Buckley 1988). In a typical situation where a speaker's voice is to be recorded, their speech signal can be corrupted by other interfering signals such as noise and reverberation. If this interference occupies the same temporal frequency band as the desired signals, then temporal filtering cannot be used to remove the noise (Van Veen & Buckley 1988). These interfering signals, however, usually originate from points in space other than the source of the desired signal. With a microphone array, this spatial difference can be used to generate a high-quality signal without the need for a microphone to be positioned close to the source (Brandstein & Ward 2001).

According to Van Veen & Buckley (1988), "The term beamforming derives from the fact that early spatial filters were designed to form pencil beams in order to receive a signal radiating from a specific location and attenuate signals from other locations." Although the term "beamforming" implies the creation of a beam by the radiation of energy, it can be used to describe the process of beamforming for the reception of energy (Van Veen & Buckley 1988). Microphone arrays for the enhancement of speech are not the only possible uses of beamforming, as can been seen by Table 4.1 which lists several applications of beamformers.

Application	**Description**
RADAR	phased-array RADAR; air traffic control; synthetic aperture RADAR
SONAR	source localisation and classification
Communications	directional transmission and reception; sector broadcast in satellite communications
Imaging	ultrasonic; optical; tomographic
Geophysical Exploration	earth crust mapping; oil exploration
Astrophysical Exploration	high resolution imaging of the universe
Biomedical	foetal heart monitoring; tissue hyperthermia; hearing aids

Table 4.1: Arrays and beamformers provide an effective and versatile means of spatial filtering. This table lists a number of applications of spatial filtering and gives examples of arrays and beamformers (Van Veen & Buckley 1988, p. 5).

Beamformers can generally be placed into two categories - narrowband beamformers and broadband beamformers (Ward et al. 1998). Narrowband beamformers generally work over a relatively small number of frequencies, unlike broadband beamformers which can operate over a large frequency range. Signals that fit this description, such as speech, are known as broadband signals (Ward et al. 1998). This chapter introduces several concepts, including that of narrowband beamforming using the simple Delay and Sum beamformer, Near- and Far-field signals, spatial sampling and spatial response. The next chapter then introduces several broadband beamformers.

4.1 Delay and Sum Beamformer

There are two different notations that are used to describe beamformers. The simple one will be used first to introduce some of the basic concepts. The more complex notation will then be introduced.

4.1.1 A Simple Notation

A simple Delay and Sum beamformer can be created using a uniform linear array, which is an array of microphones placed with equal-distance spacing. Assuming that the sound waves are arriving parallel, and assuming that a sound source produces a stationary stochastic signal, $f(t)$, the output of each microphone in the array will be a time-delayed version of the original signal with respect to the first or "reference" microphone (Naidu 2001). Therefore the output of the first microphone will be:

$$f_1(t) = f(t) \tag{4.1}$$

and the second:

$$f_2(t) = f(t - \Delta t) \tag{4.2}$$

Hence, the output of the m^{th} microphone will be:

$$f_m = f(t - (m-1)\Delta t) \tag{4.3}$$

where
Δt is the time delay between each microphone.

This time delay can essentially be seen as a phase change of the signal between the microphones. Figure 4.1 depicts the arrival of the signal at a microphone array and Figure 4.2 shows what will be recorded by each microphone. (Note - Naidu (2001) references microphones from 0 onwards, however in this project the microphones will be referenced from 1.)

The time delay associated with each microphone, with respect to the first, is due to the difference in distance to the source of the sound from each successive microphone in the array (Finnigan et al. 2004). With a known distance, d, between the microphones the extra distance can be calculated using trigonometry rules.

When a signal arriving at a specific angle, θ, from perpendicular to the array, like what is shown in Figure 4.1, the extra distance travelled, with respect to the first microphone, can be calculated by equation 4.4.

$$d_m = (1-m)d\sin\theta \tag{4.4}$$

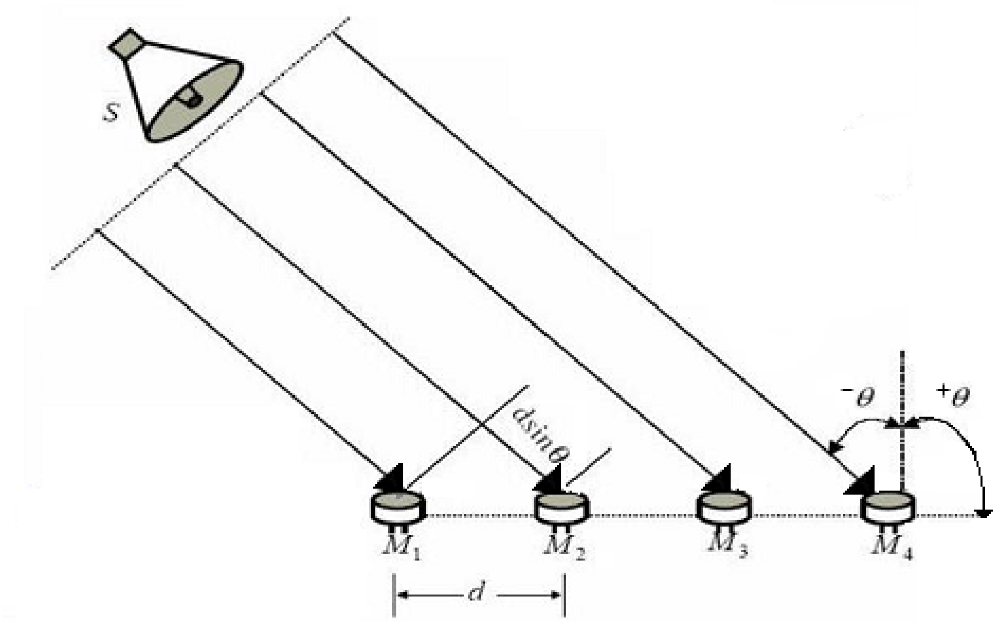

Figure 4.1: Microphones arranged in a uniform linear array geometry (Finnigan et al. 2004, p. 1).

where

d_m is the extra distance travelled by each wave with respect to the first microphone.

m is the microphone.

d is the distance between the microphones.

Using the speed of sound of 344m/s at 20^oC (Microsoft Encarta Encyclopedia Standard 2004), it is possible to calculate the time delay between the first microphone and the mth microphone using Equation 4.5.

$$\tau_m = \frac{d_m}{c} \tag{4.5}$$

where

d_m is the extra distance travelled by each wave with respect to the first microphone.

m is the microphone.

c is the speed of sound.

To steer the microphone to listen in a particular direction, θ, also known as the "look direction", delays are applied to each of the microphone inputs and then added together.

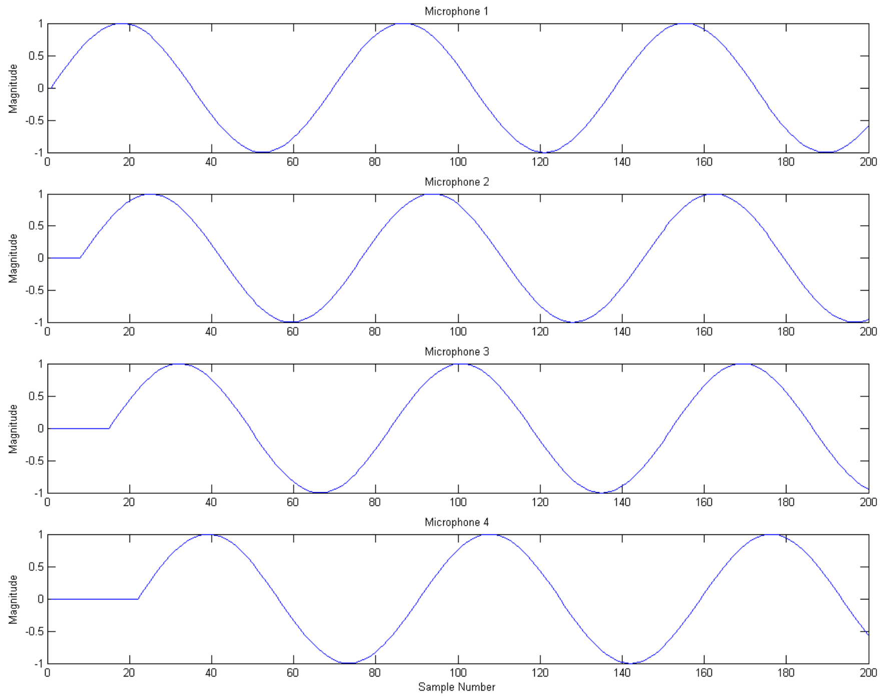

Figure 4.2: What is recorded by a microphone array, when the signal is arriving from an angle of -60^o. Notice the increase in delay or phase change of the signal between the microphones

The delays are calculated using equation 4.5. The signals from the desired direction should then reinforce each other while the noise will tend to cancel out (Finnigan et al. 2004). Figure 4.3 shows a diagram representing the Delay and Sum beamformer.

4.1.2 Near-field and Far-field Signals

As mentioned earlier, the aim of beamforming is to enhance certain signals and attenuate other signals based on their location in space relative to the microphone array receiving them. It therefore makes sense to consider how sound waves travel through the air.

The Delay and Sum beamformer presented assumes that the arriving sound waves are planar, allowing the calculation of a constant delay between consecutive sensors.

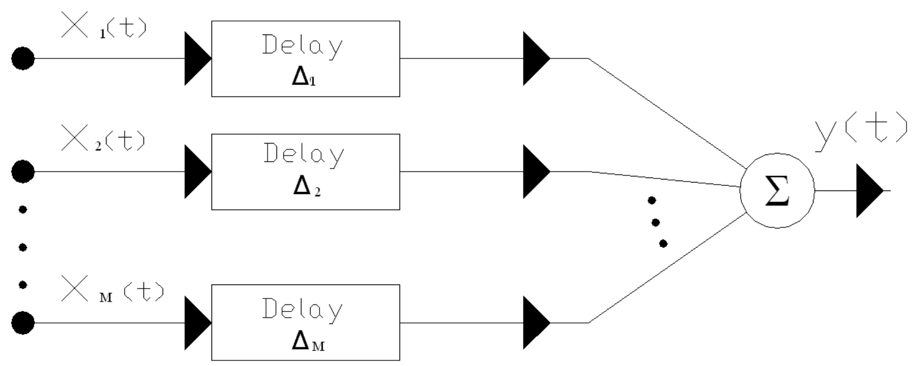

Figure 4.3: A block diagram of a Delay and Sum beamformer showing the delays associated with each microphone in the array (Finnigan et al. 2004).

However, this planar assumption is not always true. Sound waves actually radiate out from a source in a spherical pattern (Finnigan et al. 2004) but as the source moves away, they begin to appear parallel, thus allowing the assumption used in the Delay and Sum beamformer. Signals originating from such a distance that their arriving waves are parallel are known as Far-field signals or signals in the far-field (Finnigan et al. 2004). A Far-field signal is one that fulfils the following condition (McCowan et al. n.d.):

$$r > \frac{2L^2}{\lambda} \tag{4.6}$$

where

r is the distance from the source of the signal to the closest microphone in the array.

L is the array length.

λ is the wavelength of the frequency of interest.

If the desired signal does not fulfil this condition, it is know as a Near-field signal and a spherical propagation model must be used to calculate the time delays (McCowan et al. n.d.).

Microphone arrays used in a security context may be required to use either of these two models depending on its exact application. Voice authentication systems that would employ microphone arrays to reduce noise would probably use a Near-field model as the sound source would most likely be close to the array. A surveillance system, on the other hand, would use the Far-field model as the desired signal in this application would likely originate at a large distance from the array. This project focuses on signals originating in the Far-field and therefore all following theory relates to using the Far-field model.

4.1.3 Spatial Filter

The role of a beamformer is to discriminate signals from a desired direction from signals from all other directions. Therefore it is possible to think of a beamformer as a form of filter. Typical filters are used to take a signal and increase or decrease that signal's strength, but only over certain frequency ranges (Leis 2002). This is a similar concept to what we seek to do with a beamformer, which is to increase signals from a certain direction and reduce signals from all other directions - basically a spatial filter.

A Finite Impulse Response (FIR) filter is a filter whose transfer function always has zeros but does not have any poles, meaning that the system only has non-recursive coefficients (Leis 2002). This is analogous to describing some beamformers. Consider the general form difference equation, Equation 4.7, that can be used to describe filters.

$$\begin{aligned} y(n) &= (b_0 x(n) + b_1 x(n-1) + \ldots b_N x(n-N)) \\ &\quad -(a_1 y(n-1) + a_2 y(n-2) + \ldots a_M x(n-M)) \end{aligned} \tag{4.7}$$

where b_k controls how much of each of the previous kth input.
$x(n-k)$, goes into creating the current output.
a_k controls how much of the previous kth output.
$y(n-k)$, goes into creating the current output.

As previously mentioned, a FIR filter has no recursive coefficients, therefore its general form is given by Equation 4.9. A block diagram representing the filter can also be seen in Figure 4.4

$$y(n) = b_0 x(n) + b_1 x(n-1) + \ldots b_N x(n-N) \tag{4.8}$$

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) \tag{4.9}$$

Considering the block diagram for the simple Delay and Sum beamformer in Figure 4.5, it is possible to see a similarity between the two. That is, both take a given signal and,

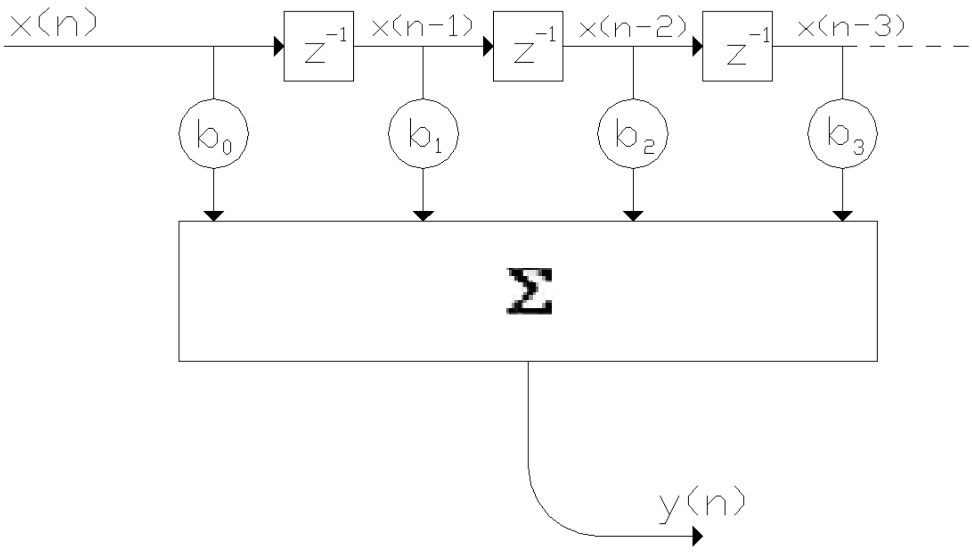

Figure 4.4: A block diagram representing a FIR filter (Leis 2002, p. 200).

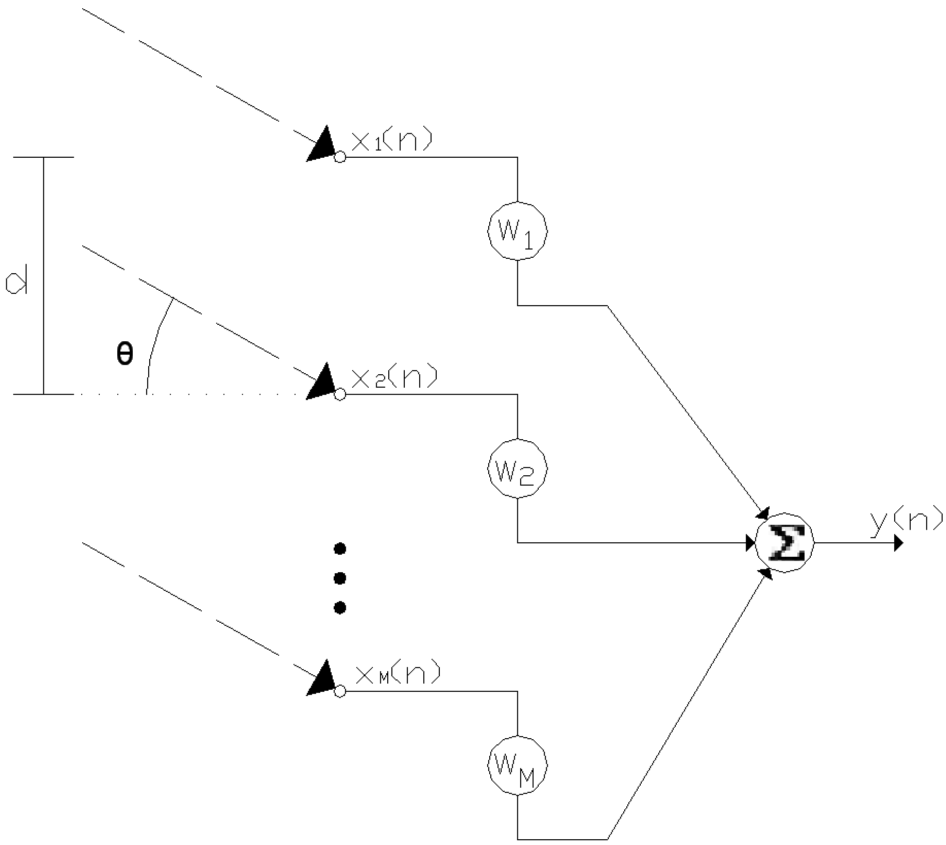

Figure 4.5: A block diagram representing a delay and sum beamformer (Van Veen & Buckley 1988, p. 8).

through a series of delays and weighting coefficients, the output is formed. The main difference is that in the FIR filter, the delays are created using hardware or software but in the beamformer the delays are the time delays between signals arriving at the sensors. Thus, the output of the array for a single source can be given by Equation 4.10 (Ward et al. 1998).

$$y(t) = \sum_{m=1}^{M} w_m x(t - \tau_m) \tag{4.10}$$

where

w_m is a complex weight representing a time delay, applied to the mth microphone to counter the time delay as a result of the microphone spacing.

$x(t - \tau_m)$ is the propagating sound wave.

τ_m is the time delay of the mth microphone from the reference microphone.

4.1.4 Complex Notation

A common way for describing the operation of microphone arrays is through the use of complex weights (Van Veen & Buckley 1988) which are applied to the output of each microphone. The output of the Delay and Sum beamformer is therefore given by the following equation (Ward et al. 1998):

$$y(k) = \sum_{m=1}^{M} w_m x_m(k) \tag{4.11}$$

where

$y(k)$ is the output of the beamformer.

M is the number of microphones.

w_m is the complex weight associated with each microphone in the array.

$x_m(k)$ is the k^{th} sample of the m^{th} microphone.

The complex weighting, w_m, introduced by Equation 4.11, as stated by McCowan et al. (n.d.) can be expressed in terms of its magnitude and phase components as:

$$w_m(f) = a_m(f) e^{j\varphi_m(f)} \tag{4.12}$$

where

$a_m(f)$ is the real amplitude weight, and

$\varphi_m(f)$ is the real, frequency dependent, phase weight.

To change the position of the directivity pattern's main lobe, $\varphi_m(f)$ is modified and to change the shape of the directivity pattern, $a_m(f)$ is modified. The phase weight $\varphi_m(f)$ is given by Equation 4.13.

$$\varphi_m(f) = -2\pi\alpha(m-1)d \tag{4.13}$$

where

θ is the look angle of the array.

$\alpha = \frac{\sin\theta}{\lambda}$ and λ is the wave-length of the frequency which is being evaluated.

As stated by McCowan et al. (n.d.), "A negative phase shift in the frequency domain corresponds to a time delay in the time domain." So Equation 4.13 can be easily converted back into a time delay:

$$\tau_m = \frac{\varphi_m(f)}{2\pi f} \tag{4.14}$$

$$\tau_m = \frac{-2\pi\alpha(m-1)d}{2\pi f} \tag{4.15}$$

$$\tau_m = \frac{-2\pi(m-1)d\sin(\theta)}{2\pi f\lambda} \tag{4.16}$$

$$\tau_m = \frac{-(m-1)d\sin(\theta)}{f\lambda} \tag{4.17}$$

remembering $c = f\lambda$

$$\tau_m = \frac{(1-m)d\sin(\theta)}{c} \tag{4.18}$$

which is the same equation as 4.5.

4.1.5 Spatial Response

One characteristic of a beamformer that can be used to help gauge its expected performance is the beamformer's spatial response. The spatial response is an indication of how well a beamformer will pass signals from a desired direction but reject signals from other directions (Ward et al. 1998). There are a number of different names for a beamformer's spatial response. Van Veen & Buckley (1988) define the response as the "beampattern" while McCowan et al. (n.d.) define the response as the "directivity

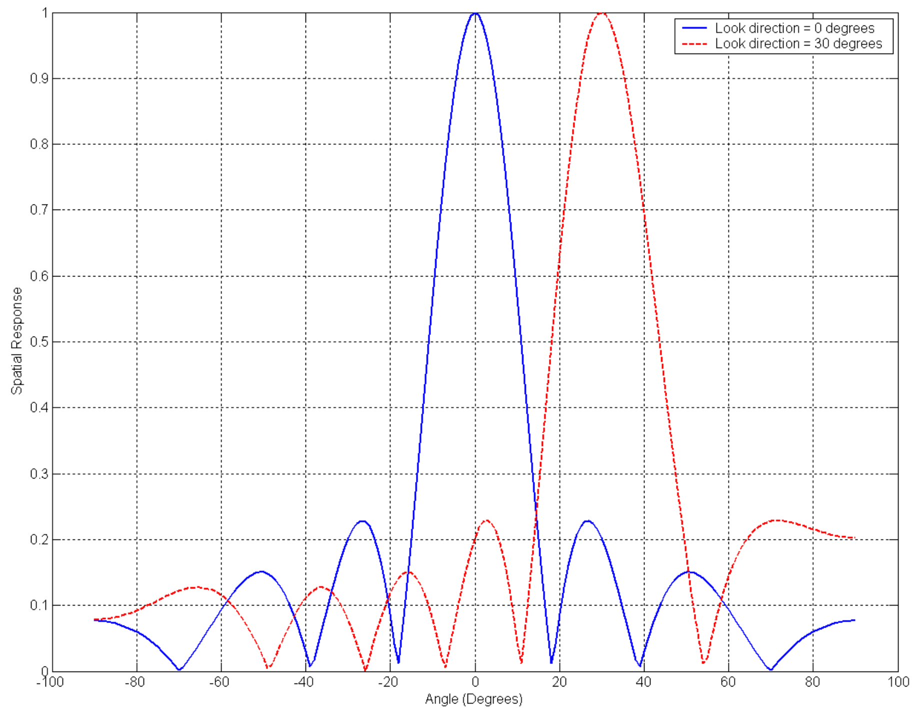

Figure 4.6: Spatial response of a microphone array at 8000Hz when pointing at 0^o and 30^o. Notice the oscillations in the stopbands of the response. Number of microphones = 8, d = 0.055m, $a_m = \frac{1}{M}$.

pattern". McCowan et al. (n.d.) define a Far-field beamformer's spatial response as:

$$R(f, \alpha') = \sum_{m=1}^{M} w_m(f) e^{j2\pi\alpha'(m-1)d} \tag{4.19}$$

where

$w_m(f)$ is the complex weight associated with the m^{th} microphone, defined in Equation 4.12.

$\alpha' = \frac{\sin\theta'}{\lambda}$ where θ' is the angle for which to calculate the response of the beamformer.

The spatial response of a Delay and Sum beamformer operated at a single frequency can be seen in Figure 4.6. In this figure, the solid blue line represents the spatial response when the look direction of the beamformer is 0^o and the dashed red line represents the spatial response when the look direction of the beamformer is 30^o. It is also possible to see oscillations in the stopbands. These oscillations are like what can be seen in a low order FIR filter without windowing (Leis 2002).

This is where the amplitude weight part of the complex weight has an effect, as it can

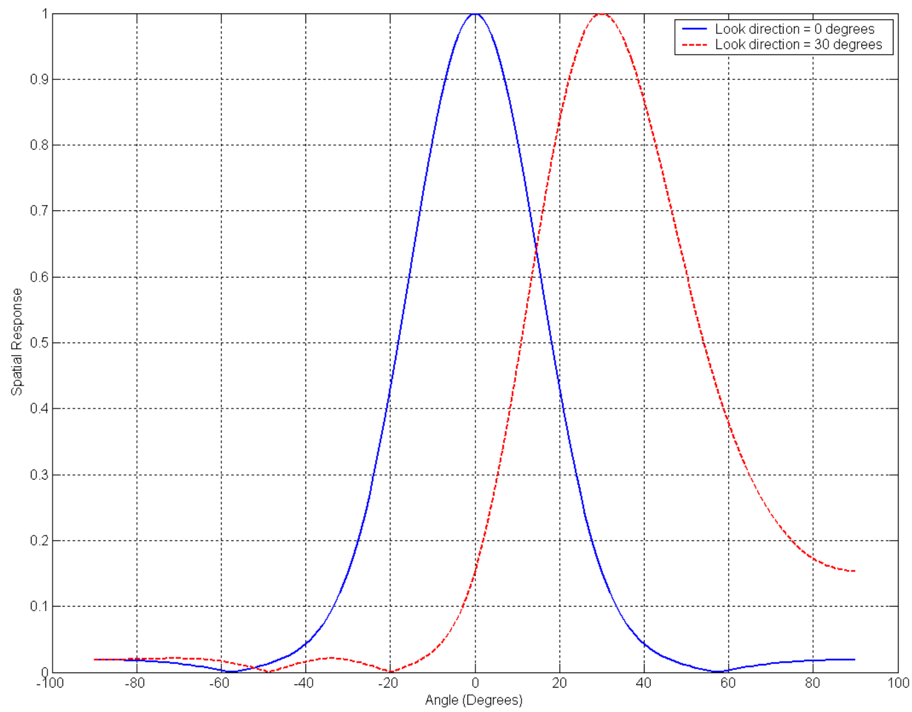

Figure 4.7: Spatial response of a microphone array at 8000Hz when pointing at 0^o and 30^o. Notice the oscillations in the stopbands have largely been removed, however, the beamwidth has increased. Number of microphones = 8, d = 0.055m, $a_m = h(n)$.

change the shape of the beam pattern and, therefore, they are chosen so as to trade-off the relationship between the average sidelobe level and beamwidth (Griffiths & Jim 1982). One possible method uses Chebyshev polynomials (Griffiths & Jim 1982), however, the effect of changing the amplitude weights can more easily be explained using a common windowing method used in FIR filter design - the Hamming window (Leis 2002). The equation for the hamming window is:

$$h(n) = 0.54 + 0.46 \cos\left(\frac{2n\pi}{N}\right) \tag{4.20}$$

The spatial response of the same beamformer is presented in 4.6 but, with the Hamming window applied to its complex weights it can be seen in Figure 4.7.

If the frequency response of the Delay and Sum beamformer was then to be calculated over a wide frequency range, Figures 4.8 and 4.9 are obtained. These figures demonstrate a significant characteristic of the Delay and Sum beamformer, that is, its response is frequency dependent, i.e. its response is not consistent over a large fre-

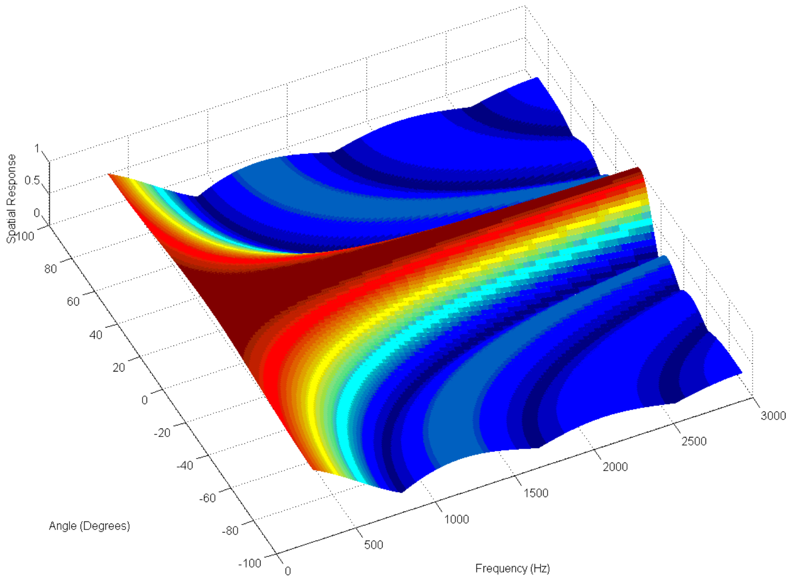

Figure 4.8: Spatial response of a Delay and Sum beamformer array pointing at 0 degrees. Microphones = 8, d = 0.055m, $a_m = \frac{1}{M}$.

quency range. As a result, the Delay and Sum beamformer is known as a narrowband beamformer (Ward et al. 1998). The code for generating the Delay and Response, as shown in Figures 4.8 and 4.9 can be seen in Appendix C.4.1.

4.1.6 Narrowband and Broadband Beamforming

Generally, a beamformer is either classed as a narrowband beamformer or a broadband beamformer (Van Veen & Buckley 1988). As such, narrowband beamformers are only able to effectively attenuate and reinforce narrowband signals and broadband beamformers are only able to attenuate and reinforce broadband signals. The characteristic of a signal that determines whether it is a narrowband or broadband signal, and hence requires either a narrowband or broadband beamformer, is the ratio of the signal's highest frequency to its lowest frequency (Chen, Yao & Hudson 2002). An example of a narrow-band signal is the 802.11b ISM wireless LAN system whose ratio of highest to lowest is 2.4835GHz/2GHz = 1.03, while typical audio signals have a ratio of 15kHz/30Hz = 500 and are therefore considered to be broadband (Chen et al. 2002).

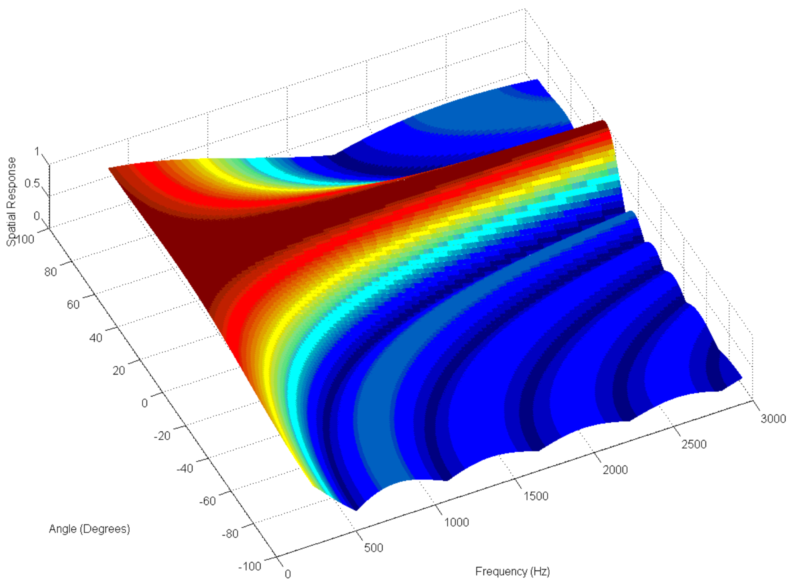

Figure 4.9: Spatial response of a Delay and Sum beamformer pointing at 30 degrees. Microphones = 8, d = 0.055m, $a_m = \frac{1}{M}$.

To determine why a Delay and Sum beamformer is a narrowband beamformer consider the following: If a microphone array has a set number of microphones spaced at a set distance, then the array has a fixed length. However, a microphone array can receive a range of different frequencies, each having a different wavelength, therefore, the relative size of the array, with respect to the signals impinging on the array, can vary (Ward et al. 1998). This means that the length of the array is not important, but rather the length of the array in wavelengths of an arriving frequency. Thus, for high frequency signals, which have a small wavelength, the array will look large, therefore resulting in a narrow main beam. However, lower frequency signals, with a large wavelength, will result in the array looking small and therefore resulting in the main beam being spread out (Ward et al. 1998). The end result is an inconsistent beamwidth across a range of frequencies, and hence a narrowband beamformer.

A broadband beamformer is one that maintains a constant spatial response over a wide range of frequencies and is sometimes called a "frequency-invariant" beamformer (Ward et al. 1998).

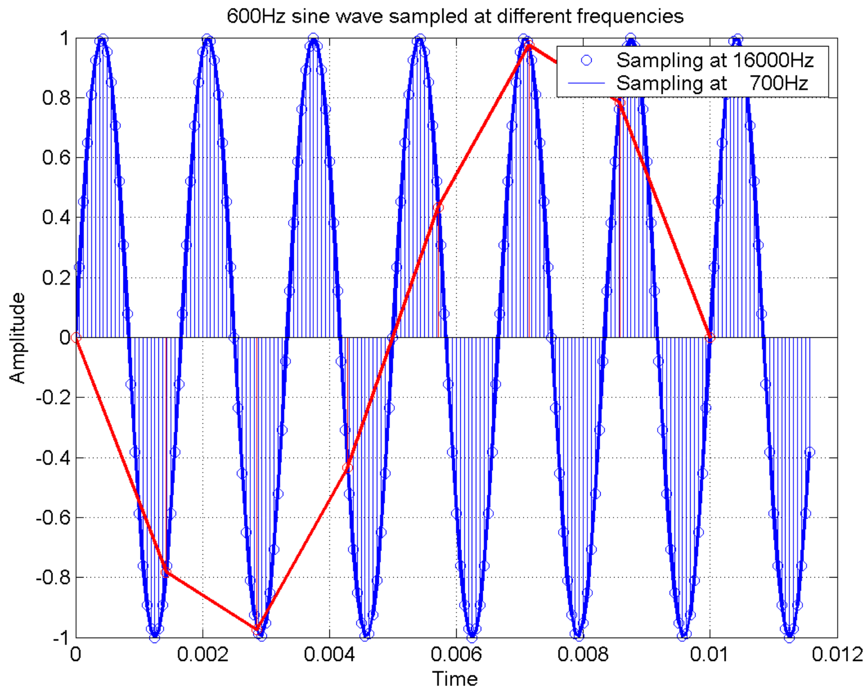

Figure 4.10: The effect of temporal aliasing. Here a 600Hz sine wave is sampled at 16000Hz and 700Hz. It is possible to see the aliasing when the signal is sampled at 700Hz.

4.1.7 Temporal Aliasing

Temporal aliasing is the result of an insufficient sampling speed for the signal being sampled (Leis 2002). It effectively makes the incoming signal appear slower than it really is. To avoid temporal aliasing, the sampling rate must be twice the highest frequency component of what is being sampled. This frequency is known as the Nyquist frequency. An example of temporal aliasing can be seen in Figure 4.10. Here a 600Hz sine wave is sampled at 16000Hz and 700Hz and the 700Hz waveform appears slower than the true wave - represented by the 16000Hz samples.

4.1.8 Spatial Aliasing

Spatial aliasing is the result of placing the microphones in the array too far apart for the signals that are being sampled and results in incorrect phase delay calculations of signals between microphones (Finnigan et al. 2004). The longest distance a wave will

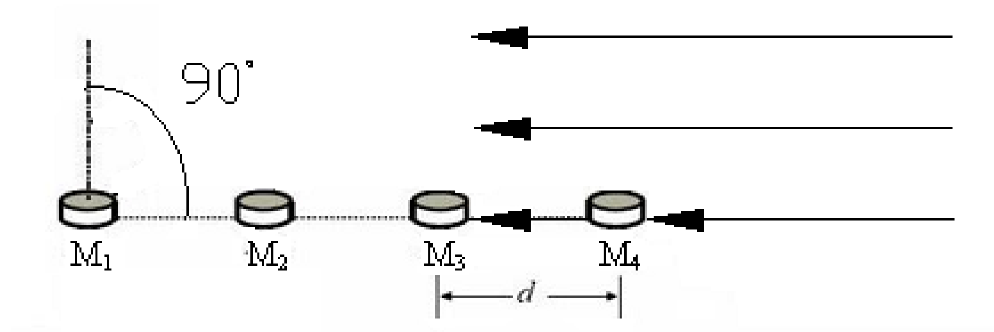

Figure 4.11: The longest distance a wave will have to travel is when its direction of arrival is 90 degrees.

have to travel is when its direction of arrival is 90 degrees from the perpendicular, see Figure 4.11.

Finnigan et al. (2004) tells us that "to avoid spatial aliasing, we would like to limit phase differences between spatially sampled signals to π or less because phase differences above π cause incorrect time delays to be seen between received signals." Therefore:

$$2\pi t f_{max} \leq \pi \tag{4.21}$$

Recall the equation for the time difference between each microphone:

$$t = \frac{d}{c}\sin\theta \tag{4.22}$$

Substituting 4.22 into 4.21 and rearranging gives:

$$2\pi\frac{d}{c}\sin\theta f_{max} \leq \pi \tag{4.23}$$

$$d \leq \frac{c}{2\sin\theta f_{max}} \tag{4.24}$$

As stated previously, the longest distance travelled by a wave occurs at 90^o, therefore:

$$d \leq \frac{c}{2f_{max}} \tag{4.25}$$

The shortest wavelength that is going to be received is given by:

$$\lambda_{min} = \frac{c}{f_{max}} \tag{4.26}$$

$$c = \lambda_{min} f_{max} \tag{4.27}$$

Rearranging equation 4.27 to find f_{max} and substituting into equation 4.25 gives:

$$d \leq \frac{\lambda_{min} f_{max}}{2 f_{max}} \tag{4.28}$$

$$d \leq \frac{\lambda_{min}}{2} \tag{4.29}$$

Brandstein & Ward (2001) call this condition the "Spatial Sampling Theorem" which states that the distance between each microphone, d, must be less than half the wavelength of the highest frequency sampled, λ_{min}.

The effect of using a spacing greater than $\frac{\lambda_{min}}{2}$ can be seen in Figure 4.12. Here, three microphones are used to record a 600Hz sine wave. The first microphone, represented by the blue line, acts as a reference point in space for the other two microphones. The second microphone, represented by the red line, is placed a quarter of a wavelength away from the first. The third microphone, represented by the green circles, is placed at one and one quarter wavelengths away from the first. The result, as can be seen from the graph, is that the signal from the third microphone looks as though it is only a quarter phase different from the first microphone, when in reality it is one and one quarter.

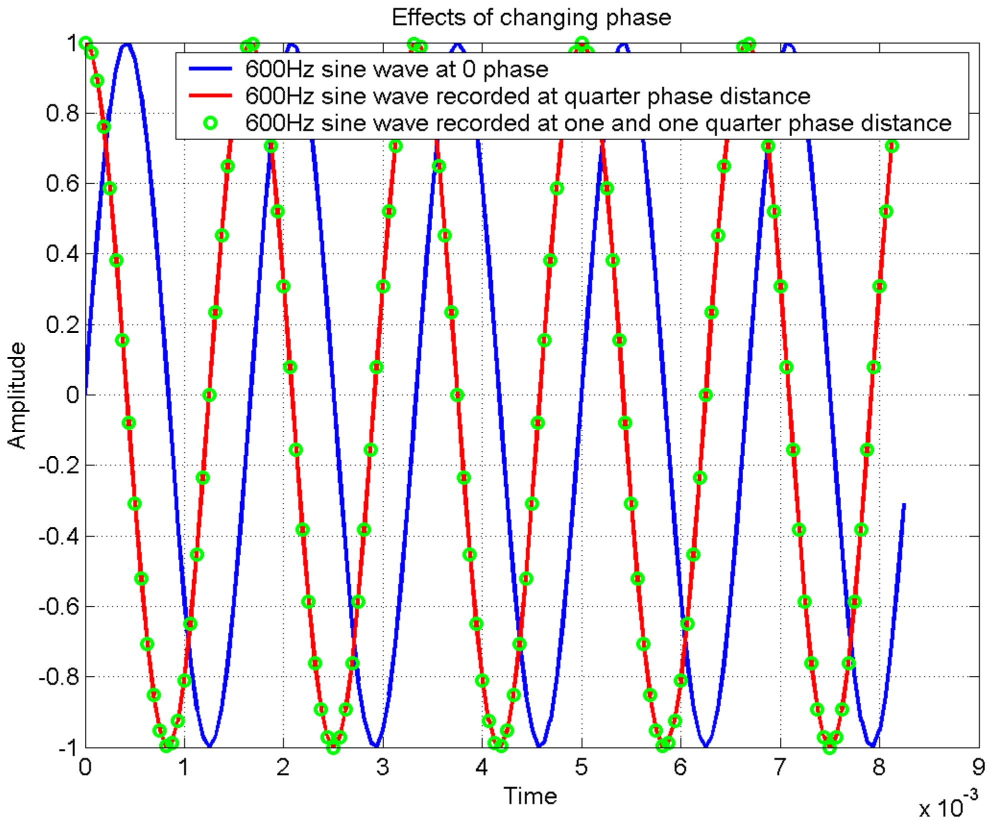

Figure 4.12: Using a spacing greater than $\frac{\lambda_{min}}{2}$ results in spatial aliasing. The samples represented by the blue line are sampled at a distance of $\frac{\pi}{4}$ wavelengths and the samples represented by the green circles are sampled at a distance of $\frac{5\pi}{4}$, however to the beamformer they both appear to have the same phase.

4.2 Chapter Summary

Beamforming is a method of spatial filtering that can be used to discriminate between signals based on their physical locations in space. It provides the ability to filter out signals that occupy the same temporal frequency band as a desired signal, provided it originates at a different location in space from the desired signal.

As well as security applications, beamformers, are used in several other applications:

- RADAR
- SONAR
- Communications
- Imaging
- Geophysical Exploration
- Astrophysical Exploration
- Biomedical Applications

The simplest form of beamforming is Delay and Sum beamforming, which achieves spatial filtering by correcting for delays introduced into sequential microphones based on the extra distance that has to be travelled by a wave to reach each microphone. The theory presented, however, assumes that the signals being received are Far-field signals, i.e. the waves originate at a large enough distance away that they are planar.

Beamformers can be classed as narrowband or broadband beamformers, depending on the type of signal that beamformer is designed to receive.

"Spatial Response" is the response pattern of the beamformer which can be used to gauge the beamformers expected response.

Chapter 5

Broadband Beamforming

The main goal of broadband beamforming is to generate a frequency-invariant beam over a wide frequency range, allowing the array to receive broadband signals and attenuate other broadband signals located at other points in space (Ward et al. 1998).

There are many ways that broadband beamforming can be accomplished. This chapter will cover some common broadband beamformers. These include the 2D Frequency filter, the Frost beamformer and the Generalised Sidelobe Canceller.

5.1 2D Frequency Filter

For arrays to achieve noise cancellation over a wide bandwidth, tapped-delay lines (Liu et al. 2005) are attached to the output of each microphone, see Figure 5.1. The design allows the beamformer to not only sample spatially, but also temporally (Van Veen & Buckley 1988).

The coefficients in this beamformer serve two purposes. Firstly, the coefficients determine the gain and phase response of the beamformer (Van Veen & Buckley 1988), since the tapped-delay line is essentially a FIR filter. Secondly, the coefficients also affect the spatial filtering characteristics of the beamformer (Van Veen & Buckley 1988). The

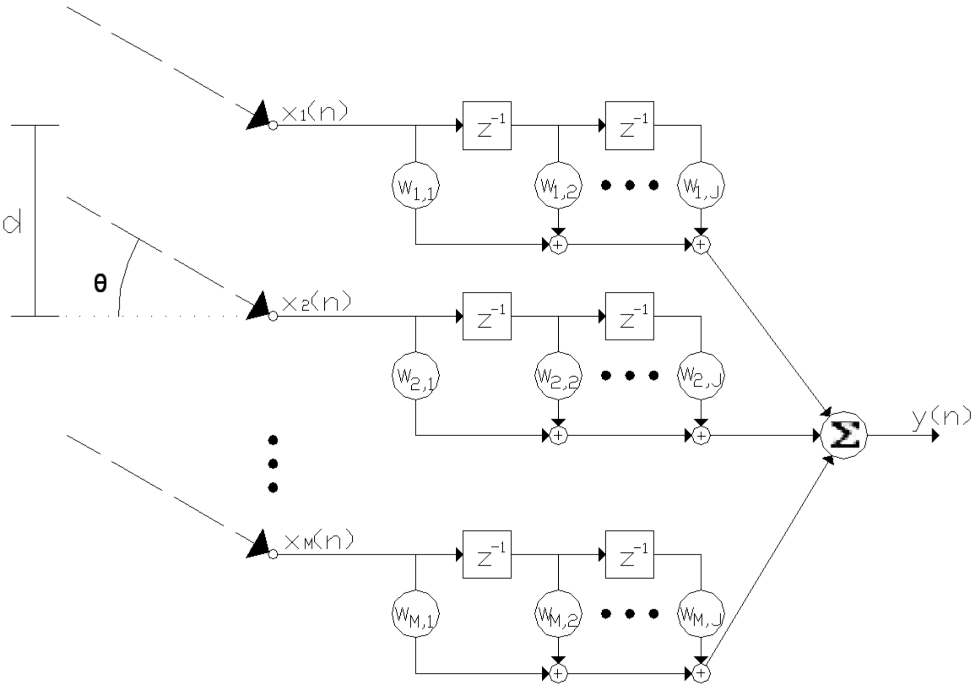

Figure 5.1: Microphone array utilising M microphones and tapped-delay line filters (Liu et al. 2005, p. 1).

equation that represents the response of this beamformer is given by equation 5.1.

$$R(f, \alpha') = \sum_{n=1}^{N} \sum_{k=1}^{K} w_{n,k}(f) e^{j2\pi\alpha'(n-1)d} e^{jk2\pi f T_s} \tag{5.1}$$

The goal of many broadband beamformers is, then, the selection of these weights to set the response and beam pattern. How these weights are chosen determines the class of the beamformer.

5.2 Beamformer Classification

Apart from narrowband and broadband, beamformers can be categorised into two main classes depending on how the weights are chosen (Van Veen & Buckley 1988). These are “data independent” and “statistically-optimum”.

Data-independent beamformers have their weights chosen so that the beamformer response is consistent and not dependent on the data arriving at the array. Designing such a beamformer is similar to classical FIR filter design (Van Veen & Buckley 1988). Examples of such designs include the Delay and Sum beamformer and the 2D Frequency filter. The 2D Frequency filter, however, can also be statistically optimum.

Statistically-optimum beamformers are those beamformers where the weights are chosen based on the statistics of the sounds arriving at the array (Van Veen & Buckley 1988). Essentially, the goal is to optimise the weights to exclude as much noise as possible in the output of the beamformer. Generally, the array data is not known in advance, therefore adaptive filters are used to adapt the weights to the statistically-optimum solution. Two examples of this kind of beamformer are the Frost beamformer and the Generalised Sidelobe Canceller (GSC).

5.3 Adaptive Frost Method

The Adaptive Frost Method is an adaptive beamformer which uses a constrained Least Mean Square algorithm to adapt the weights to the optimum solution (Griffiths & Jim 1982). Initially, the beamformer is pointed in the direction of interest using steering delays. Subject to a constraint, known as the "Frost Constraint", the output power is minimised, becoming a minimum-variance estimate of the filtered signal (Griffiths & Jim 1982).

The Frost beamformer is of particular interest because, as mentioned, it is an adaptive beamformer, which means the weights used in the beamformer adjust to new values every time a new set of samples is received from the array. This allowing it to adapt in situations where the interference signals are either spatially or temporally time varying (Griffiths & Jim 1982).

Griffiths & Jim (1982, p. 1) denote the output of the mth time delayed microphone as:

$$x_n(k) = s(k) + e_n(k) \tag{5.2}$$

where

$s(k)$ is the desired signal

$e_n(k)$ is the totality of the noise for the nth steered microphone.

If a tapped-delay line is applied to each steered output, the output of the beamformer can be calculated by (Griffiths & Jim 1982, p. 1) :

$$y(k) = \sum_{n=1}^{N} \sum_{l=-K}^{K} w_{n,l} x_n(k-l) \tag{5.3}$$

where

N is the number of microphones in the array

K gives the length of the tapped delay line to be $2K+1$ with the zero time reference at the filter midpoint

$w_{n,l}$ is the weight applied to the nth microphone at the lth delay

To simplify the equations, matrix notation is often used. Re-writing Equation 5.3 gives (Griffiths & Jim 1982):

$$y(k) = \sum_{l=-K}^{K} \mathbf{W}^T(l)\mathbf{X}(k-l) \tag{5.4}$$

where

$\mathbf{W}(l) = [w_{1,l}, w_{2,l}, ..., w_{N,l}]$ I.e. the weights for each microphone at the lth tapped-delay point.

$\mathbf{X}^T(k-l) = [x_1(k-l), x_2(k-l), ..., x_N(k-l)]$ I.e. the input at each microphone at the lth tapped-delay point.

To ensure that the desired signal has the required gain and phase response, the sum of each tapped-delay coefficient in the delay lines are constrained to specific values, which will be denoted by $f(l)$ to represent the value at a specific delay, l, (Griffiths & Jim 1982). This is analogous to constraining the entire beamformer to a non-adaptive FIR filter, see Figure 4.4, where each b element corresponds to a $f(l)$ element. Using matrix notation, the Frost constraint is represented by Equation 5.5. (Note: **1** is a column vector of ones.)

$$\mathbf{W}^T(l)\mathbf{1} = f(l) \tag{5.5}$$

Therefore, $f(l)$ represents the impulse-response of the system and the output of the beamformer, assuming that only the desired signals are present, is given by:

$$y_s(k) = \sum_{l=-K}^{K} f(l)s(k-l) \tag{5.6}$$

Griffiths & Jim (1982, p. 2) tell us that "the objective of linearly constrained adaptive beamforming is then to find filter coefficients, $\mathbf{A}(l)$, which satisfy [Equation 5.5] and simultaneously reduce the average value of the square of the output noise component." This means that the goal is to find coefficients that produce the desired filtering of the desired signal in terms of gain and phase and reduce the total output power of all other interfering signals present.

As the system is adaptive, the coefficients change with each new set of sensor inputs. This can be described by (Griffiths & Jim 1982):

$$\mathbf{W}_l(k+1) = \mathbf{W}_l(k) + \mathbf{\Delta}_l(k) \tag{5.7}$$

The term $\mathbf{\Delta}_l(k)$ is determined by the particular algorithm in use. In this case, it is Frost's, method which defines it as (Griffiths & Jim 1982):

$$\mathbf{\Delta}_l(k) = \mu y(k)(q_x(k-1)\mathbf{1} - \mathbf{X}(k-1)) - q_{a,l}(k)\mathbf{1} + \frac{1}{M}f(l)\mathbf{1} \tag{5.8}$$

where

$$q_x(k-1) = \frac{1}{M}\mathbf{X}^T(k-l)\mathbf{1} \tag{5.9}$$

$$q_{a,l}(k) = \frac{1}{M}\mathbf{W}_l^T(k)\mathbf{1} \tag{5.10}$$

The step, μ, is a scalar which determines the steady-state noise behaviour and the convergence rate. It is also normalised by the total power contained in the beamformer and given by the following equations (Griffiths & Jim 1982):

$$\mu = \frac{\alpha}{P(k)} \tag{5.11}$$

$$P(k) = \sum_{m=1}^{M}\sum_{l=-K}^{K} x_m^2(k-l) \tag{5.12}$$

where

α controls the convergence of the system and, according to Griffiths & Jim (1982), is assured if $0 < \alpha < 1$

The structure of the Frost method can be seen in Figure 5.2

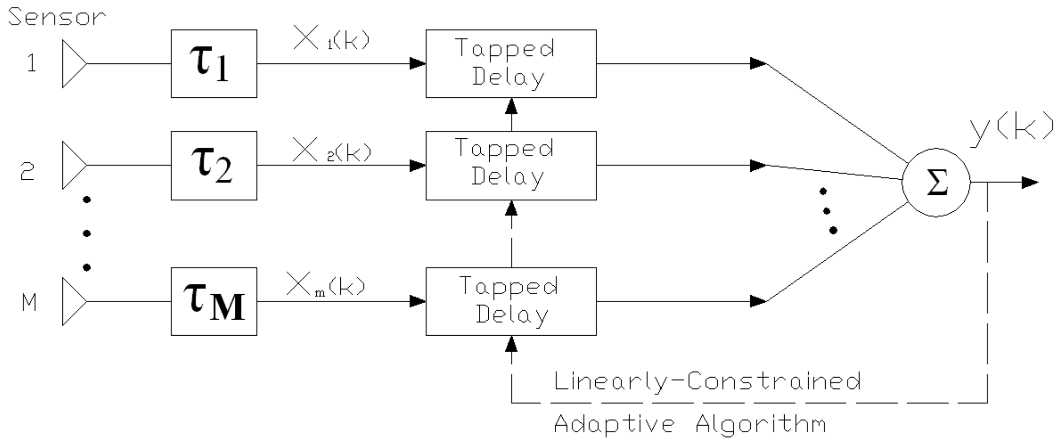

Figure 5.2: A block diagram of the adaptive Frost method (Griffiths & Jim 1982, p. 29).

5.4 Generalized Sidelobe Canceller

The Generalized Sidelobe Canceller (GSC), according to Griffiths & Jim (1982), can be seen as an alternate implementation to Frost's algorithm, however, it uses an unconstrained Least Mean Square algorithm to achieve the Frost constraint (Oteri & Waterston 2001).

The GSC beamformer and the Frost beamformer, however, have certain problems associated with them which many variations of the beamformers aim to address. These will be discussed in a subsequent section.

The structure of the Generalized Sidelobe Canceller can be seen in Figure 5.3. As can be seen in the diagram, there is both an upper path and a lower processing path.

The upper processing path is a conventional, non-adaptive beamformer, which uses fixed amplitude weights, denoted by $w_{c1}, w_{c2}, ..., w_{cM}$, and produces the signal given by $y_c(k)$:

$$y_c(k) = \mathbf{W}_c^T \mathbf{X}(k) \tag{5.13}$$

where
$\mathbf{W}_c^T = [w_{c1}, w_{c2}, ..., w_{cM}]$ i.e. The set of fixed weights.

The $\mathbf{W}_c^T$ coefficients are usually chosen in order to make a trade-off between the average

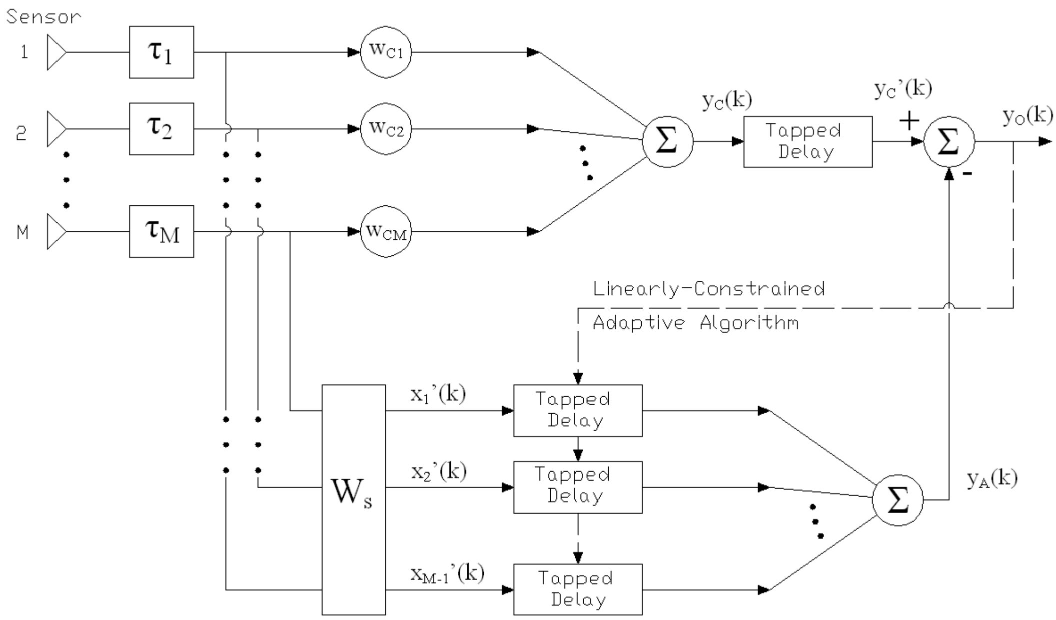

Figure 5.3: Block diagram of Generalized Sidelobe Canceller (Griffiths & Jim 1982, p. 30).

sidelobe level and beamwidth. One way to choose these coefficients, which is often used in digital filter design, is to use Hamming window coefficients (Leis 2002). Griffiths & Jim (1982) also tell us that a widely used method for selecting these coefficients involves using Chebyshev polynomials. The output of this stage is then filtered by the constraint values, $f(l)$, introduced in the previous section. This is to ensure that the desired signal has the required phase and gain response and gives the signal $y_c'(k)$:

$$y_c'(k) = \sum_{l=-K}^{K} f(l) y_c(k-l) \tag{5.14}$$

The lower path of the GSC, as seen in Figure 5.3, is what Griffiths & Jim (1982) call the "Sidelobe Cancelling Path". The path consists of two elements. The first element is the blocking matrix, W_s, which is designed to block the desired signal from the lower path. Conceptually, the blocking matrix is a filter and, as the desired signal, $s(k)$, is common to each sensor output, blocking can be achieved if each row of the blocking matrix adds up to 0. The output of the filter is then given by:

$$\mathbf{X}'(k) = W_s \mathbf{X}(k) \tag{5.15}$$

Griffiths & Jim (1982) also require that each row in the blocking matrix, represented by b_m^T, is linearly independent and satisfies the following equation:

$$\mathbf{b}_m^T \mathbf{1} = 0 \tag{5.16}$$

This means that the output of the blocking matrix, $\mathbf{X}'(k)$, can have no more than $M-1$ linearly independent components. This also means that the row dimension of the blocking matrix, W_s, must be $M-1$ or less.

The second part of the lower path is a set of tapped-delay lines, also known as a "multiple-input canceller" (Hoshuyama, Sugiyama & Hirano 1999), containing $2K+1$ adaptive weights. The output of the lower path of the GSC can now be represented by (Griffiths & Jim 1982):

$$y_A(k) = \sum_{l=-K}^{K} [\mathbf{A}'_l(k)]^T \mathbf{X}'(k-l) \tag{5.17}$$

The output of the lower path is now subtracted from the fixed beamformer in the upper path. This reduces the power of the noise terms in the GSC output because the signal in the lower path, $y_A(k)$, only contains noise terms, as the desired signal was removed by the blocking matrix. The final output of the GSC is given by (Griffiths & Jim 1982):

$$y(k) = y'_c(k) - y'_A(k) \tag{5.18}$$

The last thing that needs to be defined are the filter adaptive coefficients, $\mathbf{A}'_l(k)$, for the tapped-delay line in the lower path. What is desired are coefficients which minimize the power of $y_A(k)$ contained in the output, $y(k)$. Griffiths & Jim (1982, p. 30) tell us that 'the unconstrained Least Mean Square (LMS) algorithm can be employed to adapt the filter coefficients to the desired solution':

$$\mathbf{A}'_l(k) = \mathbf{A}'_l(k) + \mu y(k) \mathbf{X}'(k-l) \tag{5.19}$$

where

μ is the step size normalised by the power contained in $\mathbf{X}'(k-l)$.

5.5 Other Beamformers

There is, however, a problem associated with adaptive beamformers - signal cancellation as the result of steering errors (Hoshuyama et al. 1999). According to Hoshuyama et al. (1999), these errors may be caused by errors in the "microphone positions, microphone gains, reverberation, and target direction." Hoshuyama et al. (1999) go on to say that these errors are therefore inevitable in real microphone arrays and can be a serious problem. As a result of this problem, several beamformers have been developed to avoid target-signal cancellation. A beamformer that is resistant to errors is known as a "Robust" beamformer. Many robust beamformers have been proposed (Hoshuyama et al. 1999, Zou, Liang Yu & Lin 2004).

Traditional adaptive algorithms can also suffer from another major flaw - the large computational complexity associated with large tapped-delay lines which require their weights to be updated on every sample. One proposal to address this problem is given by Sekiguchi & Karasawa (2000, p. 277) and involves using a "wideband beamspace adaptive array that uses FIR fan filters to construct a multibeam forming network". This is essentially a series of 2D Frequency Filters, presented previously, with non-adaptive weights pointing in different directions. The output of each beamformer is then combined using adaptive weights, one for each beamformer, with the main advantage being fewer adaptive weights, see Figure 5.4. In this figure, BFN stands for "Beamforming Network" which are the 2D frequency filters presented previously.

Instead of using Uniform Linear Arrays, Ward et al. (1998, p. 4) tell us that another "common approach ... is to use harmonically nested subarrays. In this case, the array is composed of a set of nested, equally-spaced subarrays, each of which is a single-frequency design. The outputs of the subarrays are then combined by appropriate bandpass filtering. The effect of harmonic nesting is to reduce the beamwidth variation to that which occurs within a single octave." Figure 5.5 gives an example of such an array.

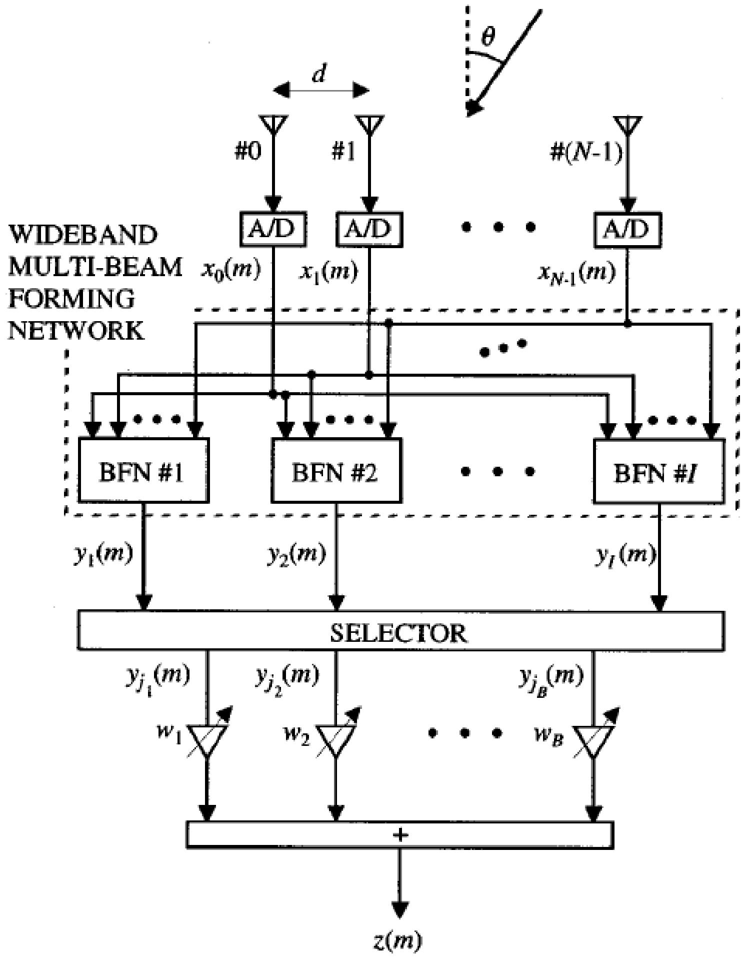

Figure 5.4: Beamspace adaptive array for wideband signals. (Sekiguchi & Karasawa 2000, p. 278)

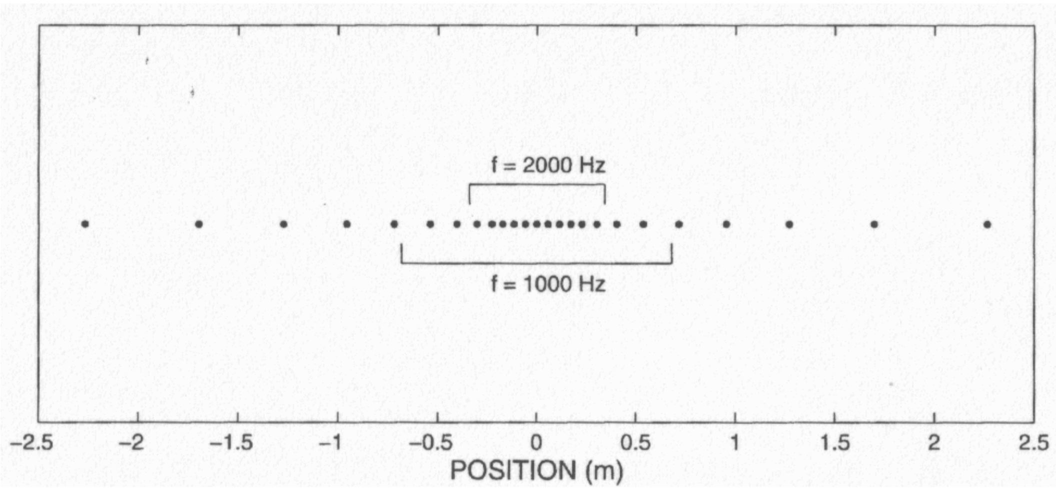

Figure 5.5: Example of a possible array geometry for a broadband beamformer using nested subarrays (Brandstein & Ward 2001, p. 13).

5.6 Chapter Summary

To process speech and other broadband signals, beamformers that generate frequency-invariant beams are required. This is often achieved through the use of tapped-delay lines attached to the output of each microphone. The coefficients in the tapped-delay lines serve two purposes. Firstly they set the gain and phase response of the beamformer and secondly, they affect the spatial filtering characteristics of the beamformer. The simplest form of beamformer to implement this configuration is the 2D frequency filter. It is then important to consider how the weights of the beamformer are chosen.

How the weights are chosen determines the class of the beamformer. Apart from narrow-band and broadband, there are two main classes of beamformer, data independent and statistically-optimum. Data independent beamformers choose their weights to produce a constant beam response independent of the data arriving at the array. Statistically-optimum beamformers choose their weights based on the statistics of the arriving data. This data, however, is generally not known in advance. Therefore, adaptive filters are used to adapt the weights to the statistically-optimum solution.

Two types of adaptive beamformers have been covered - the Adaptive Frost beamformer and the Generalized Sidelobe Canceller. The Adaptive Frost beamformer employs a constrained Least Mean Square algorithm to adapt the beamformer weights to the optimum solution. The Frost beamformer forms the basis for many other beamformers, including the Generalized Sidelobe Canceller which uses an unconstrained Least Mean Square algorithm to achieve the frost constraint.

There are, however, a few problems associated with adaptive beamformers. The first is signal cancellation as a result of steering errors and the second is the high computational requirement of adapting long tapped-delay lines. However, there have been several other beamformers proposed to address these problems.

Chapter 6

Beamformer Implementation

To demonstrate the potential of microphone arrays in security applications, three different beamformers were implemented as well as a microphone array simulation, in MATLAB. The beamformers that were implemented include the Delay and Sum beamformer, the Adaptive Frost Beamformer and the Generalized Sidelobe Canceller. Additionally, to try and validate the models using real-world data, a experimental array was set up using off-the-shelf studio equipment.

6.1 Microphone Array Model

The microphone array model function was written to simulate a real-world microphone array, the code for which can be found in Appendix C.1. This function takes the sampling frequency, the number of microphones in the array, the spacing between the microphones, the speed of sound, a N-by-M matrix of source signals (N source signals of M samples) and a 1-by-N vector of angles (in radians) corresponding to the direction of arrival for each of the source signals.

Into an empty N-by-M matrix, where N equals the number of microphones and M equals the length of the source signals, an appropriately delayed version of each signal was added. The assumption, when doing this, is that signals coming from a negative angle arrive at a time delay of 0, or no phase shift, at the first microphone, M_1, and

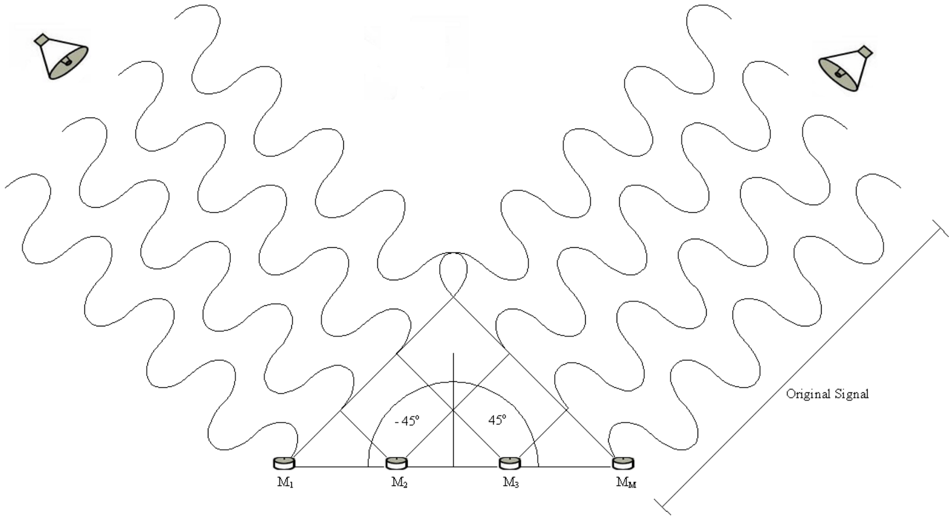

Figure 6.1: This diagram demonstrates how the microphone array simulation code combines signals coming from different directions. Signals from a negative direction are in phase at the first microphone and signals from a positive direction are in phase at the last microphone. All other microphones receive appropriately delayed versions of the different signals.

signals coming from a positive angle arrive at a time delay of 0, or no phase shift, at the Mth microphone, M_M, when there are M microphones. This can be seen in Figure 6.1.

The delays are easily calculated. For example, for a signal arriving at the array from an angle of -60^o or 1.047 radians, the time delay for the signals arriving at each consecutive microphone, when the spacing, d, is 0.055m and the speed of sound, c, is 344m/s, is:

$$\begin{aligned}\Delta_t &= \frac{d}{c}\sin(\theta)\\ \Delta_t &= \frac{0.055}{344}\sin(1.047)\\ \Delta_t &= 1.259\times10^{-4}\text{ seconds}\end{aligned}$$

The signals, however, are recorded as discrete samples, so this time delay needs to be converted to a sample delay. Assuming a sampling frequency of 48000Hz, then the delay becomes:

$$\begin{aligned}\Delta_s &= \frac{\Delta_t}{\frac{1}{Fs}}\\ \Delta_s &= \frac{1.384\times10^{-4}}{\frac{1}{48000}}\end{aligned}$$

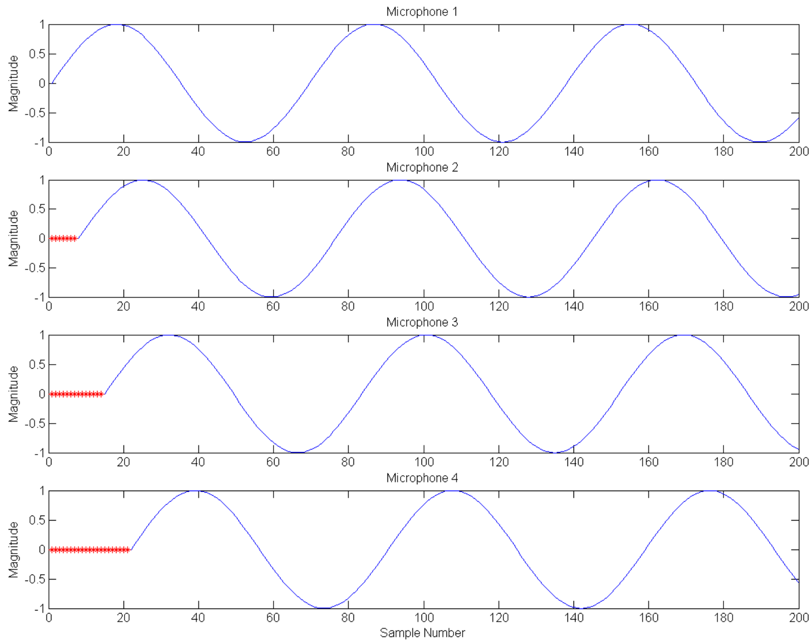

Figure 6.2: The output of the microphone array model when a $700Hz$ sine wave arrives from an angle of -60^o. The stars represent the delayed samples.

$$\Delta_s = 6.645 \text{ samples}$$

It is not possible to have 0.645 of a sample, therefore the value is rounded to 7 samples. An example of this operation can be seen in Figure 6.2.

Each microphone in the array will then add a copy of this signal, however, every microphone after the first will have the signal delayed by 7 samples by inserting zeros. This method of simulating a microphone array is valid considering sound is made up of travelling compression waves (Microsoft Encarta Encyclopedia Standard 2004) where each particle moves back and forth within a small distance as the wave expands outwards (Russell 2001). Figure 6.3 depicts a sound pressure wave radiating out from a source.

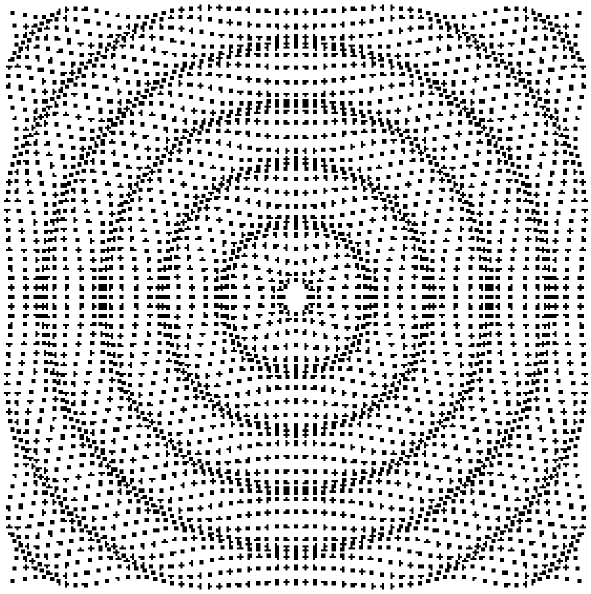

Figure 6.3: A monopole sound source radiating sound in all directions.
(Russell 2001)

6.2 Delay and Sum Beamformer Implementation

The main aim of the Delay and Sum beamformer is to counter the delays in the microphone signals resulting from the difference in distances travelled. The MATLAB code for this beamformer can be seen in Appendix C.2.1. This function takes a series of variables including the speed of sound, the sampling rate, the number of microphones in the array, the spacing between the microphones, a boolean indicating whether to apply a Hamming window, the look direction of the beamformer, and a N-by-M matrix of source signal vectors, where N is the number of microphones and M is the length of the signals.

The function first generates the microphone weights, using a Hamming window if required. It then performs the beamforming algorithm, which is achieved by calculating and applying delays, based on the look angle, to the microphone signals and then adding the signals together.

If the example presented in the previous section were to be used and the beamformer was steered to an angle of -60^o, or 1.047 radians, the same delay of 7 samples would be

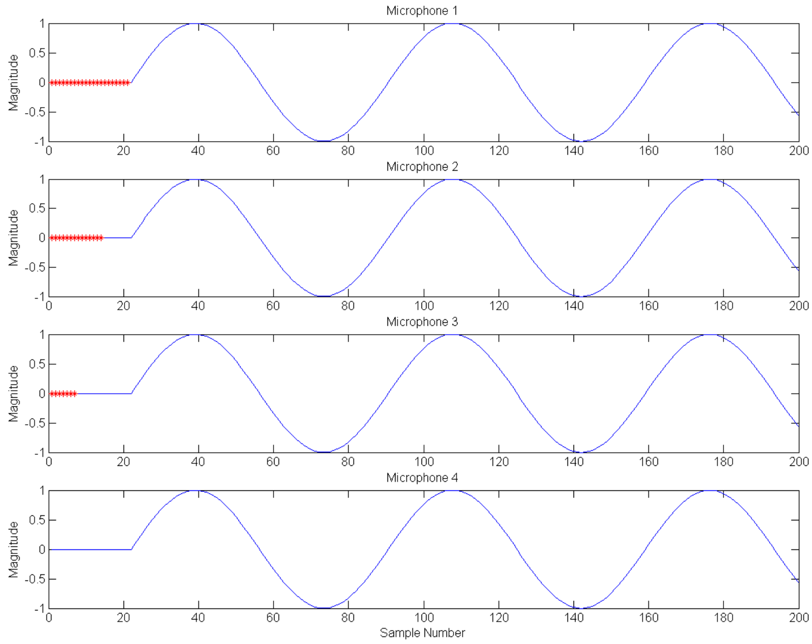

Figure 6.4: The output of the Delay and Sum beamformer when a $700Hz$ sine wave arrives from an angle of -60^o and the look direction is -60^o. The stars represent the delayed samples.

calculated. Recall that in the array model, signals arriving from the negative direction reach the microphone on the left-hand side first (microphone 1) and the others record an increasingly delayed version. Therefore, to bring the signals back into phase, the beamformer applies the most delay to the left-hand microphone with decreasing delay for each successive microphone. An example of this can be seen in Figure 6.4, where the stars represent the delayed samples.

6.3 Frost Adaptive Beamformer Implementation

The Frost beamformer uses a constrained adaptive Least Mean Square algorithm to adapt the weights in its tapped-delay line to the optimum solution, the code for which can be seen in Appendix C.2.2. This function takes a series of variables including the speed of sound, the sampling rate, the number of microphones in the array, the spacing between the microphones, the order of the tapped-delay lines, the look direction of the beamformer, and a N-by-M matrix of source signal vectors, where N is the number of microphones and M is the length of the signals.

The function first calculates the highest frequency that the beamformer can process without spatial aliasing occurring. This value is then used to calculate the constraint coefficients, $f(l)$, which constrain the tapped-delay line coefficients to produce a set gain and phase response. This was done using a simple FIR design method presented by Leis (2002). Leis (2002, p. 200) states that “the time-domain impulse response of a filter corresponding to a given (desired) frequency-response may be calculated from the Inverse Fourier Transform of the desired frequency response:”

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(w)e^{jnw}dw$$

When designing the filter, the frequency response must be mirrored in the negative frequency range to allow complex numbers to cancel as conjugates in the frequency domain and real numbers in the time domain (Leis 2002). Using the current equation will result in an infinite length impulse response. Therefore, the number of coefficients needs to be limited to be used. Using an odd length filter, the response can be truncated by calculating the response over the range (Leis 2002):

$$-\frac{N-1}{2} \leq n \leq +\frac{N-1}{2} \tag{6.1}$$

The function written to perform this task has been adapted from Leis (2002, p. 202) and is called filterCoeffs.m, the code for which can be seen in Appendix C.3.1.

The function then performs the beamforming algorithm. The first step of Frost beamforming is to synchronise the arriving signals. This is the same operation performed by the Delay and Sum beamformer. Once the signals are aligned, the constrained LMS algorithm can be applied to the signals by altering the tapped-delay line weights.

The LMS algorithm can then be applied to the signals, the results of which can be seen in section 7.3.

6.4 Generalized Sidelobe Canceller Beamformer Implementation

The Generalized Sidelobe Canceller beamformer uses an unconstrained adaptive LMS algorithm to adapt the weights in its tapped-delay line to the optimum solution, the code for which can be seen in Appendix C.2.3. This function takes a series of variables including the speed of sound, the sampling rate, the number of microphones in the array, the spacing between the microphones, the order of the tapped-delay lines, the look direction of the beamformer, and a N-by-M matrix of source signal vectors, where N is the number of microphones and M is the length of the signals. It then performs the GSC beamforming algorithm.

The first step of the GSC, like the Frost beamformer, is to synchronise the arriving signals. Once this is done the signals are split between two paths. The top path is implemented as Delay and Sum beamformer followed by a FIR filter to set the gain and phase response. The set of weights, W_c^T, in the fixed beamformer are chosen to control the sidelobe level, and for this task, a Hamming window has been chosen. The following filter has its coefficients chosen in the same way as presented in the Frost beamformer.

In the lower path, the first component is the blocking matrix which is used to block the desired signal. One of the matrices that have been used in the implementation is given below:

$$W_s = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix} \tag{6.2}$$

Every row in this matrix satisfies the condition:

$$b_m^T \mathbf{1} = 0 \tag{6.3}$$

For example, if $m = 1$, i.e. the first row:

$$\begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0 \tag{6.4}$$

In addition all rows of the matrix need to be linearly independent, that is, one vector cannot be made from any of the other vectors. Represented mathematically, there should exist no numbers, a_m, such that:

$$a_1 b_1^T + a_2 b_2^T + \cdots + a_m b_m^T = 0 \tag{6.5}$$

To prove this condition let a, b and c be real numbers such that:

$$a(1, 1, -1, -1) + b(1, -1, -1, 1) + c(-1, 1, -1, 1) = (0, 0, 0, 0)$$

Then:

$$(a + b - c, a - b + c, -a - b - c, -a + b + c) = (0, 0, 0, 0)$$

$$\begin{aligned} \therefore a + b - c &= 0 \\ a - b + c &= 0 \\ -a - b - c &= 0 \\ -a + b + c &= 0 \end{aligned}$$

Solving for a, b and c found that $a = 0$, $b = 0$ and $c = 0$ and therefore the vectors are linearly independent.

The LMS algorithm of the multiple-input canceller can then be applied to the outputs of the blocking matrix, adjusting its output to match the output of the fixed beamformer as closely as possible and then subtracting from it, leaving the desired signals. Finally, the energy of the output signal is normalised with respect to the energy of the microphones.

Figure 6.5: The setup used to record the test sound files.

6.5 Obtaining Real World Data

In an attempt to confirm the results obtained using the simulation, real data was recorded using a experimental microphone array.

The microphone array was constructed in room R127 at the University of Southern Queensland, which contained a recording studio. The studio has the ability to record on four channels simultaneously, therefore limiting the maximum size of the array that can be set up to four microphones. The recording was performed using Rode Broadcaster microphones which fed directly into a mixer, a Soundcraft K1. Here, the levels of each microphone could be adjusted before being sent to the computer which did the recording. The sound was recorded using Pro Tools V6.9 on a Mac running OS 10.3. An image of the setup can be seen in Figure 6.5.

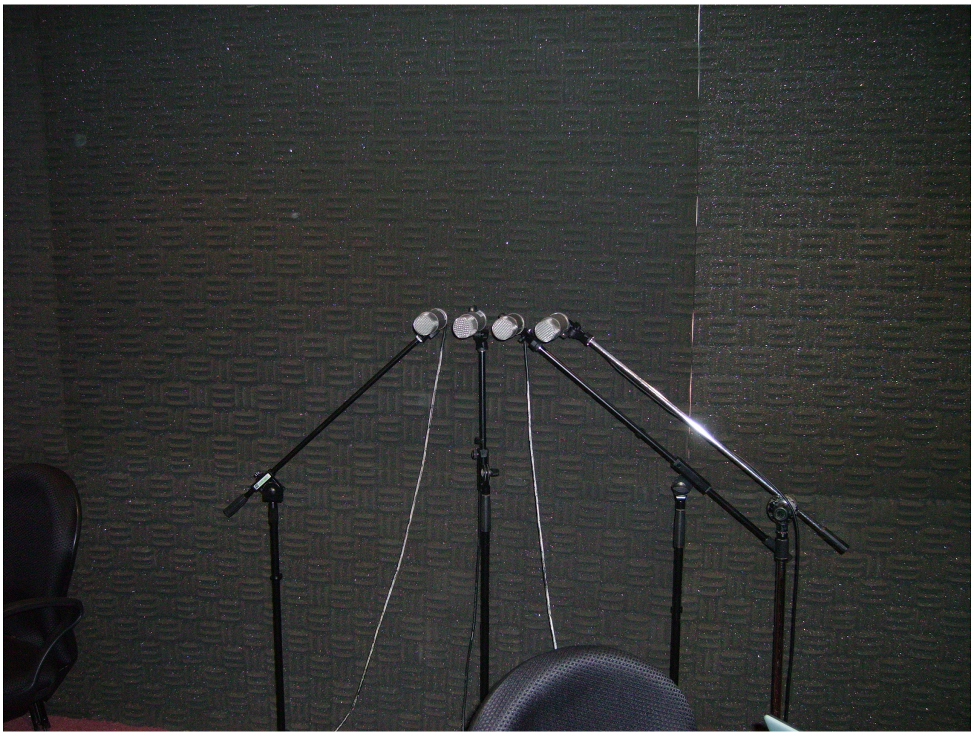

Figure 6.6: Experimental setup of a four element microphone array, spaced at $0.1m$.

6.5.1 The Process

During recording session, three different microphone configurations were used and each microphone was attached to a stand which was adjusted for each microphone. The configurations varied in the amount of spacing between each microphone. The first configuration had $0.15m$ between the centre of each microphone, the second had $0.1m$ between each microphone and the final configuration had $0.055m$ between each microphone. This was the closest that the microphones could be placed. Figures 6.6 and 6.7 show the second configuration and Figure 6.8 shows the third configuration.

Each array configuration imposes a limit on the maximum frequency that the array is capable of sampling spatially without suffering spatial aliasing. Recall the spatial sampling theorem which stated that the smallest wavelength received by the array must be twice the length of the distance between the microphones. In other words, the distance between each microphone must be half the length of the smallest wavelength that is to be sampled.

Figure 6.7: Experimental setup of a four element microphone array, spaced at $0.1m$.

Figure 6.8: Experimental setup of a four element microphone array, spaced at $0.055m$.

For the first configuration with a spacing of $0.15m$, the smallest wavelength received was:

$$\begin{aligned}
\lambda_{min} &= d \times 2 \\
\lambda_{min} &= 0.15 \times 2 \\
\lambda_{min} &= 0.3
\end{aligned}$$

As frequency increases, its wavelength gets smaller, therefore the wavelength of $0.3m$ corresponds to the highest frequency that this array configuration can handle. This frequency is:

$$\begin{aligned}
f_{max} &= \frac{c}{\lambda_{min}} \\
f_{max} &= \frac{344}{0.3} \\
f_{max} &\approx 1150\text{Hz}
\end{aligned}$$

The next thing that needs to be ensured is that all the sound sources are placed in the far field so that a planar signal radiation model can be used. Using Equation 4.6, the minimum distance required is:

$$\begin{aligned}
r &> \frac{2L^2}{\lambda} \\
r &> \frac{2 \times (3 \times 0.15)^2}{0.3} \\
r &> 1.35
\end{aligned}$$

Applying the above equations to the 0.1m configuration gives a minimum wavelength of 0.2m, a maximum frequency of approximately 1700Hz and a minimum distance to the far field of 0.9m. For the 0.055m configuration, these values are respectively 0.11m, 3100Hz and 0.5m.

6.5.2 The Recordings

For each configuration, a number of recordings were made. Each recording involved placing two sound sources in the room at pre-defined angles in relation to the array. The angles and the frequencies of the sound sources were chosen so that a variety of data would be recorded. This was done to allow tests to be carried out on the effects of changing these different parameters.

A summary of the recordings made, which have been used in the evaluation of the beamformers in the following chapter is presented in Table 6.1.

ID	Spacing (m)	Signal 1	Angle 1	Signal 2	Angle 2
1	0.15	1000Hz	45^o	800Hz	-45^o
4	0.10	1600Hz	60^o	1400Hz	-60^o
6	0.10	600Hz	45^o	900Hz	-45^o

Table 6.1: Summary of results for Frost beamformer using simple tones

Appendix B.1, lists all sound files provided on the accompanying CD. Note that only those recordings from which meaningful data could be obtained are included.

6.6 Chapter Summary

Three beamforming algorithms have been implemented and they are the Delay and Sum beamformer, the Frost beamformer and the Generalized Sidelobe Canceller beamformer, the code for which can be seen in appendices C.2.1 C.2.2 and C.2.3 respectively. In addition, a microphone array simulator has been written to test the beamformers and its code can be seen in appendix C.1.

To test the validity of the microphone array simulation, real recordings were made using a experimental microphone array.

Chapter 7

Results and Discussion

The three beamformers that have been described in the previous chapters, the Delay and Sum, the Adaptive Frost and the GSC, have all been tested using MATLAB.

This chapter presents the procedures used to test each of the beamformers and the results obtained to determine the effect of:

1. the number of microphones in the array;
2. a Hamming window on the Delay and Sum beamformer;
3. the distance between the microphones on the Delay and Sum beamformer;
4. the length of tapped-delay lines and the number of microphones on the Frost and GSC beamformers.

Note that the effect of changing the distance between microphones has not been found for the Frost beamformer and GSC beamformer because the goal was to process broadband signals, in particular, speech. As presented in the previous chapter, a microphone spacing of 0.055m allows frequencies up to 3100Hz to be processed without spatial aliasing. This is slightly higher than what is required for telephone quality speech (Beasley & Miller 2002), but it provides a small margin of error.

7.1 Testing Procedures

The beamformers have each been tested using different types of data to determine their performance. Their performance is defined by how well they remove unwanted noise from the output of the beamformer.

Three different types of data have been used for the following tests:

- Single Frequency Tones
- Complex Signals
- Real Data

7.1.1 Single Frequency Tones

A single frequency tone, in the context of this project, is a signal of one frequency, and combinations of these tones are combined by the microphone array model to what would be expected from a real microphone array for a given scenario.

For testing using this data type, the results have been presented in graphs that display the relative magnitudes of all signal components arriving at the array against the varied parameter.

Two testing scenarios have been used. The first simulates narrowband signals and the second simulates broadband signals.

Scenario 1 - Narrowband signal test

In this scenario, the beamformer being tested, has been steered towards one of two signals, a 700Hz signal and a 900Hz signal, which are close enough together in the frequency spectrum to be considered narrowband. These signals have been placed at an angle of 60^o and -60^o from the perpendicular of the array respectively.

Scenario 2 - Broadband signal Test

In this scenario, the beamformer being tested, has been steered towards one of two signal groups and its performance has been determined for a range of different parameters. Each signal group comprises of a number of signals that cover a range large enough to be considered broadband. The first group of signals is comprised of a 330Hz signal, a 700Hz signal, a 1350Hz signal and a 2100Hz signal. The second group is comprised of a 410Hz signal, a 900Hz signal and a 2810Hz signal. These signal groups were placed at 60^o and -60^o from the perpendicular of the array respectively.

7.1.2 Complex signals

Complex signals, in the context of this project, refer to speech signals and other sources of broadband signals. These signals are combined using the microphone array model to what would be expected from a real array for a given scenario.

The Frost and GSC beamformers have been tested with this data type, to determine the performance of the beamformers, with different parameters. The results have been presented in graphs that display the original waveform the beamformer is directed at, the combination of all signals received by the array and the output of the beamformer.

Determining the relative success of this test has been assessed by comparing the resulting waveforms with both the original waveform and the waveform received at the array, as well as by listening to the results. Originally, it was to be evaluated by calculating the signal-to-noise ratio (SNR) of the output signal. However, because the beamformers employ filters as part of their operation, the output signal would not have been in phase with the original desired signal, making any calculation of SNR inaccurate.

Scenario 3 - Complex signal Test

In this test the beamformer has been steered towards one of three broadband signals, a female human speaker, a male human speaker and a television recording. These signals have been placed at an angle of 60^o, -60^o and 0^o from the perpendicular of the array respectively.

7.1.3 Real Data

Real Data refers to the data collected by the experimental microphone array that was described in the previous chapter. It was used to evaluate the beamformers to confirm the results seen using the simulated data. Three sets of real data were used to perform testing on the beamformers.

A more in-depth analysis using the recorded data to determine the effect of varying different parameters was intended, however, the poor quality of the recorded data has prevented this. Possible reasons for this poor quality have been discussed later in this chapter.

The results from testing using real data have been presented in sets of six plots. The first four plots present the frequency spectrum of the data that was received by the array. The fifth plot presents the frequency spectrum of the output of the beamformer and the sixth plot displays the waveform produced by the beamformer.

Scenario 4 - Real Data

For this scenario, two signals, 800Hz and 1000Hz, were directed at an array of four microphones from angles of -45^o and 45^o respectively. The spacing of the microphones in the array was 0.15m.

Scenario 5 - Real Data

For this scenario, two signals, 1400Hz and 1600Hz, were directed at an array of four microphones from angles of -60^o and 60^o respectively. The spacing of the microphones in the array was 0.1m.

Scenario 6 - Real Data

For the final test to be presented, two signals, 900Hz and 600Hz, were directed at an array of four microphones from angles of -45^o and 45^o respectively. The spacing of the microphones in the array was 0.1m.

7.2 Evaluation of Delay and Sum Beamformer

The Delay and Sum beamformer is the simplest form of beamforming. Therefore, many of the results from this section will apply to other beamformers that use uniform linear arrays. The Delay and Sum beamformer was tested by varying several of its parameters and observing the effect they had on the output of the beamformer. The parameters varied were:

- The number of microphones in the array.
- The effect of applying a Hamming window to the microphone outputs.
- The spacing between the microphones of the array.

Although the Delay and Sum beamformer has been established to be a narrowband beamformer, it has been tested with the simulated broadband signals to investigate the effects it has on beamforming on broadband signals.

7.2.1 Evaluation using Single Frequency Tones

Simulated Narrowband Signals

The first test performed on the Delay and Sum beamformer was used to determine the effect of changing the number of microphones in the array on how well it discriminates between two narrowband signals originating from different points in space. This test was performed using data from Scenario 1.

For this beamformer, the relative magnitudes of the signals after beamforming were found for different numbers of microphones in the array. The result, after the steering beamformer towards the 900Hz signal at 60^o, making the 700Hz signal noise, can be seen in Figure 7.1.

From the first test, it can be seen that as the number of microphones in the array increased, so did the beamformer's ability to cancel out the undesired signal. In this

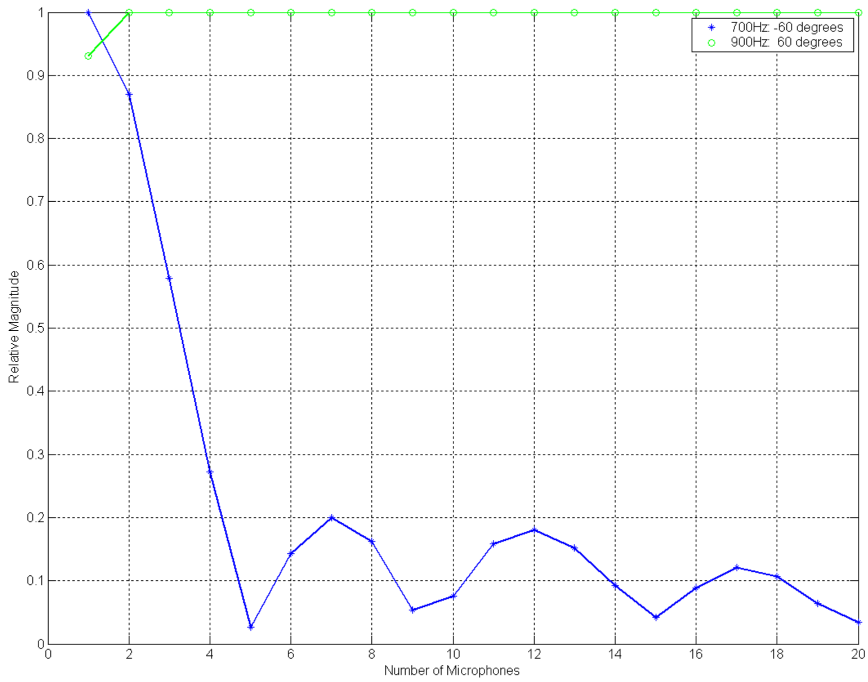

Figure 7.1: The effect of changing the number of microphones in a Delay and Sum beamformer on the relative magnitudes of narrowband signals arriving at the array.

test, the best performance was achieved with five microphones, after which the relative magnitude of the undesired signal oscillated as the number of microphones continued to increase. These oscillations were caused by ripples in the stopband of the beamformer, as discussed in chapter 4, and can be reduced by applying a Hamming window to the output of the microphones. Conducting the same test, but with a Hamming window applied to the microphones, gave the results presented in Figure 7.2.

Notice that with the Hamming window, the oscillation stopped, however, 12 microphones were then required to obtain the same noise cancellation as achieved previously. Notice also that there is a dip in the relative magnitude of the 700Hz signal before it begins to descend. The reason for this is that a Hamming window with two elements is effectively a square window and therefore has no effect on the output of the beamformer. When the third microphone was added, the Hamming window began to take its usual shape, dampening the effect of the third microphone, resulting in the performance becoming worse before getting better.

Figure 7.3 presents the results of the beamformer when it is steered towards the 700Hz

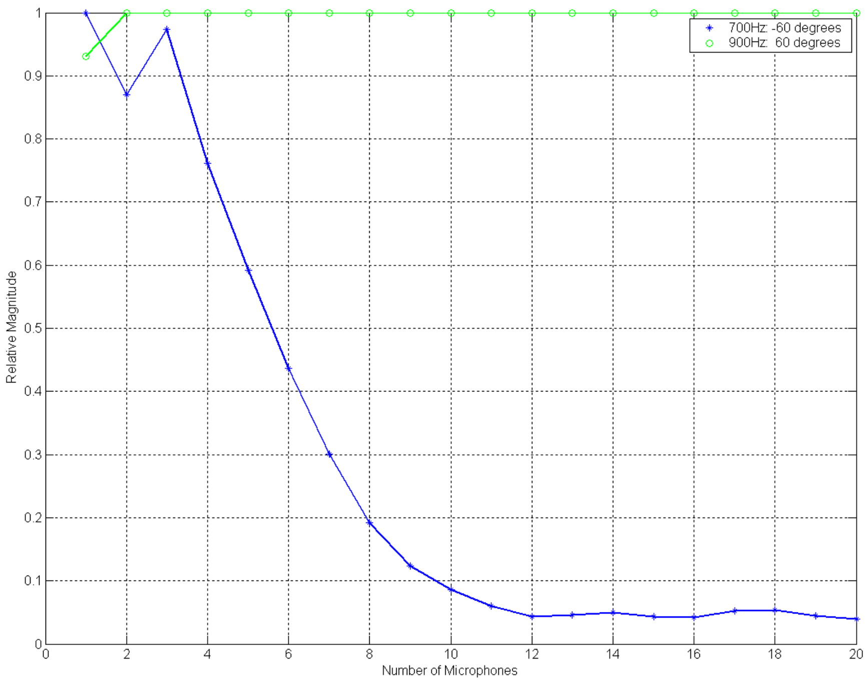

Figure 7.2: The effect of changing the number of microphones in a Delay and Sum beamformer and the use of a Hamming window on the microphones on the relative magnitudes of narrowband signals arriving at the array.

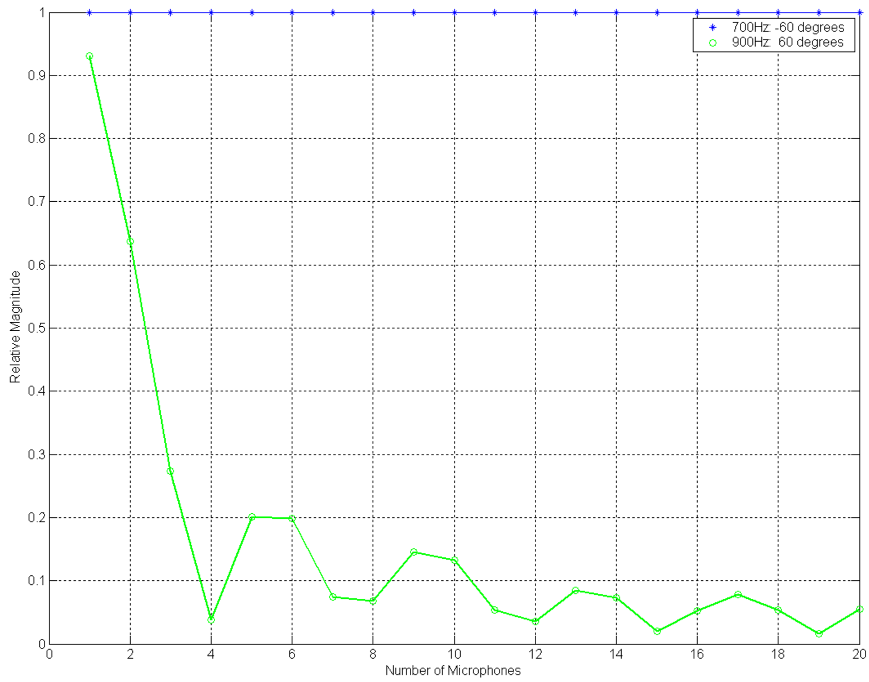

Figure 7.3: The effect of changing the number of microphones in a Delay and Sum beamformer on the relative magnitudes of narrowband signals arriving at the array.

signal, making the 900Hz signal noise. Figure 7.4 presents the same test but with a Hamming window applied to the inputs of the microphones. From this result it can be seen that, without a Hamming window, four microphones were needed to achieve the best performance. With a Hamming window, 10 microphones were needed.

Effect of Simulated Broadband Signals

The second test was performed to determine how well the beamformer attenuated and reinforced simulated broadband signals. This test was performed using Scenario 2. Steering the beamformer towards 60^o gave the result in Figure 7.5.

From this figure it can be seen that higher frequencies were attenuated with less microphones then lower frequencies. This is as expected because, as stated in chapter 4, the wider the array, the larger it is relative to the size of the impinging waves and therefore the more narrow the beam. This meant that for a low frequency, with a large wavelength, more array elements needed to be added to achieve an increase in the array's relative size, than a high frequency signal with a small wavelength. The

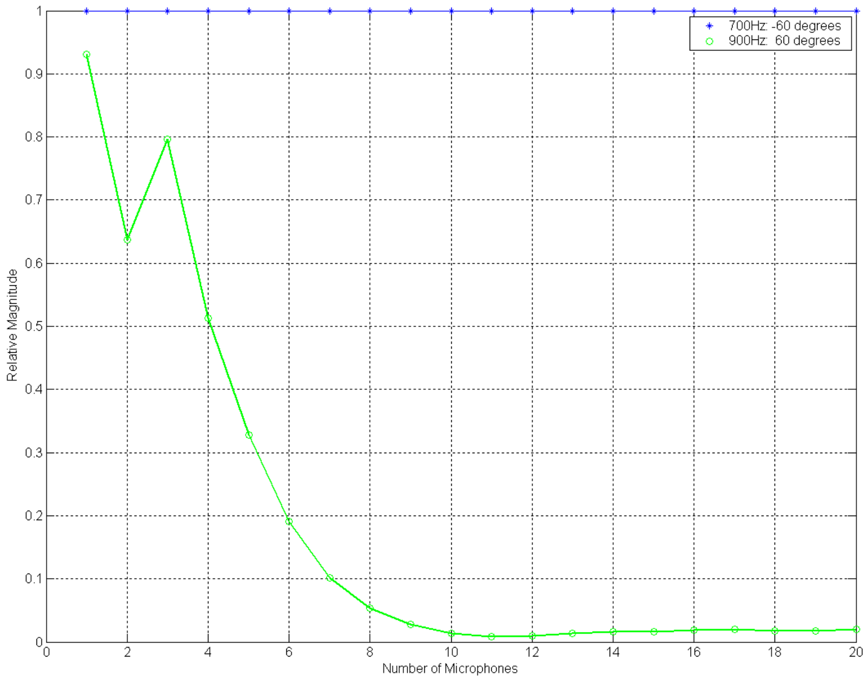

Figure 7.4: The effect of changing the number of microphones in a Delay and Sum beamformer and the use of a Hamming window on the microphones on the relative magnitudes of narrowband signals arriving at the array.

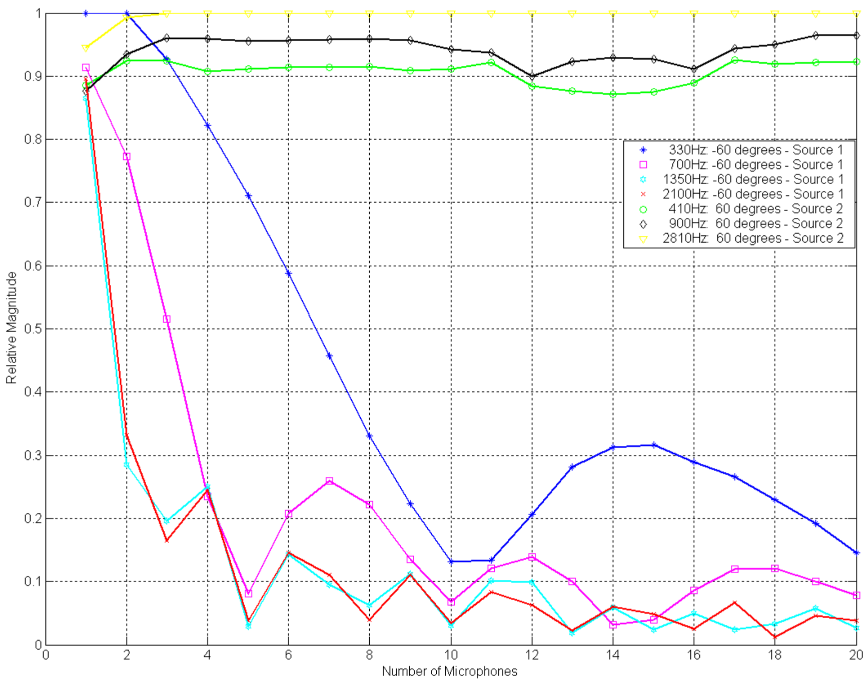

Figure 7.5: The effect of changing the number of microphones in a Delay and Sum beamformer on the relative magnitudes of broadband signals arriving at the array.

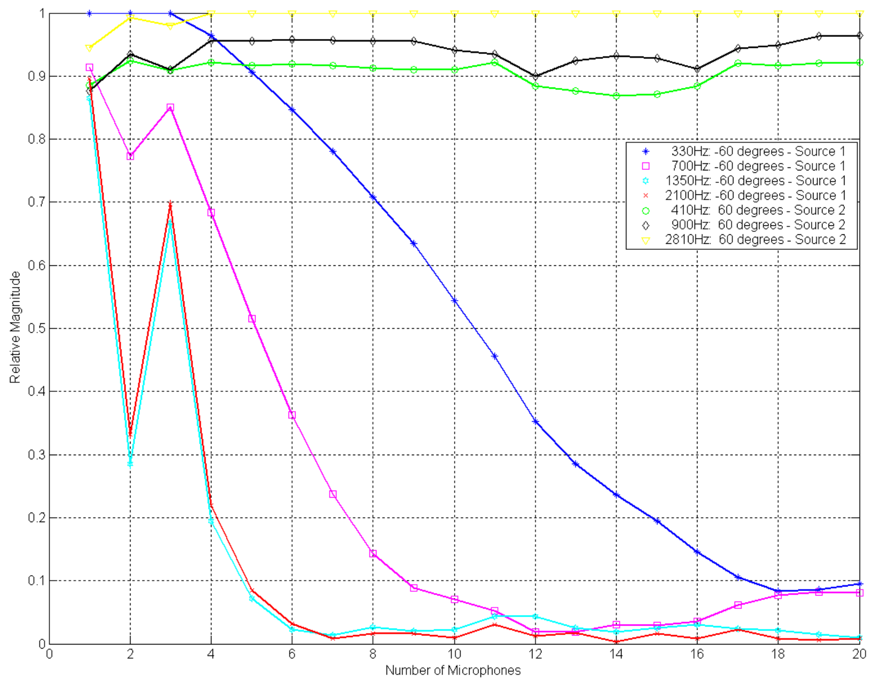

Figure 7.6: The effect of changing the number of microphones in a Delay and Sum beamformer and the use of a Hamming window on the inputs on the relative magnitudes of signals arriving at the array.

best performance was achieved with 10 microphones, after which oscillations were seen. After applying a Hamming window, the microphones again gave the result in Figure 7.6. As a result, the oscillations were reduced, however the best performance was not achieved until 18 microphones were used.

Steering the beamformer towards the other broadband signal gave the result in 7.7.

From this figure, it is interesting to note that in this test the 900Hz signal was attenuated with less microphones than the 2810Hz signal. The reason for this can be seen in Figure 7.8 which presents the response of a Delay and Sum beamformer. Here it can be see that in the high frequency ranges, at a large negative angle, the response was close to one. This was because of spatial aliasing. It was calculated in the previous chapter that the maximum frequency that could be received without spatial aliasing for an array with elements spaced at 0.055m was 3100Hz. This value, however, is the ideal value and the actual value is determined by the number of microphones in the array, which determines the accuracy of the beamformer's response. Figure 7.9 presents the

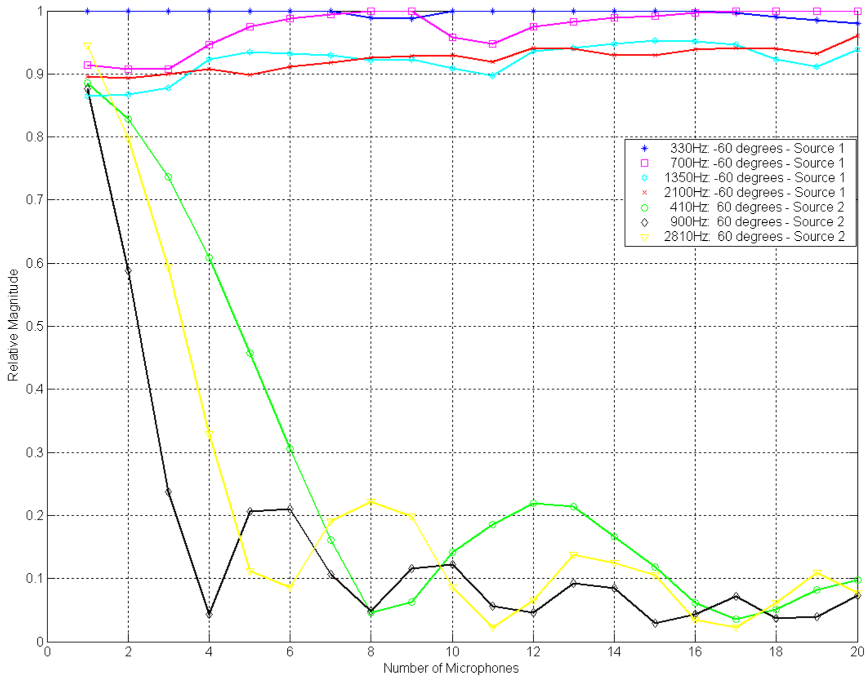

Figure 7.7: The effect of changing the number of microphones in a Delay and Sum beamformer on the relative magnitudes of broadband signals arriving at the array.

array response with 10 microphones and, as can be seen, the accuracy improved such that the aliasing could no longer be seen on the graph.

Applying a Hamming window to the microphone inputs while directed at -60^o gave the result in Figure 7.10. The best results were then achieved at 16 microphones.

Effect of Changing Distance

The effect of changing the distance between microphones was also tested for the Delay and Sum beamformer. This was done using the scenario 2. As discussed in chapter 4, the wider the array, the larger it is relative to the size of the impinging waves and therefore the more narrow or accurate the beam. This effect is shown in Figure 7.11 where an array of two microphones was steered towards the 900Hz signal.

Figure 7.12 presents the results from when the array was steered towards the 700Hz signal. Notice that the 900Hz signal was attenuated as the distance became larger, but then at 0.11m it began to get stronger. This was the result of spatial aliasing again.

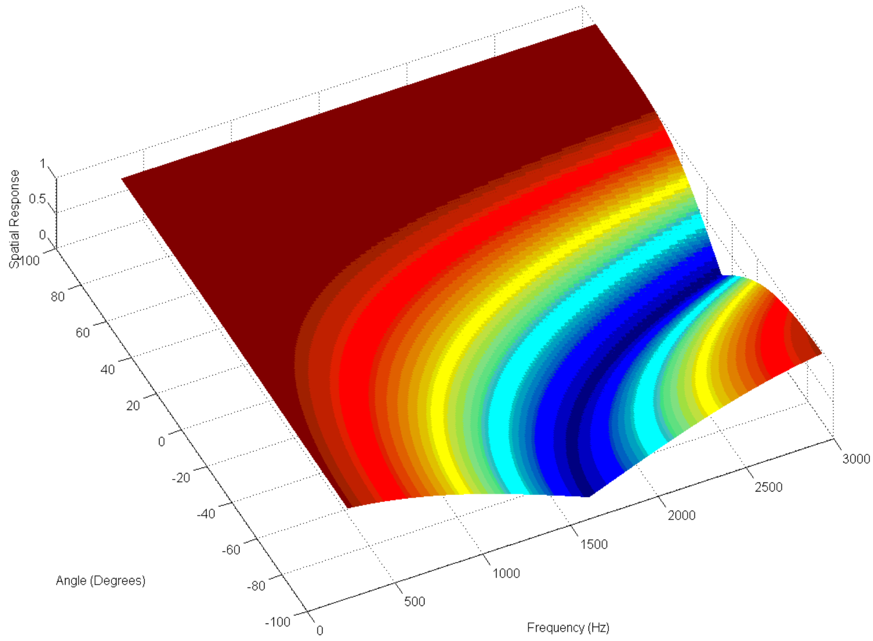

Figure 7.8: Response of a Delay and Sum beamformer with two microphones, a spacing of 0.055m and a look direction of 60^o.

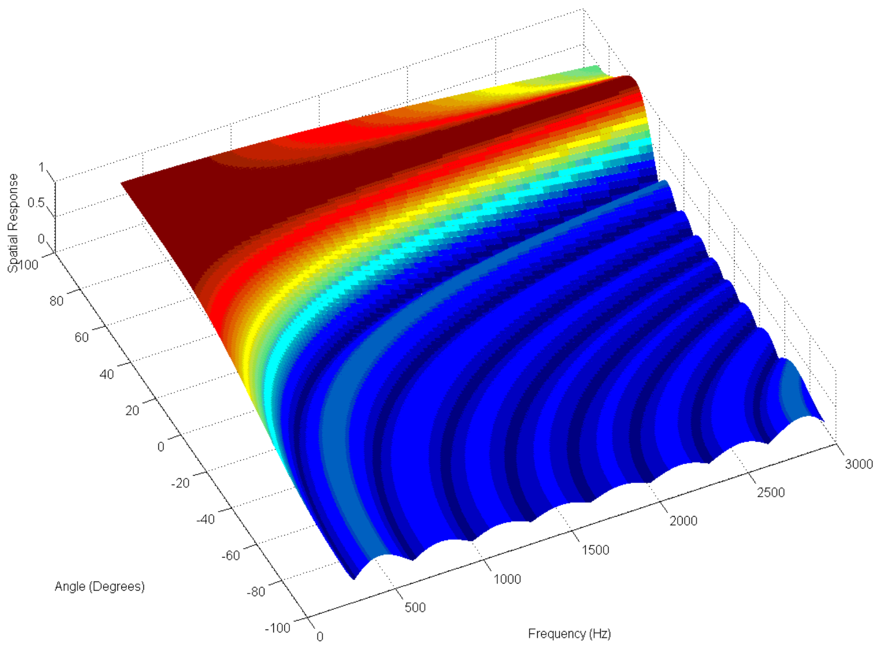

Figure 7.9: Response of a Delay and Sum beamformer with 10 microphones, a spacing of 0.055m and a look direction of 60^o.

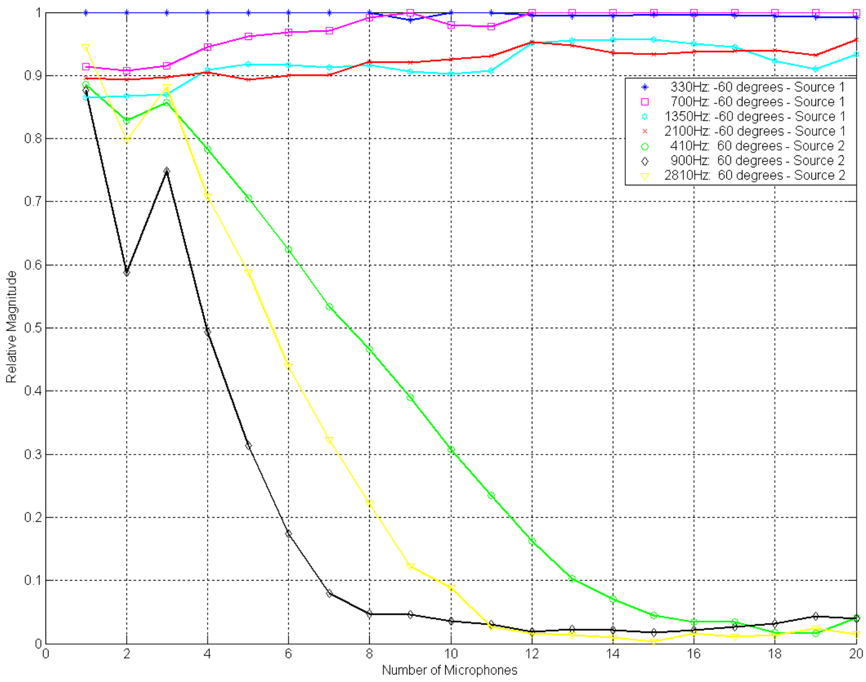

Figure 7.10: The effect of changing the number of microphones in a Delay and Sum beamformer and the use of a Hamming window on the inputs on the relative magnitudes of broadband signals arriving at the array

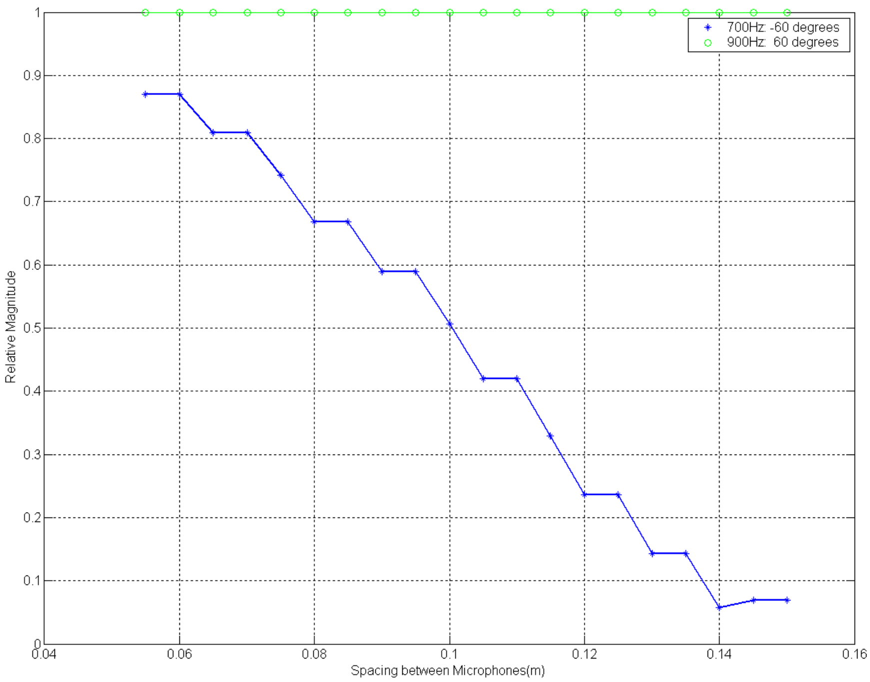

Figure 7.11: The effect of changing the spacing between microphones on a Delay and Sum beamformer in an array on the output.

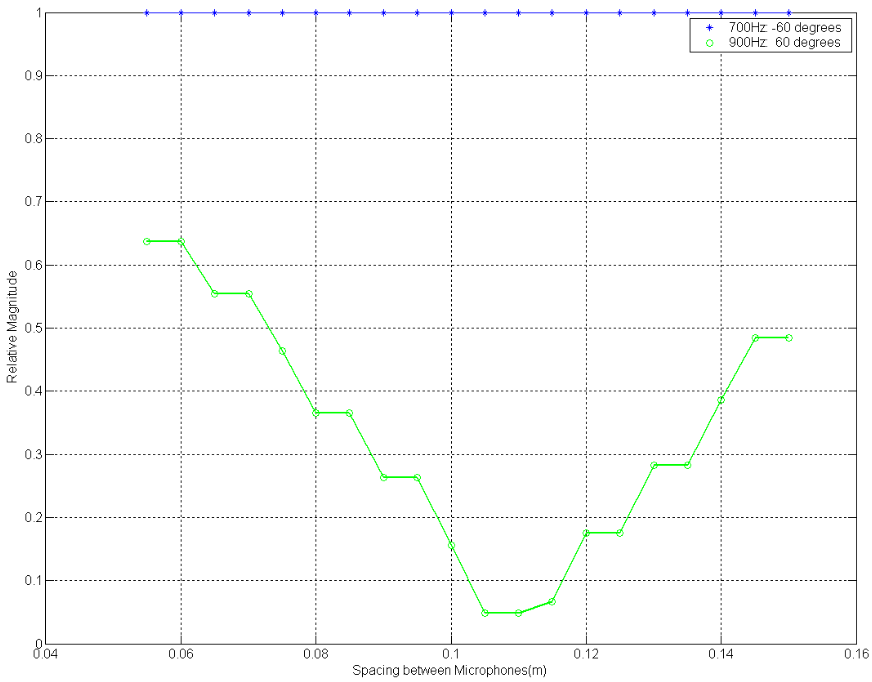

Figure 7.12: The effect of changing the spacing between microphones on a Delay and Sum beamformer in an array on the output.

As previously discussed, as the spacing between the array gets larger, the maximum frequency that the array can process without spatial aliasing decreases and, at a spacing of 0.15m, the limit is 1150Hz. The 900Hz signal was well under this limit, however, as pointed out previously, it is the ideal limit, with the actual limit dependent on the number of microphones.

It is important to point out that the last test was conducted using only two microphones so as to illustrate the effect of changing distance. Had more microphones been used to conduct this test, no spatial aliasing would have occurred but the effect of changing the distance would not have revealed what was happening as effectively due to the improved accuracy of the main response beam.

7.2.2 Evaluation using Real Data

A series of tests were also conducted using data recorded from the experimental microphone array. Following are a sample of the results obtained.

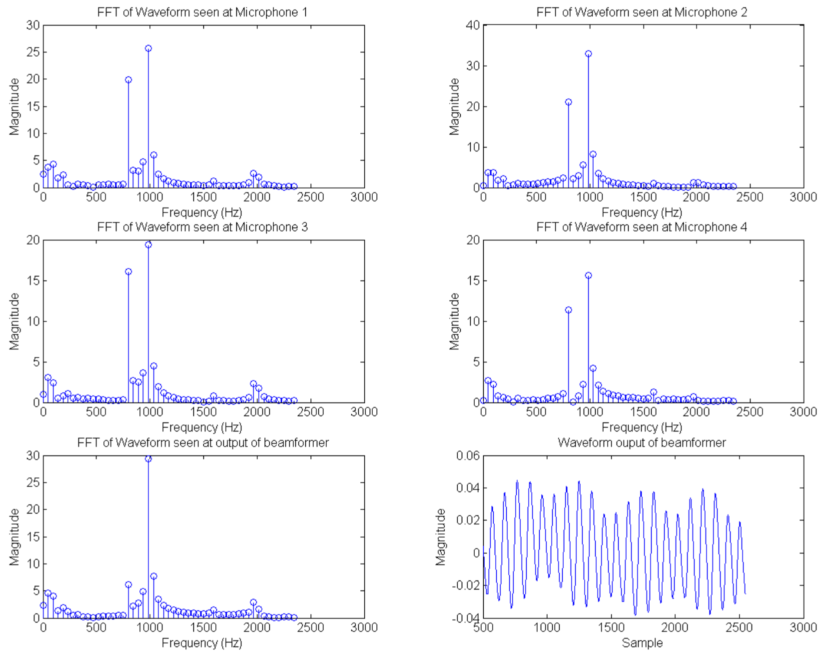

Figure 7.13: Result of Delay and Sum beamformer using real data. Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, spacing = 0.15m and look direction = 45^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

The first test was conducted using Scenario 4 and steering the beamformer towards the 1000Hz signal at 45^o gave the result in Figure 7.13. Steering the beamformer towards the 800Hz signal at -45^o gave the result in Figure 7.14.

Notice that, although in both cases the desired signal is reinforced, the result obtained when the beamformer was steered towards the 1000Hz signal (45^o) resulted in a greater reduction of the other "noise" signal. This is because, as shown in the figures, each signal was not recorded with equal strength.

The second test was conducted using Scenario 5. Steering the beamformer towards the 1600Hz signal at 60^o gave the result in Figure 7.15 and steering the beamformer towards the 1400Hz signal at -60^o gave the result in Figure 7.16.

Notice that, again, the two signals were not recorded with equal strength by the microphones, resulting in the beamformer not performing when equally pointing at the two different signals.

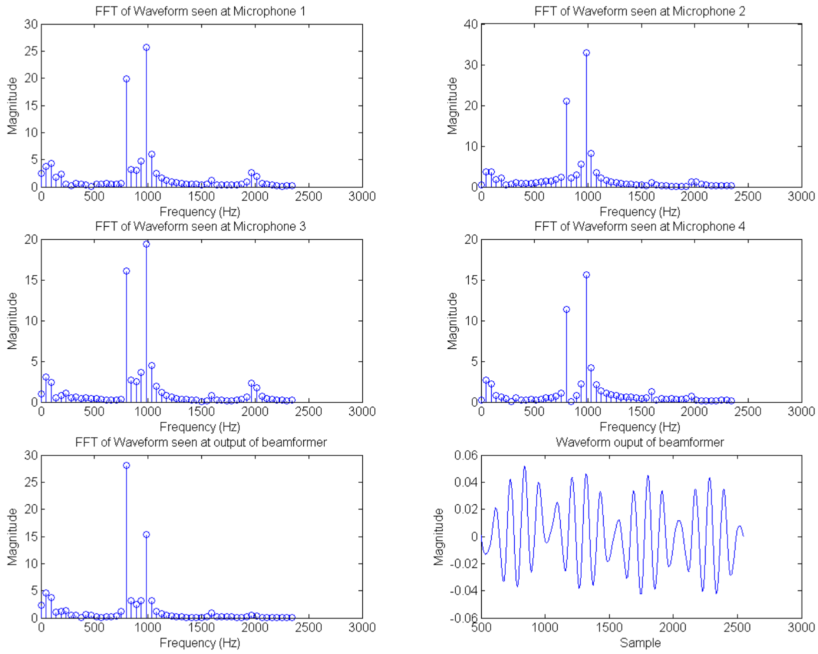

Figure 7.14: Result of Delay and Sum beamformer using real data. Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, spacing = 0.15m and look direction = -45^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

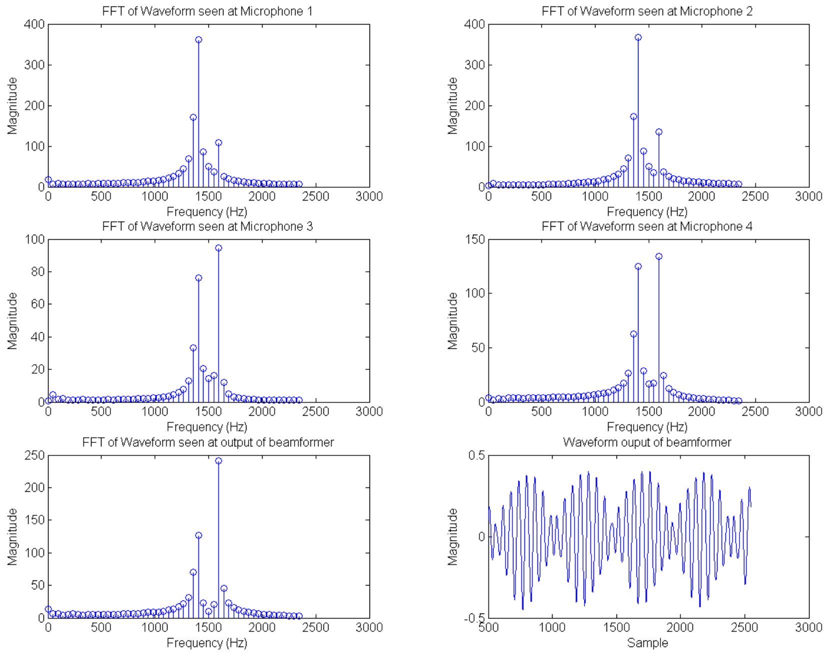

Figure 7.15: Result of Delay and Sum beamformer using real data. Source Signals = [1400Hz, 1600Hz], Angles = [-60^o, 60^o], number of microphones = 4, spacing = 0.1m, look direction = 60^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

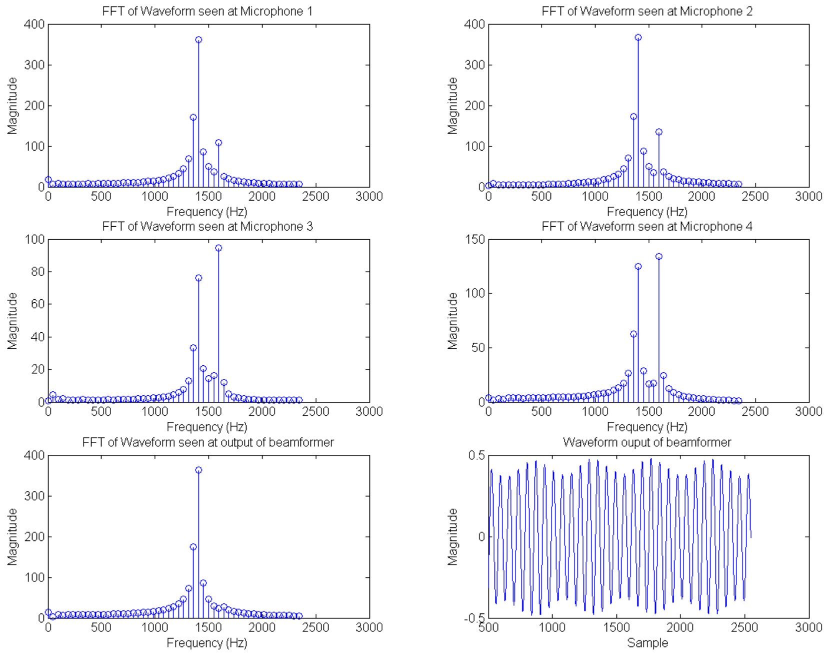

Figure 7.16: Result of Delay and Sum beamformer using real data. Source signals = [1400Hz, 1600Hz], angles to sources = [-60^o, 60^o], number of microphones = 4, Spacing = 0.1m and look direction = -60^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

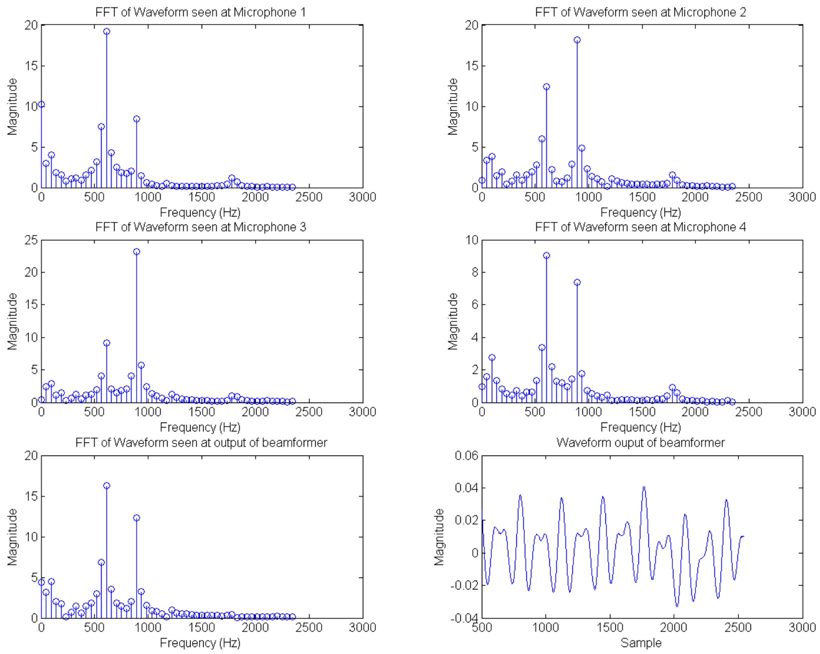

Figure 7.17: Result of Delay and Sum beamformer using real data. Source Signals = $[900Hz,\ 600Hz]$, Angles = $[-45^o,\ 45^o]$, number of microphones = 4, Spacing = 0.1m, look direction = 45^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

The final test was conducted using Scenario 6. Steering the beamformer towards the 600Hz signal at 45^o gave the result in Figure 7.17 and steering the beamformer towards the 900Hz signal at -45^o gave the result in Figure 7.18.

Notice that, again, the two signals were not recorded with equal strength by the microphones, resulting in the beamformer not performing equally pointing at the two different signals.

7.2.3 Discussion

It has been concluded from the results that, as the number of microphones in the array increased, the beamformers ability to reject unwanted noise also increased. This was as expected, due to the fact that more microphones cause greater desired signal reinforcement in the desired direction and more out-of-phase signals from undesired directions to cause destructive interference. The results also demonstrate why the Delay

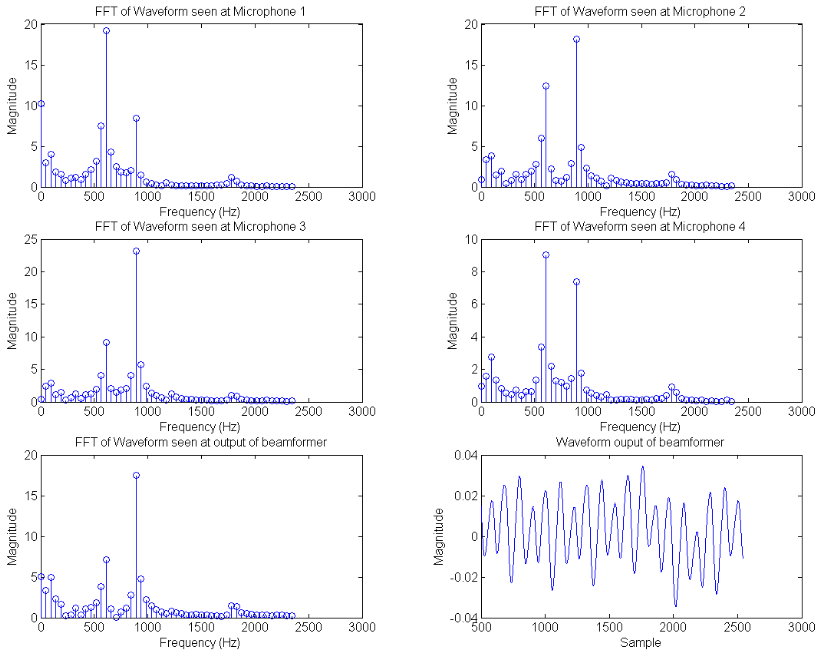

Figure 7.18: Result of Delay and Sum beamformer using real data. Source Signals = $[900Hz,\ 600Hz]$, Angles = $[-45^o,\ 45^o]$, number of microphones = 4, Spacing = 0.1m, look direction = -45^o. Notice that although the desired signal has been reinforced, the microphones did not record each signal with the same strength.

and Sum beamformer is a narrowband beamformer but can approach the performance of a broadband beamformer when the number of microphone elements is increased.

Best performance was achieved when:

- using 5 microphones for two narrowband signals and 12 for narrowband signals with a Hamming window.
- using 10 microphones for broadband signals and 18 for broadband signals with a Hamming window.
- using a large spacing between microphones. This, however, as demonstrated by the results, places limitations on the maximum frequencies that can be attenuated without spatial filtering occurring.

The results from the real data demonstrated that the Delay and Sum beamformer does reinforce signals in the look direction as well as partially cancelling other, undesired, signals. However, the performance does not match that of the simulated data. Although this would be expected to a degree, a considerable problem arose. Looking at the different frequency spectrums of the signals recorded by the microphones in the array, it can be seen that the magnitudes of the received signals varied considerably across the microphones in the array and as a result some signals reinforced better than others. These differences in magnitudes are most likely the result of errors induced in recordings, such as differences in the microphones and other equipment used in the recording process. Another likely source of error was the positioning of the microphones and the sources. If the distances between the microphones are not what are expected or, there are steering errors, the time delays calculated by the beamformers will become inaccurate, resulting in signals from the desired direction not being aligned.

These results illustrate that the Delay and Sum beamformer would not be the ideal choice for use in security applications. Although it is very simple to implement, a large number of microphones are required before significant interference rejection is achieved across a wide frequency range.

7.3 Evaluation of the Frost Beamformer

The Frost beamformer was tested by varying several parameters and determining the effect on the output of the beamformer. The parameters that were varied are:

- The length of the tapped-delay line.
- The number of microphones in the array.

The beamformer was also evaluated using data recorded from the experimental microphone array.

7.3.1 Evaluation using Tones

Simulated Narrowband Signals

The first set tests performed on the Frost beamformer were used to determine the effect of changing the length of the tapped-delay lines, in conjunction with changing the number of microphones in the array, on how well it discriminates between two narrowband signals. In these tests, the relative magnitudes of the signals after beamforming were then found for different numbers of microphones in the array and different tapped-delay line lengths. These tests were conducted using Scenario 1.

Steering the beamformer towards the 900Hz at 60^o, thereby making the 700Hz signal noise, produced the result seen in Figure 7.19. Steering the beamformer towards the other signal, 700Hz, at -60^o gave the result in Figure 7.20.

From these test results, it can be concluded that, as the length of the tapped-delay line increased so did the beamformer's ability to attenuate undesired narrowband signals. These tests also showed that the more microphones that were present in the array, the less effect increasing the length of the tapped-delay line had.

Simulated Broadband Signals

The broadband response was tested using Scenario 2. Steering the beamformer towards

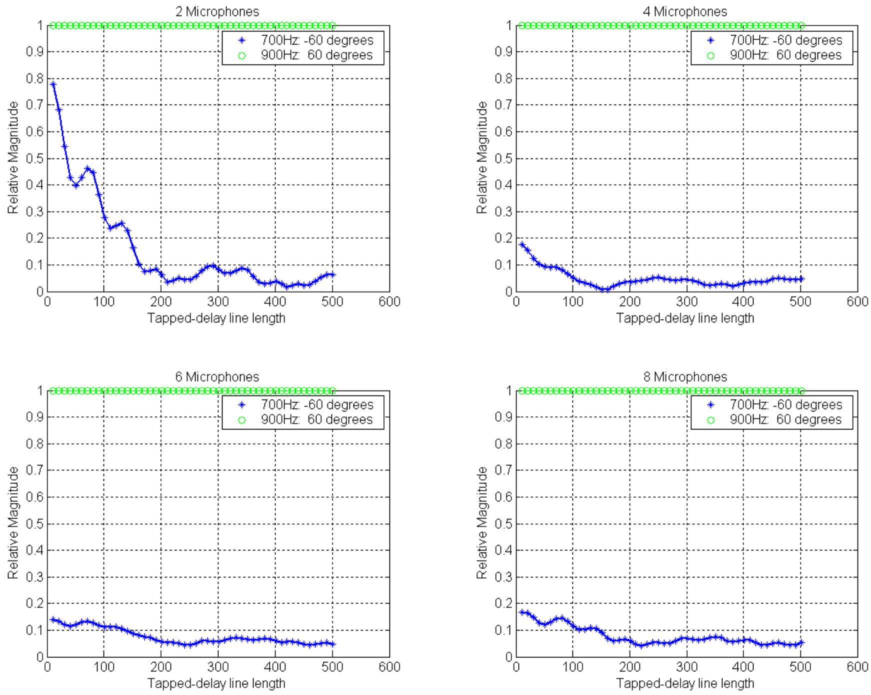

Figure 7.19: The effect of changing the length of the tapped-delay line for different microphone array sizes on the relative magnitude of the output of the Frost beamformer.

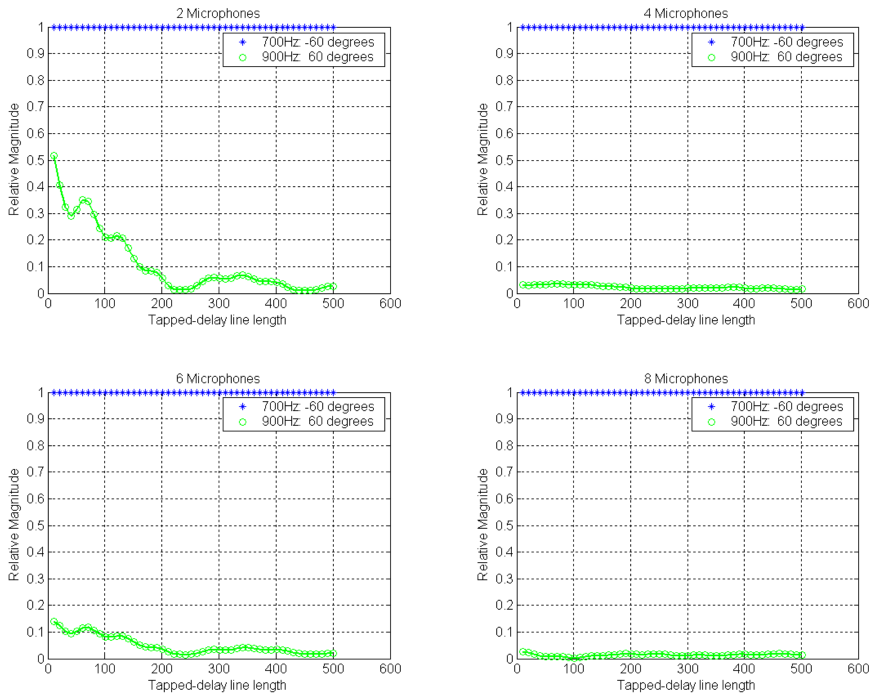

Figure 7.20: The effect of changing the length of the tapped-delay line for different microphone sizes on the relative magnitude of the output of the Frost beamformer.

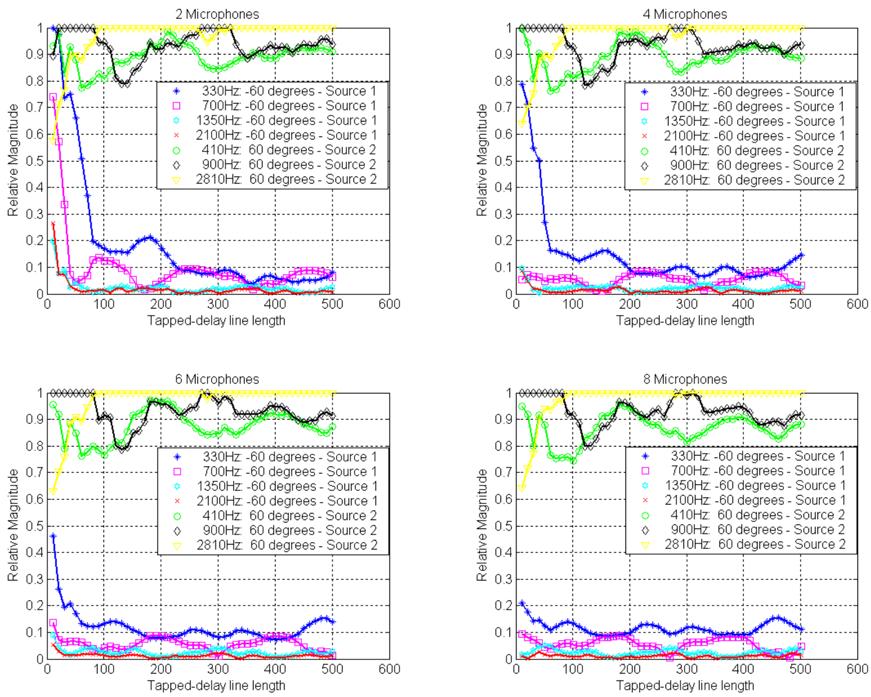

Figure 7.21: The effect of changing the length of the tapped-delay line for different microphone sizes on the relative magnitude of the output of the Frost beamformer.

60^o gave the result in Figure 7.21 and steering the beamformer towards the other signal source gave the results presented in Figure 7.22.

These two tests showed that, for all array sizes, as the length of the tapped-delay line increased, so did the ability of the beamformer to attenuate the undesired noise signals. However, the more microphones present in the array, the shorter the tapped-delay line needed to be to achieve the same performance. The point at which increasing the tapped-delay line length no longer has an effect is summarised below (Note: this point has been defined as the first time all noise signals pass below a relative magnitude of 0.1):

- For two microphones, a tapped-delay length of 231 was required.
- For four microphones, a tapped-delay length of 201 was required.
- For six microphones, a tapped-delay length of 181 was required.
- For eight microphones, a tapped-delay length of 131 was required.

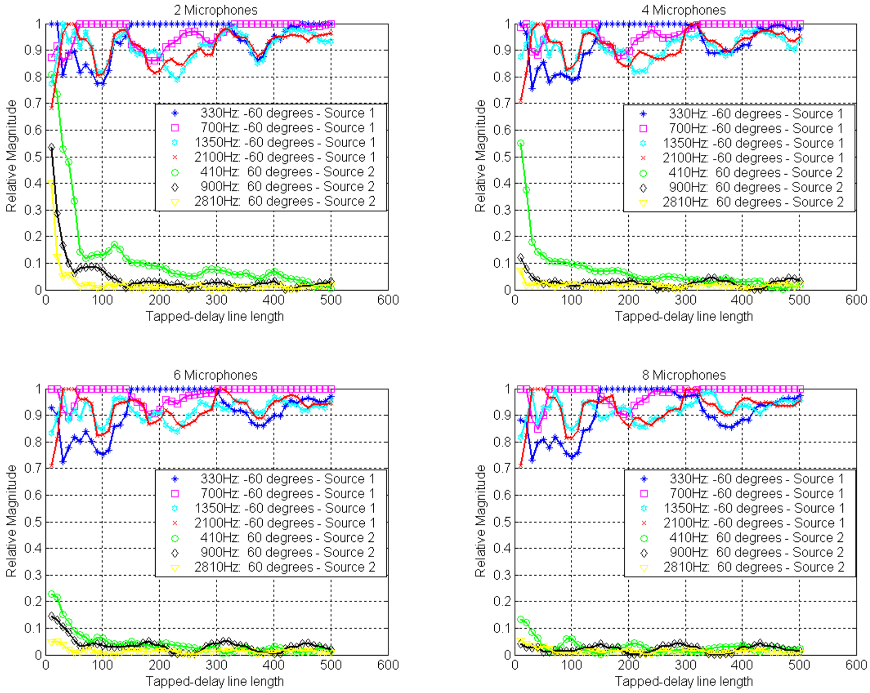

Figure 7.22: The effect of changing the length of the tapped-delay line for different microphone sizes on the relative magnitude of the output of the Frost beamformer.

7.3.2 Evaluation using Complex Signals

The Frost beamformer was also evaluated using complex signals. Scenario 3 was used to conduct this test and in each of the following tests the beamformer was steered towards the male speaker at -60^o.

The results from the previous section have been used to determine tapped-delay lengths for which to test the beamformer. To assess whether the performance would improve, for each size array, a tapped-delay line of 501 was also used. The results for:

- two microphones with tapped-delay line lengths of 231 and 501 have been presented in Figures 7.23 and 7.24 respectively;
- four microphones with tapped-delay line lengths of 201 and 501 have been presented in Figures 7.25 and 7.26 respectively;
- six microphones with tapped-delay line lengths of 181 and 501 have been presented in Figures 7.27 and 7.28 respectively; and

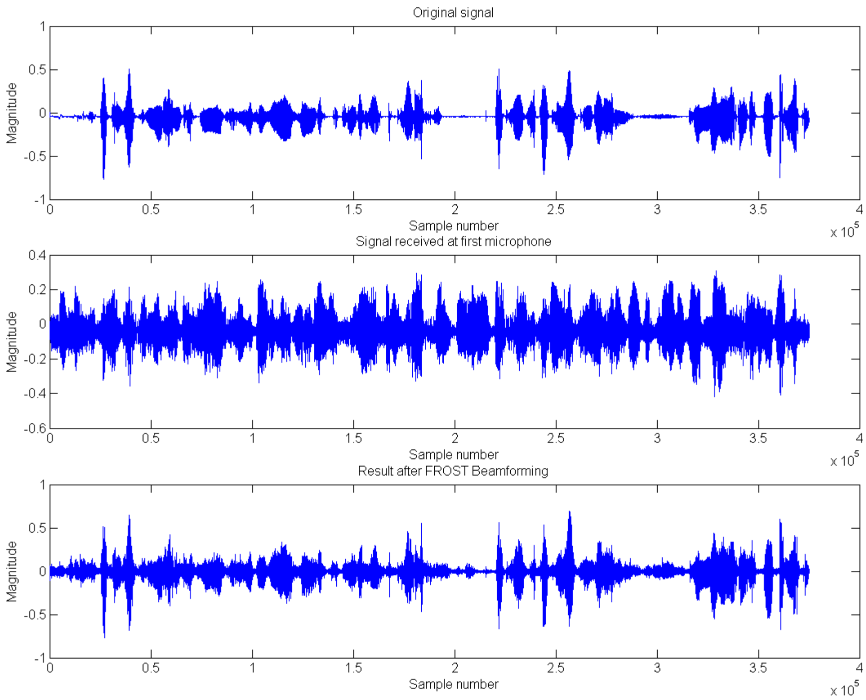

Figure 7.23: Result of Frost beamforming on complex signals. This beamformer employed two microphones and a tapped-delay length of 231

- eight microphones with tapped-delay line lengths of 131 and 501 have been presented in Figures 7.29 and 7.30 respectively.

For each array size, there has been an apparent reduction in noise in the output of the beamformer. This can be easily seen by comparing the original speech with the output signal. For each array size, there appears to be little difference between the determined optimum tapped-delay length and a tapped-delay length of 501. An assessment by listening to the outputs of the beamformer for each case backup both these results. In addition, by listening to the results it is apparent that as the number of microphones increased, the amount of noise present has decreased. As a result, the highest qualty achieved by the beamformer was with eight microphones and a tapped-delay line of 131.

When the beamformer was steered to the female speaker at 60^o a similar result was obtained. Appendix B.2 details all audio files produced as the result of these tests.

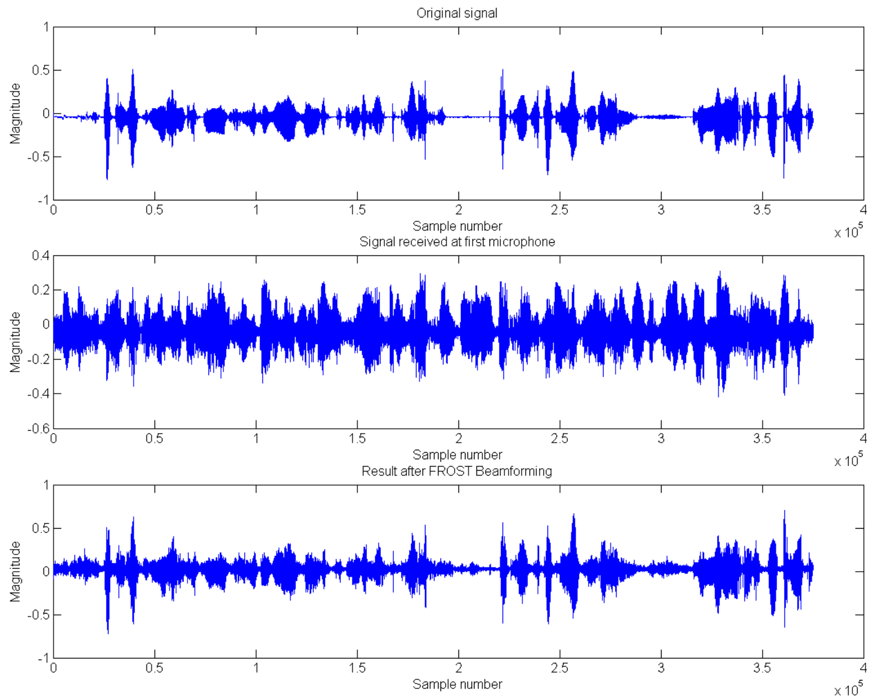

Figure 7.24: Result of Frost beamforming on complex signals. This beamformer employed two microphones and a tapped-delay length of 501

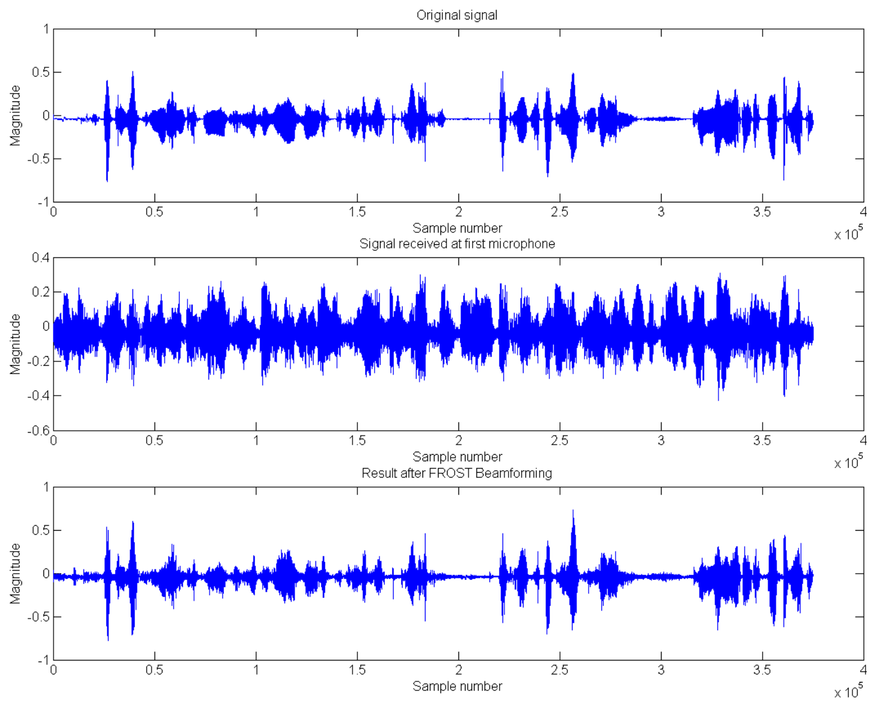

Figure 7.25: Result of Frost beamforming on complex signals. This beamformer employed four microphones and a tapped-delay length of 201.

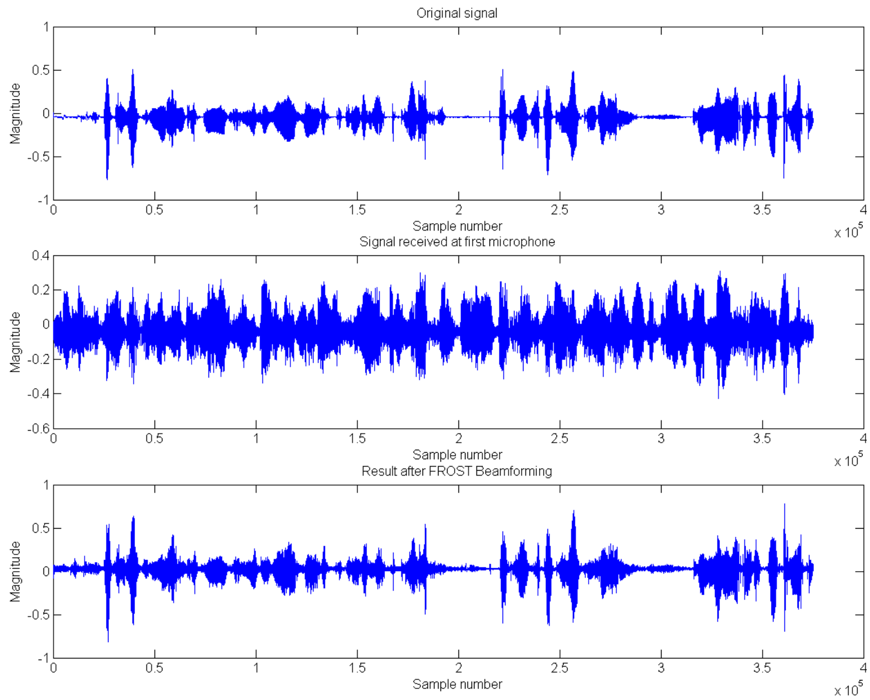

Figure 7.26: Result of Frost beamforming on complex signals. This beamformer employed four microphones and a tapped-delay length of 501.

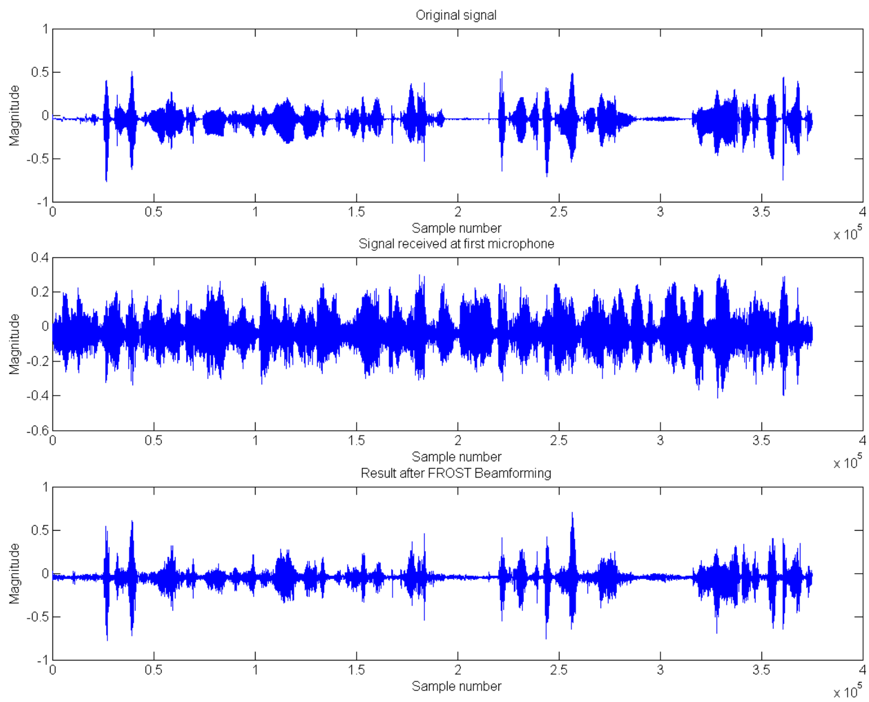

Figure 7.27: Result of Frost beamforming on complex signals. This beamformer employed six microphones and a tapped-delay length of 181.

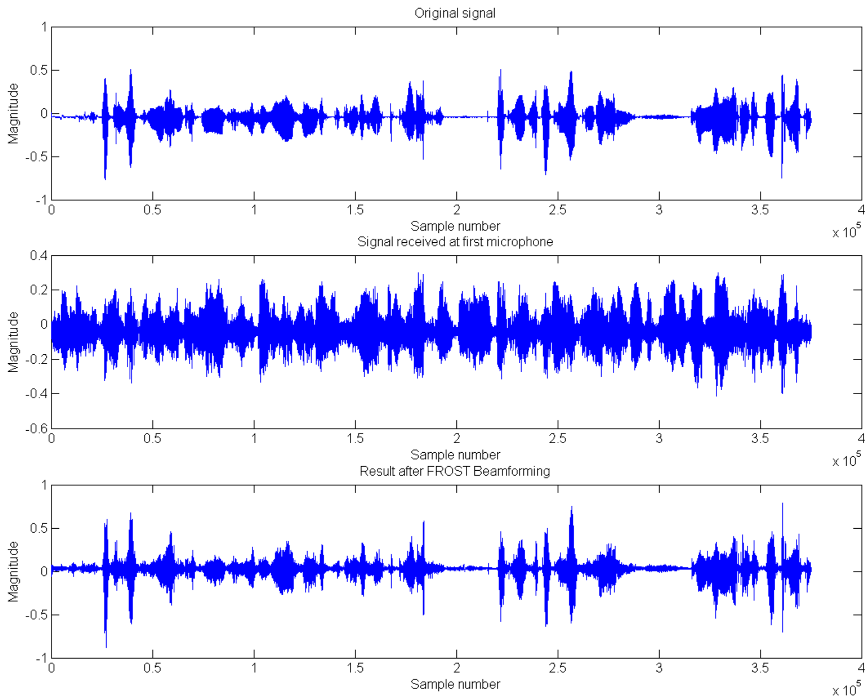

Figure 7.28: Result of Frost beamforming on complex signals. This beamformer employed six microphones and a tapped-delay length of 501.

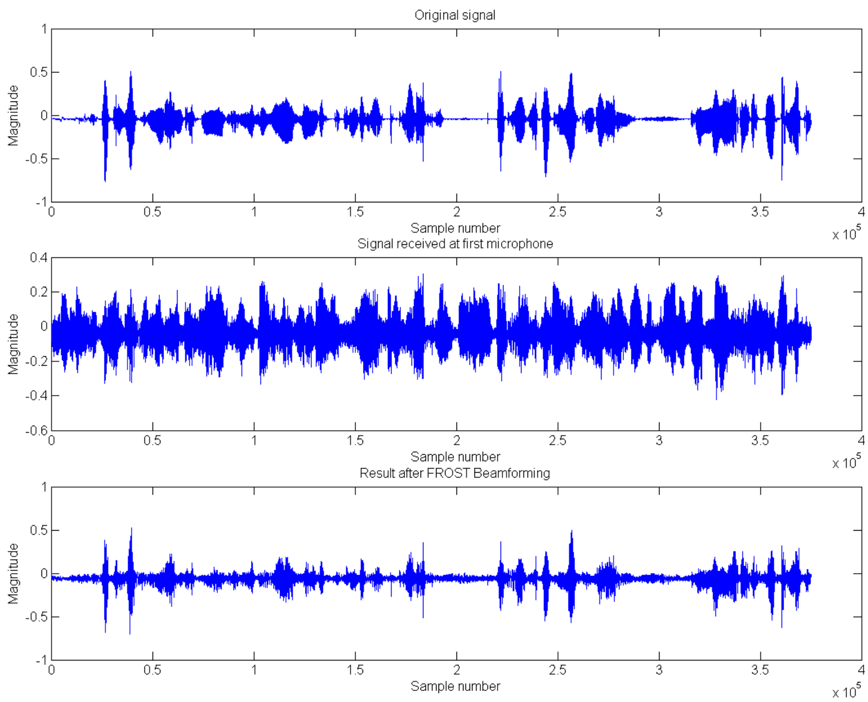

Figure 7.29: Result of Frost beamforming on complex signals. This beamformer employed eight microphones and a tapped-delay length of 131.

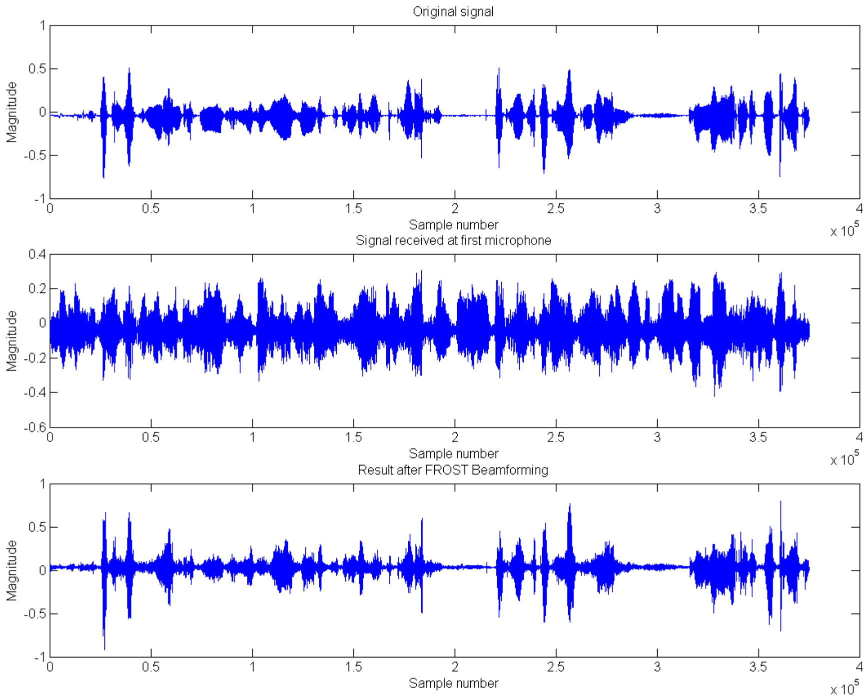

Figure 7.30: Result of Frost beamforming on complex signals. This beamformer employed eight microphones and a tapped-delay length of 501.

7.3.3 Evaluation using Real Data

A series of tests were also conducted using data recorded from the experimental microphone array. Although some positive results were obtained using the Delay and Sum method, the Frost beamformer did not produce any positive results. Figures 7.31, 7.32 where obtained using data from Scenario 4 and Figures 7.33 and 7.34 where obtained using data from Scenario 5. Notice that the beamforming cancelled out the desired signals and spurious noise was introduced. As mentioned in the broadband beamformers chapter, the Frost beamformer is very susceptible to steering errors and these errors result in signal cancellation as can be seen the figures. These steering errors are most likely the result of the calculation of incorrect angles to the sources and errors in the placing of the microphones.

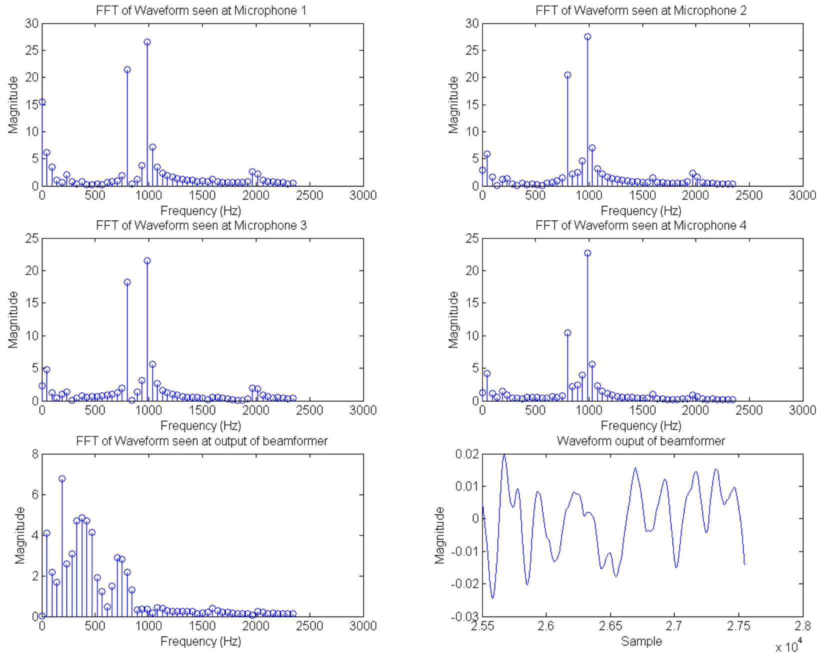

Figure 7.31: Result of Frost beamforming: Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, tapped-delay length = 501, spacing = 0.15m and look direction = -45^o. Notice that both signals have been cancelled.

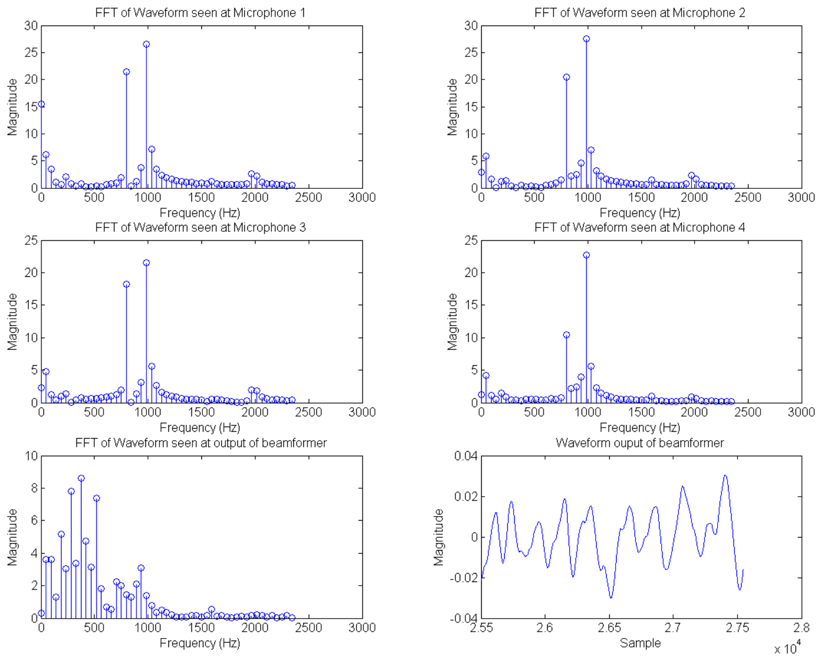

Figure 7.32: Result of Frost beamforming: Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, tapped-delay length = 501, spacing = 0.15m and look direction = 45^o. Notice that both signals have been cancelled.

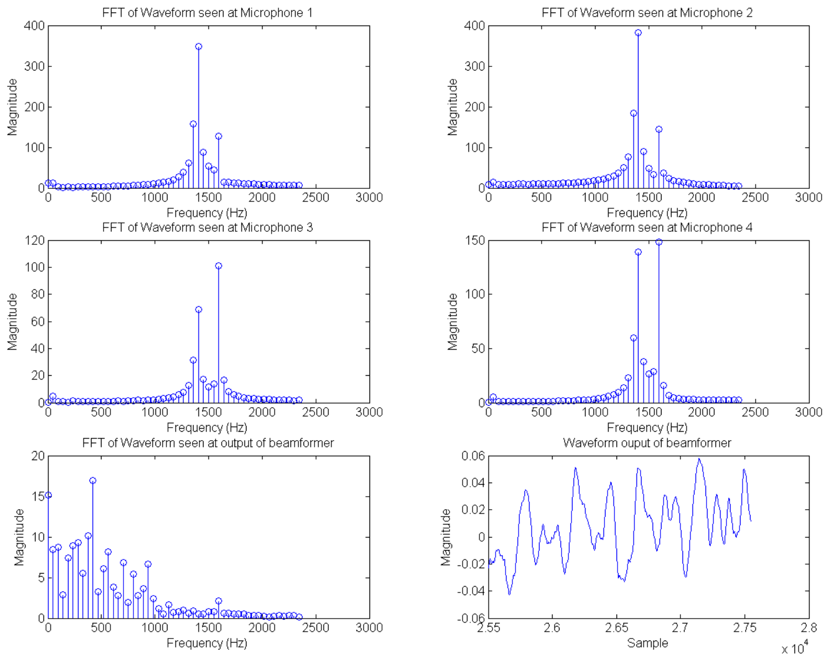

Figure 7.33: Result of Frost beamforming: Source signals = [1400Hz, 1600Hz], angles to sources = [-60^o, 60^o], number of microphones = 4, tapped-delay length = 501, spacing = 0.10m and look direction = -60^o. Notice that both signals have been cancelled.

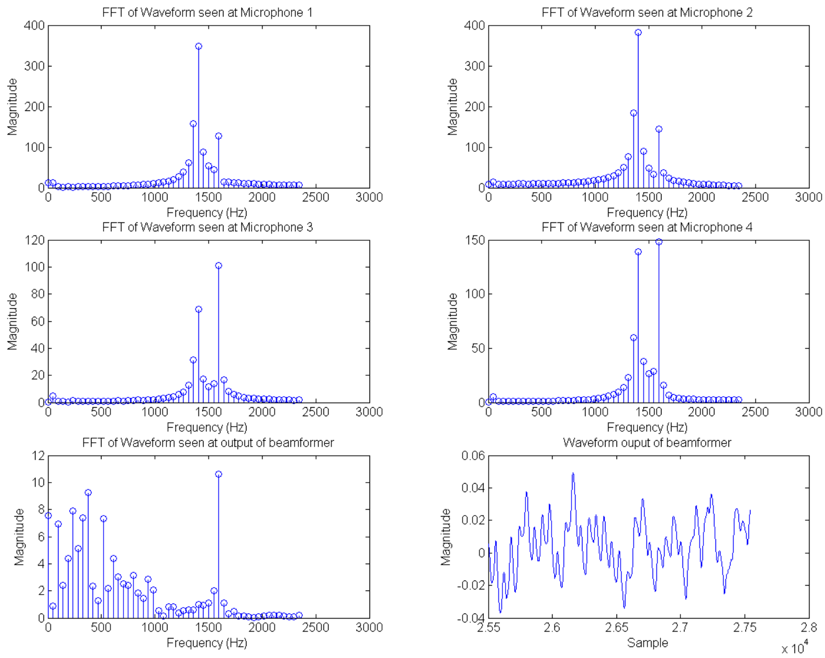

Figure 7.34: Result of Frost beamforming: Source signals = [1400Hz, 1600Hz], angles to sources = [-60^o, 60^o], number of microphones = 4, tapped-delay length = 501, spacing = 0.10m and look direction = 60^o. Notice that both signals have been cancelled.

7.3.4 Discussion

From the simulations, it has been concluded that as the length of the tapped-delay lines increased, so did the performance of the beamformer - to a point, after which the only way to increase the performance of the beamformer was to increase the number of microphones in the array. This result is supported by Liu, Wu & Langley (2006).

Using the simulated signal tone broadband signals, a range of optimum tapped-delay line lengths were found. These lengths were then used to evaluate the complex data and it was found that increasing the length of the delays past what was found produced no apparent improvement in the outputs. The results have been summarised in table 7.1.

Number of Microphones	**Optimum Tapped-Delay line length**
2	231
4	201
6	181
8	131

Table 7.1: Summary of results for Frost beamformer using simple tones.

The best overall audible performance was achieved with eight microphones with a tapped-delay length of 131.

The Frost beamformer, however, performed very poorly when real data was used. As mentioned earlier, this was mostly likely the result of steering errors produced by inaccurate angles to the source signals and errors in the placing of the microphones in the array.

Although the beamformer performed very poorly on the data from the experimental microphone array, it performed very well in the simulations and, therefore, it has been concluded that the Frost beamformer is a candidate for use in security applications, particularly because only a small number of microphones are required to achieve a reasonable quality signal, requiring only a small microphone array.

7.4 Evaluation of the Generalized Sidelobe Canceller Beamformer

The GSC beamformer has been tested by varying several parameters and determining the effect on the output of the beamformer. These tests have been performed using single frequency tones and complex signals. The parameters that have been varied are:

- The length of the tapped-delay line.
- The number of microphones in the array.

The beamformer will also be evaluated using data recorded from the experimental microphone array.

7.4.1 Evaluation using Tones

Simulated Narrowband Signals

The first test performed on the Frost beamformer was used to determine the effect of changing the length of the tapped-delay line, in conjunction with changing the number of microphones in the array, on how well it discriminates between two narrowband signals originating from different points in space.

For this test, Scenario 1 was used. The relative magnitudes of the signals, after beamforming, were then found for different numbers of microphones in the array and different tapped-delay line lengths. The result, after steering the beamformer towards the 900Hz at 60^o, thereby making the 700Hz signal noise, can be seen in Figure 7.35. Steering the beamformer towards the other signal (700Hz) at -60^o gave the result in Figure 7.35.

The results for every size microphone array and for all tested tapped-delay lengths were very high. The reason the beamformer has performed so well is that the blocking matrix successfully blocked the desired signal from its lower processing lower path, leaving only the noise signal which was adapted through the multiple-input canceller and then subtracted from the fixed beamformer.

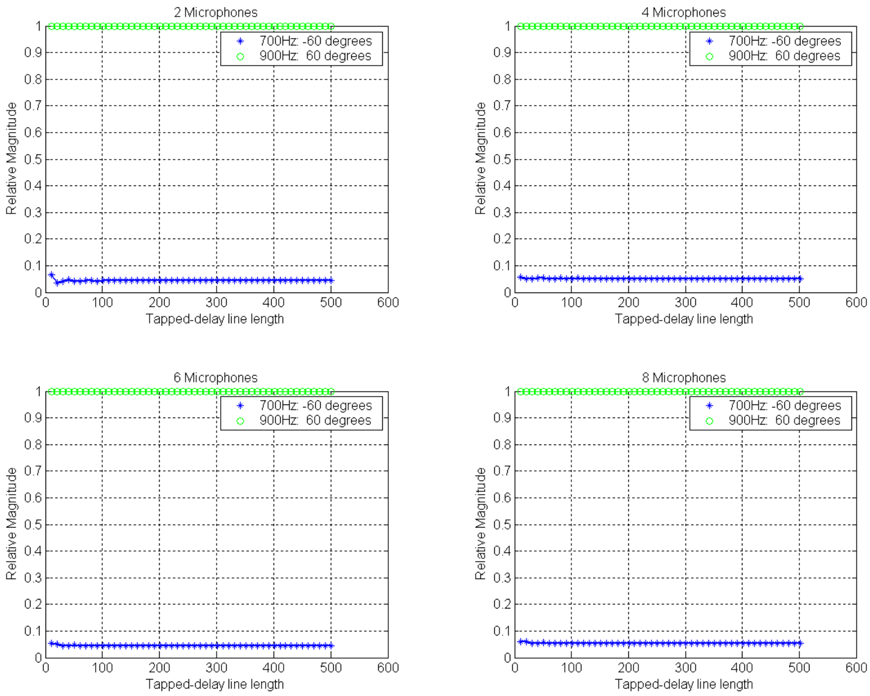

Figure 7.35: The effect of changing the length of the tapped-delay line for different microphone array sizes on the relative magnitude of the output of the GSC beamformer.

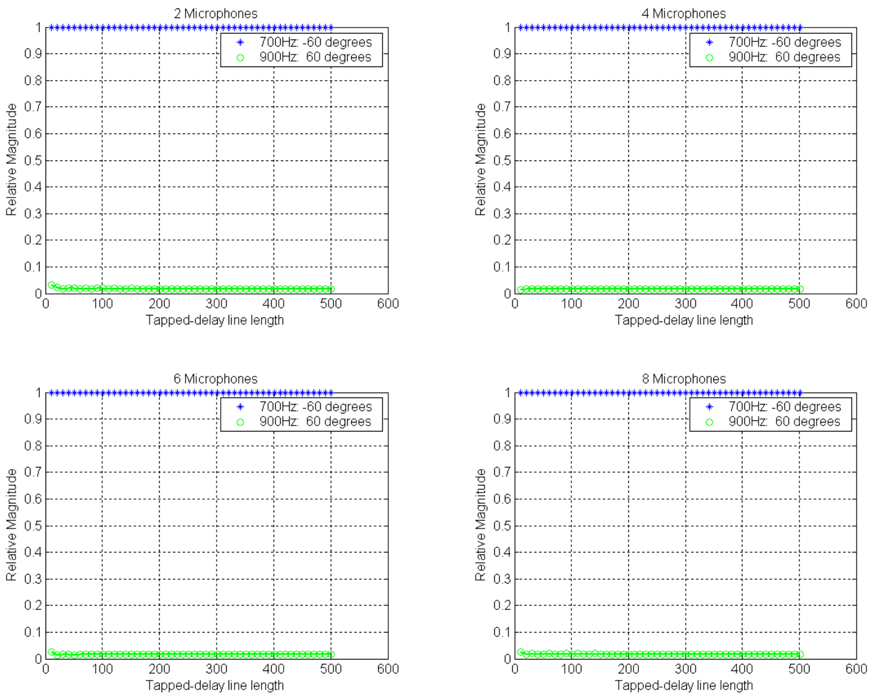

Figure 7.36: The effect of changing the length of the tapped-delay line for different microphone array sizes on the relative magnitude of the output of the GSC beamformer.

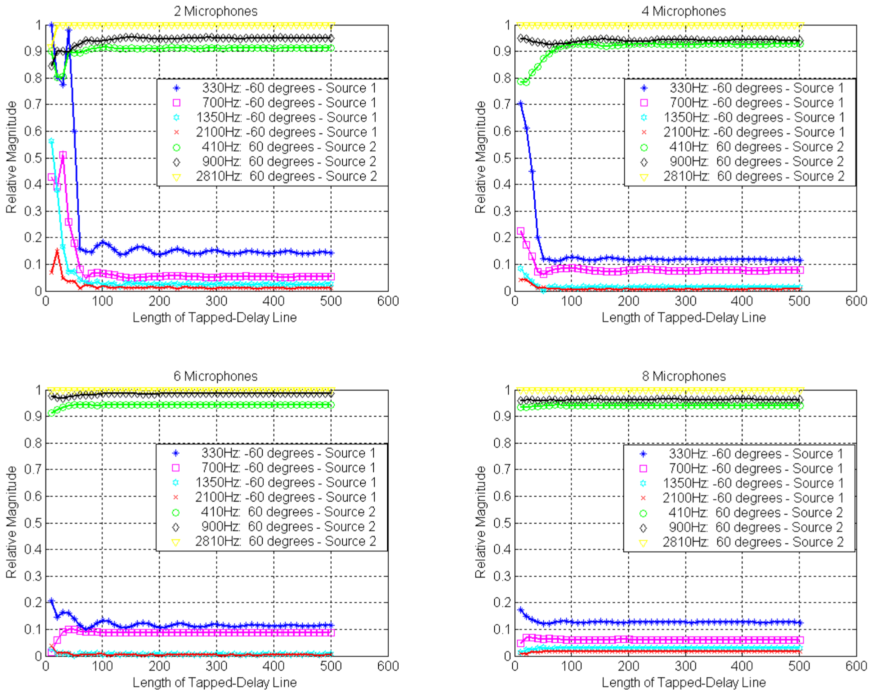

Figure 7.37: The effect of changing the length of the tapped-delay line for different microphone array sizes on the relative magnitude of the output of the GSC beamformer.

Simulated Broadband Signals

To test the broadband response, Scenario 2 was used. Steering the beamformer towards 60^o gave the result in Figure 7.37 and Steering the beamformer towards the other source gave the result in Figure 7.38.

Once again, the beamformer performed almost perfectly due to the blocking matrix successfully blocking all desired signals from the lower path of the beamformer.

A summary the performance of achieved by the GSC beamformer is presented below (Note: this point has been defined as the first time all noise signals pass below a relative magnitude of 0.1 or become stable.):

- For two microphones, a tapped-delay length of 91 was required.
- For four microphones, a tapped-delay length of 71 was required.
- For six microphones, a tapped-delay length of 71 was required.

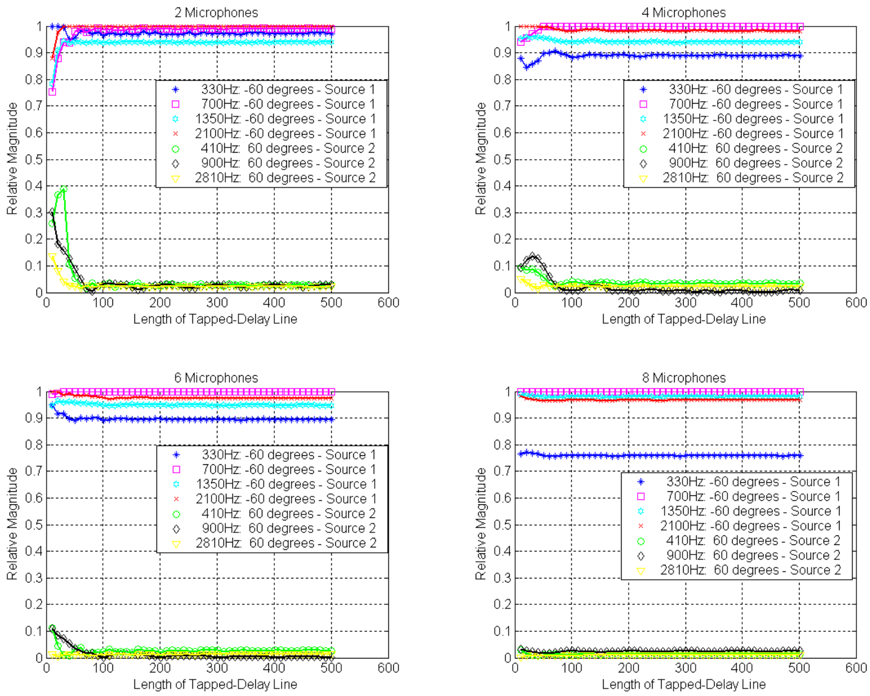

Figure 7.38: The effect of changing the length of the tapped-delay line for different microphone array sizes on the relative magnitude of the output of the GSC beamformer.

- For eight microphones, a tapped-delay length of 51 was required.

7.4.2 Evaluation using Complex Signals

The GSC beamformer was also evaluated using complex signals.

For the following tests, Scenario 3 was used and the beamformer was steered towards at an angle of -60^o.

The results from the previous section were been used to determine tapped-delay lengths for which to test the beamformer. However the performance of the beamformer was very poor with such short tapped-delay lines, therefore they were tested with the corresponding optimum lengths obtained by the Frost beamformer to compare the two. To assess whether the performance could improve, for each size array, a tapped-delay line of 501 was used. The results for:

- two microphones with tapped-delay line length of 231 and 501 have been presented in Figures 7.39 and 7.40 respectively;

- four microphones with tapped-delay line lengths of 201 and 501 have been presented in Figures 7.41 and 7.42 respectively;
- six microphones with tapped-delay line lengths of 181 and 501 have been presented in Figures 7.43 and 7.44 respectively; and
- eight microphones with tapped-delay line lengths of 131 and 501 have been presented in Figures 7.45 and 7.46 respectively.

For each array size there appeared to be a slight reduction in noise for the tapped-delay lengths obtained from the Frost beamformer. However, for each array size, a tapped-delay length of 501 appeared to produce a much greater reduction in noise. An assessment by listening to the outputs of the beamformer for each case backup these results. Additionally, the amount of noise present in each microphone array size and tapped-delay length was compared revealing that eight microphones with a tapped-delay length of 501 produced the best performance.

When the beamformer was steered to the female speaker at 60^o a similar result was obtained. Appendix B.2 details all audio files produced as the result of these tests.

7.4.3 Evaluation using Real Data

A series of tests were also conducted using data recorded from the experimental microphone array. Although some positive results were obtained using the Delay and Sum method, the GSC beamformer, like the Frost beamformer, did not produce any positive results. Figures 7.31, 7.32 where obtained using data from Scenario 4 and Figures 7.33 and 7.34 where obtained using data from Scenario 5. Notice that the beamformer cancelled out the desired signals and spurious noise was introduced. As mentioned in the chapter 5, the GSC beamformer is very susceptible to steering errors and these errors result in signal cancellation as can be seen the figures. These steering errors are most likely the result of the calculation of incorrect angles to the sources and errors in the placing of the microphones as stated previously.

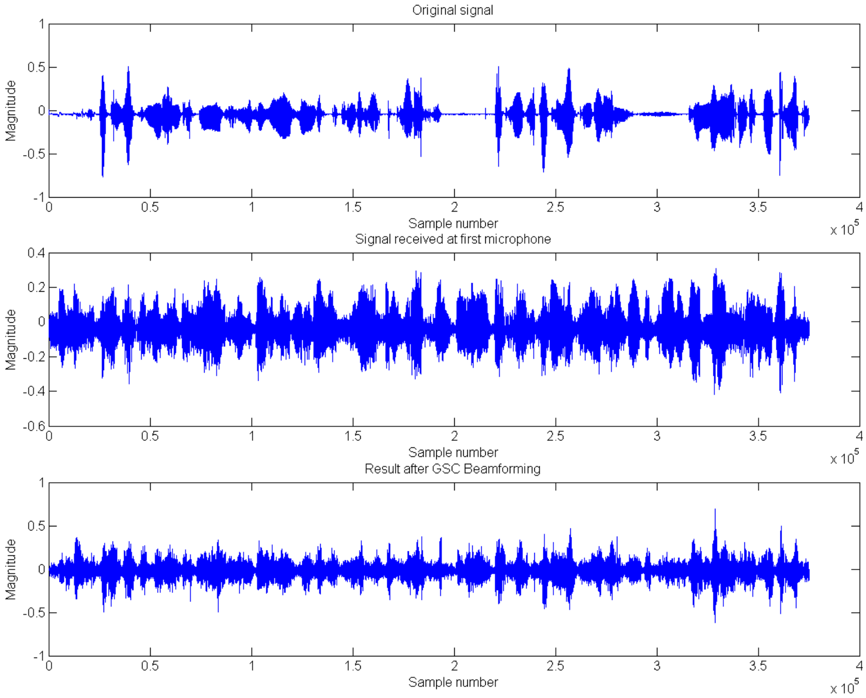

Figure 7.39: Result of GSC beamforming on complex signals. This beamformer employed two microphones and a tapped-delay length of 231.

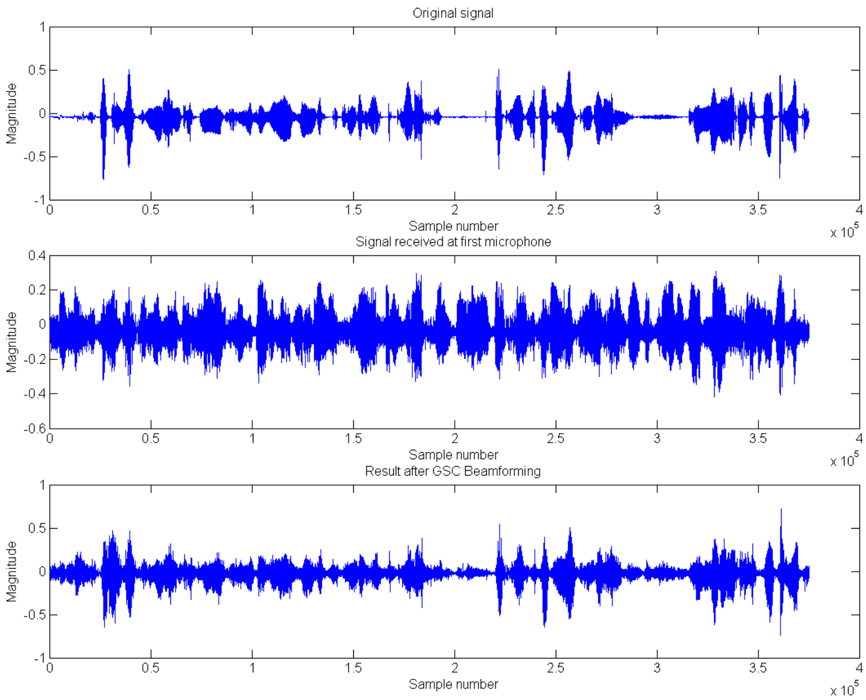

Figure 7.40: Result of GSC beamforming on complex signals. This beamformer employed two microphones and a tapped-delay length of 501.

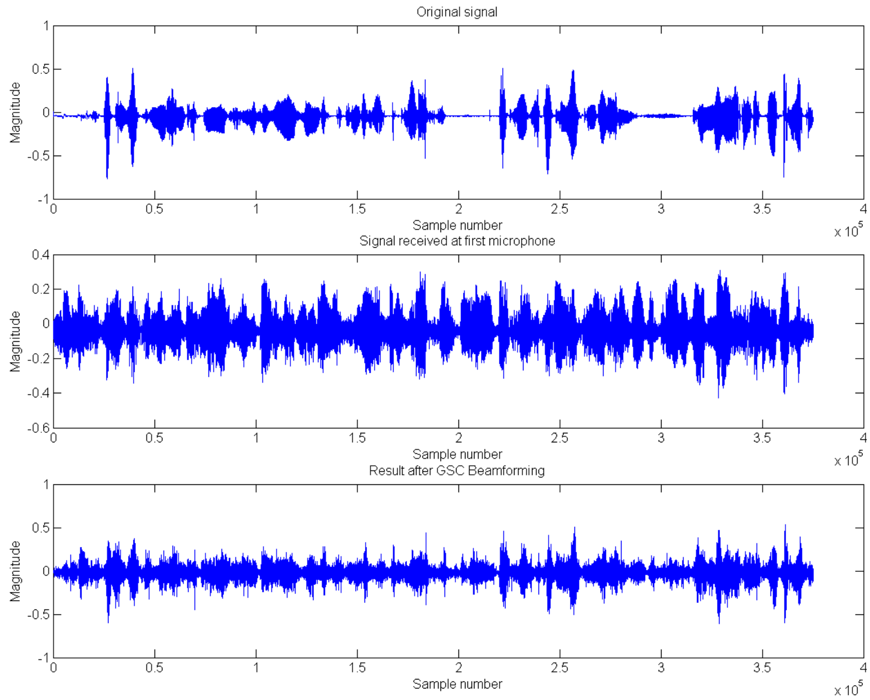

Figure 7.41: Result of GSC beamforming on complex signals. This beamformer employed four microphones and a tapped-delay length of 201.

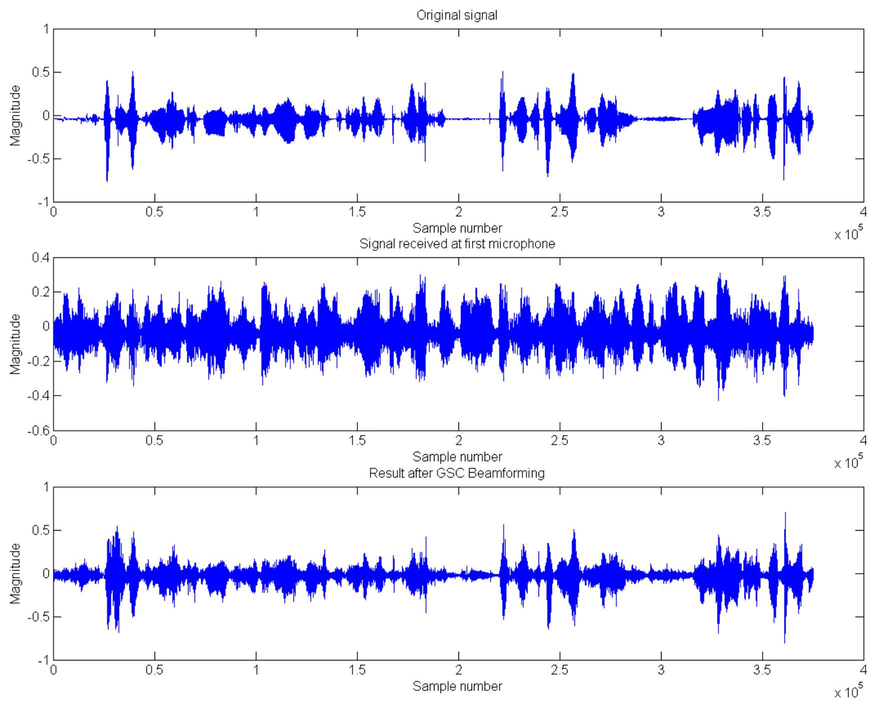

Figure 7.42: Result of GSC beamforming on complex signals. This beamformer employed four microphones and a tapped-delay length of 501.

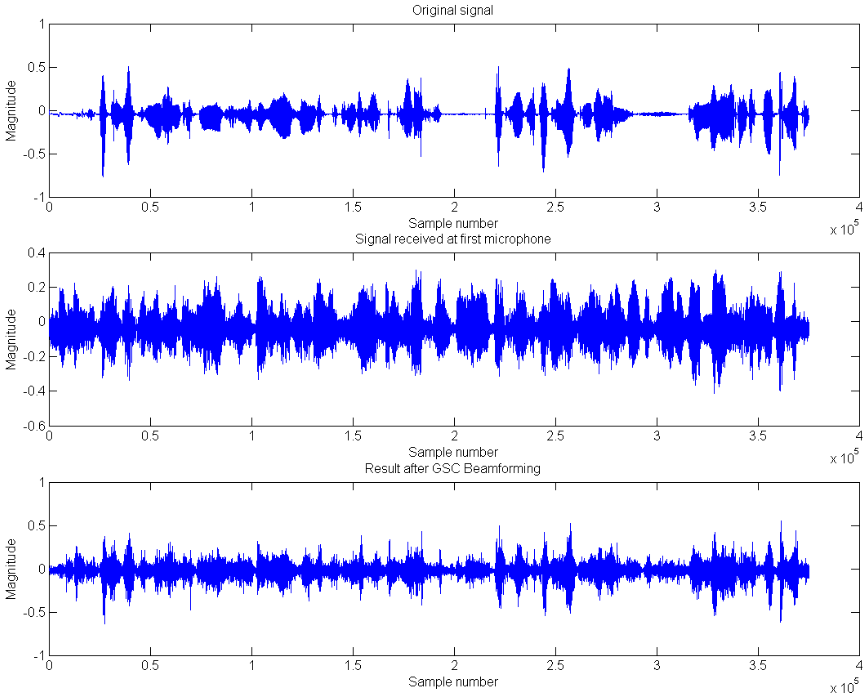

Figure 7.43: Result of GSC beamforming on complex signals. This beamformer employed six microphones and a tapped-delay length of 181.

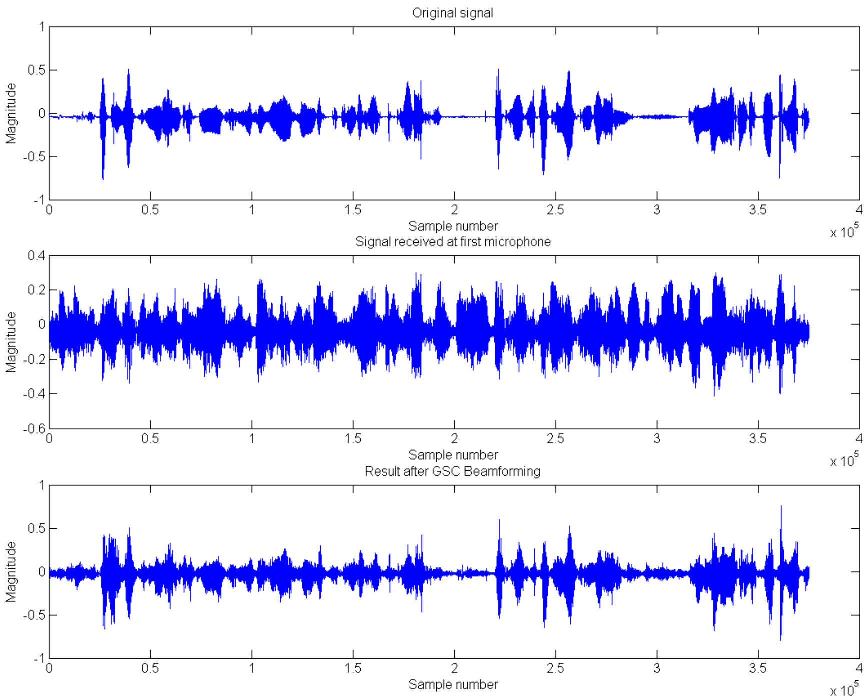

Figure 7.44: Result of GSC beamforming on complex signals. This beamformer employed six microphones and a tapped-delay length of 501.

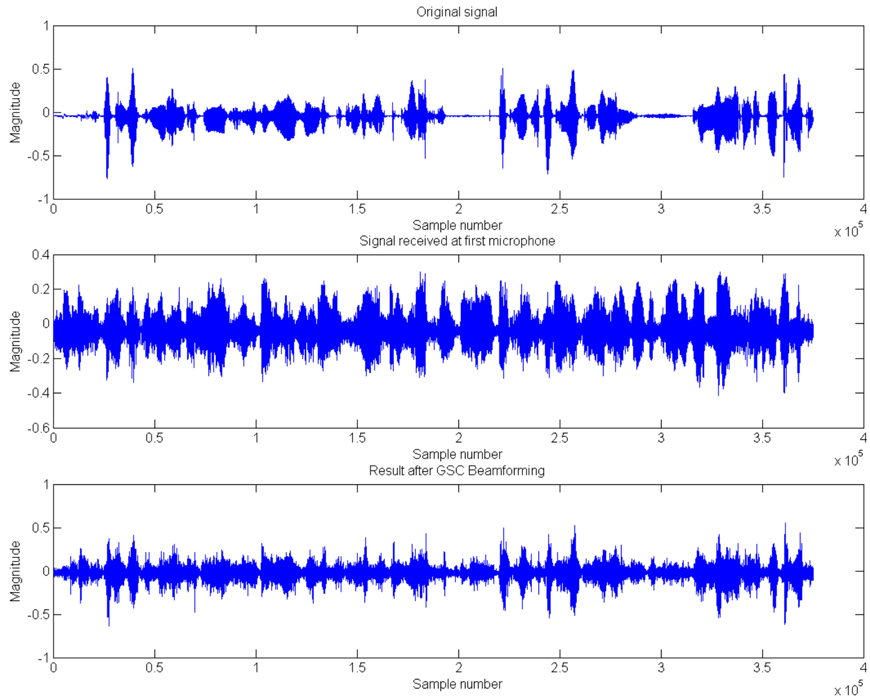

Figure 7.45: Result of GSC beamforming on complex signals. This beamformer employed eight microphones and a tapped-delay length of 131.

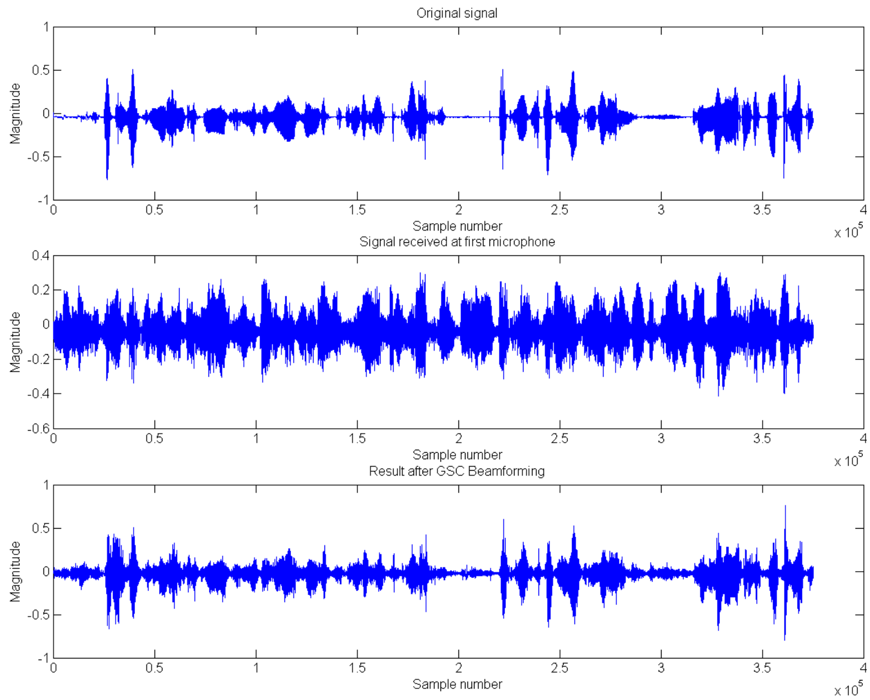

Figure 7.46: Result of GSC beamforming on complex signals. This beamformer employed eight microphones and a tapped-delay length of 501.

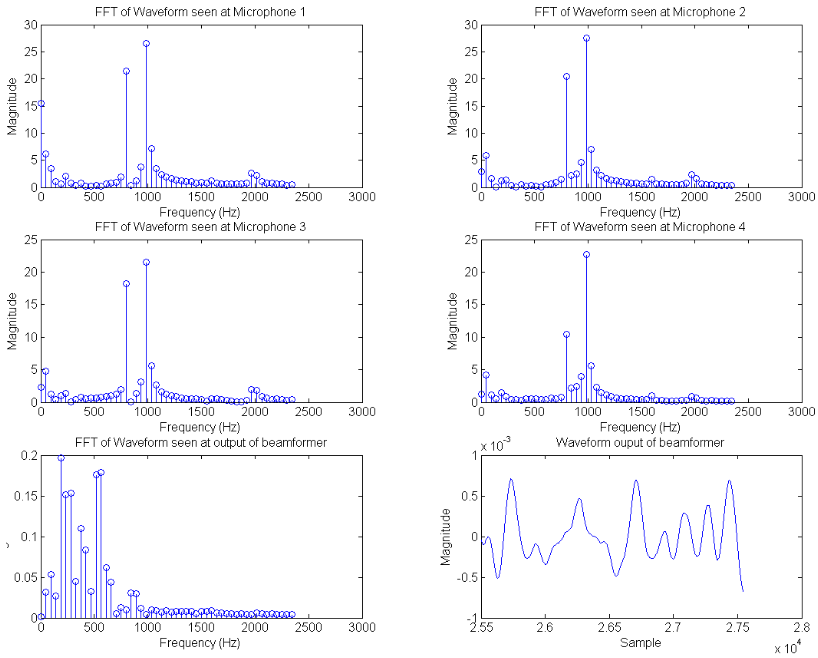

Figure 7.47: Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, J = 1001, spacing = 0.15m and look direction = -45^o.

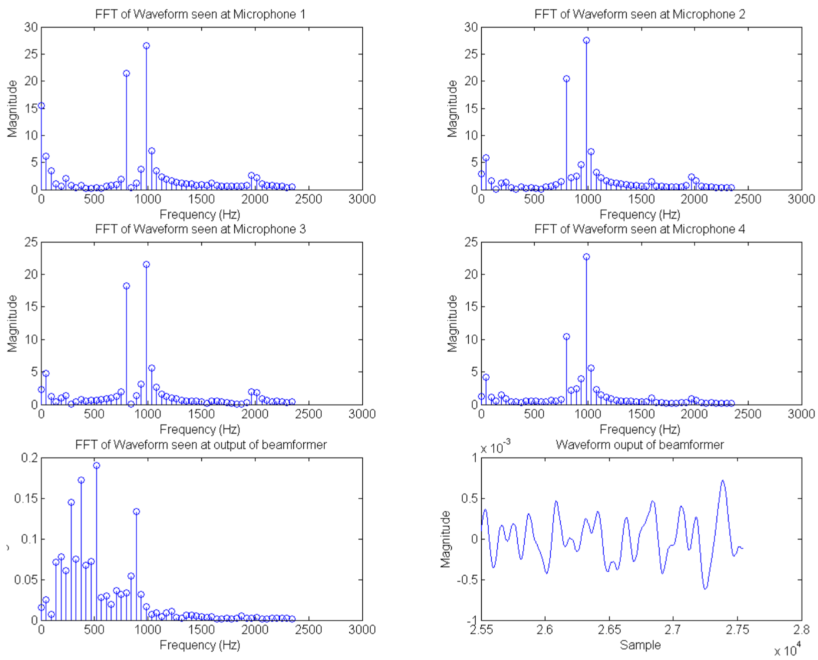

Figure 7.48: Source signals = [800Hz, 1000Hz], angles to sources = [-45^o, 45^o], number of microphones = 4, J = 1001, spacing = 0.15m and look direction = 45^o.

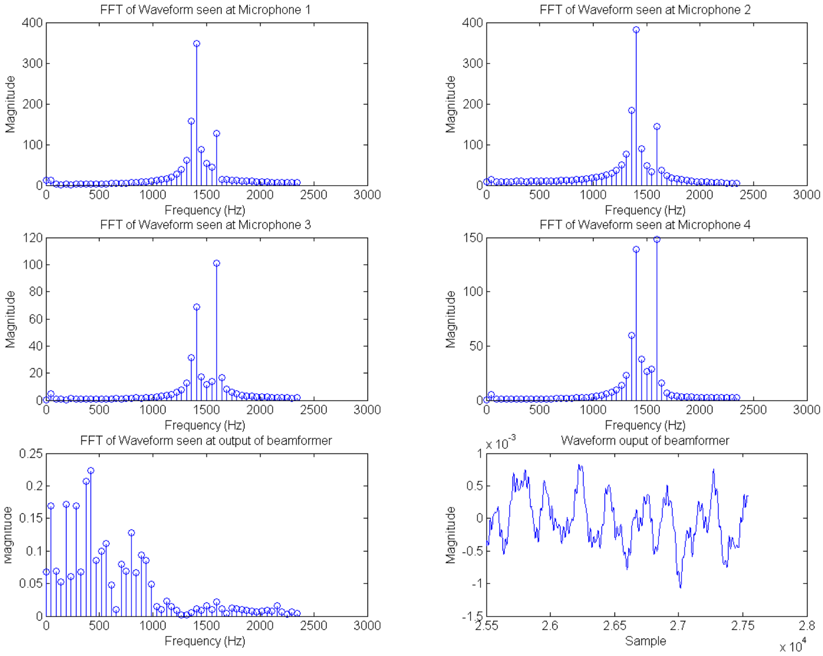

Figure 7.49: Source signals = [1400Hz, 1600Hz], angles to sources = [-60^o, 60^o], number of microphones = 4, J = 1001, spacing = 0.10m and look direction = -60^o.

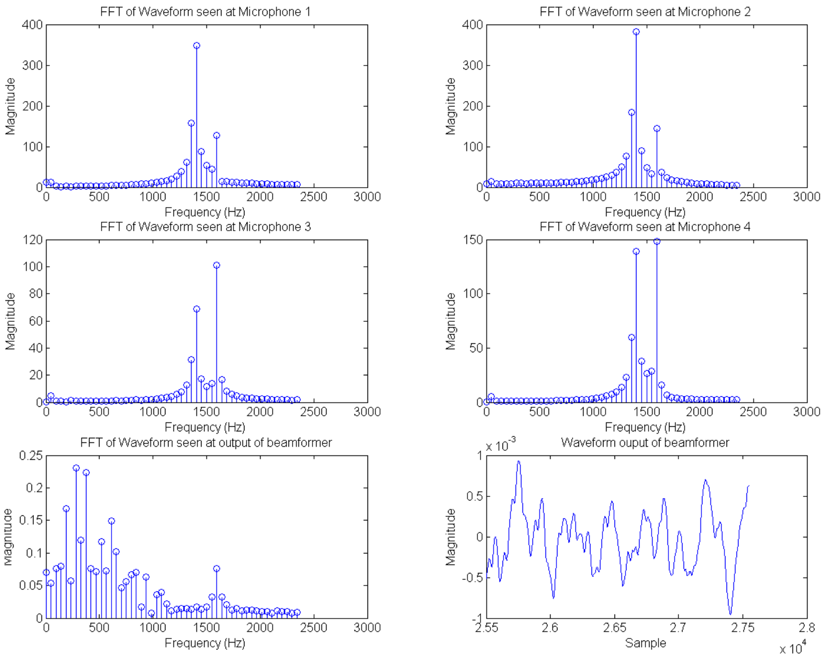

Figure 7.50: Source signals = [1400Hz, 1600Hz], angles to sources = [-60^o, 60^o], number of microphones = 4, J = 1001, spacing = 0.10m and look direction = 60^o.

7.4.4 Discussion

From the simulations using tones, it was concluded that as the length of the tapped-delay line increased, so did the performance of the beamformer - to a point. Past this point, the performance was very consistent over the remaining tapped-delay length values. Changing the number of microphones had the effect of changing the point at which the performance of the beamformer output stabilised. The more microphones that were present in the array, the sooner the output stabilised. The results have been summarised in table 7.2.

Number of Microphones	Optimum Tapped-Delay line length
2	91
4	71
6	71
8	51

Table 7.2: Summary of results for GSC beamformer using simple tones.

What was required by the complex data to achieve maximum performance can be seen in Table 7.3.

Number of Microphones	Optimum Tapped-Delay line length
2	501
4	501
6	501
8	501

Table 7.3: Summary of results for GSC beamformer using complex signals.

The GSC beamformer was tested using the same optimum tapped-delay lengths found by the Frost beamformer as well as a tapped-delay length of 501. It was determined that using the Frost tapped-delay lengths there was very little reduction in noise and with tapped-delay lengths of 501 the noise reduced considerably more, however, the performance still did not match that of the Frost beamformer.

The GSC beamformer performed very poorly when real data was used. As mentioned earlier, this was most likely the result of steering errors produced by inaccurate angles to the source signals and errors in the placing of the microphones in the array.

Although the beamformer performed very poorly on the data from the experimental microphone array, it performed very well in the simulations and, therefore, it has been concluded that the GSC beamformer is a candidate for use in security applications, particularly because only a small number of microphones are required to achieve a reasonable quality signal, requiring only a small microphone array.

7.5 Chapter Summary

The three beamformers that were implemented have all been tested using MATLAB. The microphone array and beamformer parameters were varied to determine the effects of:

1. the number of microphones;
2. a Hamming window on the Delay and Sum beamformer;
3. the effect of the distance between the microphones;
4. the length of tapped-delay lines and the number of microphones on the Frost and GSC beamformers.

Delay and Sum beamformer

For the Delay and Sum beamformer, it was determined that its performance was largely dependent on the number of microphones in the array. Increasing the number of microphones improved the beamformer's ability to reject "noise" signals up to a point from which any further addition resulted in little performance improvement. It was also determined that more microphones are required to attenuate lower frequency signals to the same degree as higher frequency signals, hence illustrating that the Delay and Sum beamformer is a narrowband beamformer. Increasing the spacing of the microphones also improved the performance of the beamformer. However this was at the expense of not being able to process higher frequency signals. Best performance was achieved when:

- using 5 microphones for two narrowband signals and 12 for narrowband signals with a Hamming window.
- using 10 microphones for broadband signals and 18 for broadband signals with a Hamming window.
- using a large spacing between microphones. This, however, as demonstrated by the results, places limitations on the maximum frequencies that can be attenuated without spatial filtering occurring.

The Delay and Sum beamformer also produced positive results when beamforming on real data. However, the performance did not match that of the simulated data. Although this was expected to a degree, it was largely emphasised by errors induced in the recording of the sound files. These errors are most likely attributed to the differences in the microphones and other equipment, and the inaccuracy in placing the microphones.

Frost beamformer

The Frost beamformer was able to achieve a high level of performance with only a small number of microphones, provided the beamformer had long tapped-delay lines. This, however, increased the computational complexity of the beamformer. For single frequency tones, the best results were achieved with:

- two microphones with a tapped-delay length of 231;
- four microphones with a tapped-delay length of 201;
- six microphones with a tapped-delay length of 181; and
- eight microphones with a tapped-delay length of 131.

The best performance achieved by the Frost beamformer, considering both the simple frequency tones and the complex signals, was eight microphones with a tapped-delay length of 131.

The Frost beamformer, however, produced no positive results using data recorded by the experimental microphone array. This was most likely due to the results presented previously.

GSC beamformer

The Generalised Sidelobe Canceller was able to achieve a high level of performance with only a small number of microphones when beamforming on the single frequency tones. For single frequency tones, the best results were achieved with:

- two microphones with a tapped-delay length of 91;

- four microphones with a tapped-delay length of 71;
- six microphones with a tapped-delay length of 71; and
- eight microphones with a tapped-delay length of 51.

However, when beamforming was performed on the complex signals, for each number of microphones a tapped-delay length of 501 was required to achieve best performance. In each instance ,the performance of the GSC beamformer did not match that of the Frost beamformer.

The GSC beamformer, however, like the Frost beamformer, produced no positive results using data recorded by the experimental microphone array most likely for the same reasons presented previously.

Chapter 8

Conclusions and Further Work

The beamformers implemented show a considerable capacity to remove unwanted noise based on the location of the source in space. For applications requiring clear and noise free speech reception, broadband beamformers, particularly adaptive beamformers, are well suited for this purpose.

This application is not without constraints, particularly that of Government legislation which has a considerable impact on how a microphone array can be used as a security device.

8.1 Reflection on Objectives

The objects of this project were:

1. Establish the potential need for directional microphone arrays in a security context and suggest examples of possible roles for which they can be used.

2. Investigate the legal implications of using microphone arrays in security applications.

3. Investigate the principles involved in creating a directional microphone array specifically designed for the purpose of beamforming.

4. Develop a software model of a directional microphone array in MATLAB for the purposes of beamforming.

5. Evaluate the software model by varying the parameters of its operation and of the test sound files.

6. Set up a microphone array using off the shelf studio microphones and record test files which can be used to evaluate the microphone array models.

In response to the objectives:

Response to Objective 1

Chapters One and Two addressed this objective. It was shown that Australia's security is constantly under threat, particularly from terrorism and organised crime. In the combat of these groups, surveillance devices, in particular listening devices, play a very important role. It was also established, from several sources, that traditional microphone systems used in a surveillance role suffer considerably in the presence of background interference and there are also devices built specifically to counter listening devices. Therefore, it was determined that there is room for a device capable of removing noise, regardless of whether is occupies the same temporal frequency band as the desired signal, based on its location in space.

It was also established that security systems such as ATMs, buildings and other resource and imformation access could greatly benefit from biometric security systems such as voice identification. However, these systems are also susceptible to noise and a robust solution is required to allow them to work effectively in noisy locations, something a microphone array can provide.

Response to Objective 2

Chapter Three addressed this objective. Extensive research was undertaken to locate relevant pieces of state and federal Government legislation relating to the use of listening devices. It was discovered that each state had its own Act(s) which set out the restrictions for the use of listening devices. In addition, there are several federal Acts for regulating the use of listening devices for federal agencies. These Acts, are all very similar. It was found that, generally, it is illegal to use any form of listening device to record any conversation unless that person has a warrant to use the listening device. A warrant can only be obtained for a serious indictable offence and, before such a warrant can be issued, numerous factors need to be considered.

The effect this legislation has on the use of a microphone array as a surveillance device is that it must be portable so that it can be installed and removed. It is important to note that, in this application, the users of the device would desire the recording to be executed discretely. Therefore the microphone array would have to be concealable.

In addition, the ethical issues that were faced in the execution of this project were presented.

Response to Objective 3

Chapter Four addressed the concept of beamforming using a microphone array. The Delay and Sum beamformer was first used to introduce the concept of beamforming and the theory behind it, as well as the concepts of Near-field and Far-field signals, spatial filtering, beamformer spatial response and narrowband and broadband beamforming.

Chapter Five then addressed the concept of the 2D frequency filter and the two different types of beamformers, data independent and statistically optimum. The theory behind two statistically optimum beamformers was then presented. The first presented was the Adaptive Frost beamformer which uses a constrained Least Mean Square algorithm to adapt the beamformer's weights to the optimum solution. The second beamformer presented was the Generalized Sidelobe Canceller, an adaptation of the Frost beamformer, which uses an unconstrained Least Mean Square algorithm to adapt the beamformer weights to the optimum solution.

Response to Objective 4

Chapter Six discussed the MATLAB implementation of the three beamformers that were explored and the microphone array model. The code for the beamformers can be seen Appendices C.2.1, C.2.2 and C.2.3.

Response to Objective 5

Chapter Seven addressed this objective. The three beamformers that were implemented were all tested using MATLAB. The microphone array and beamformer parameters were varied as well as the content of the test files to determine the effects of:

1. the number of microphones;
2. a Hamming window on the Delay and Sum beamformer;
3. the effect of the distance between the microphones; and
4. the length of tapped-delay lines and the number of microphones on the Frost and GSC beamformers.

For the Delay and Sum beamformer, is was determined that, as the number of microphones increased, so did the performance of the beamformer. It was also found that increasing the spacing between the microphones in the array improved the performance of the beamformer, however, at the reduction of the maximum frequency that can be processed by the beamformer. Best performance was achieved with the Delay and Sum beamformer when:

- using 5 microphones for two narrowband signals and 12 for narrowband signals with a Hamming window.
- using 10 microphones for broadband signals and 18 for broadband signals with a Hamming window.
- using a large spacing between microphones. This, however, as demonstrated by the results, places limitations on the maximum frequencies that can be attenuated without spatial filtering occurring.

The Frost beamformer was tested by determining the effect of varying the number of microphones in the array and the length of its tapped-delay lines on the output of the beamformer using different signals. A summary of the results obtained can be seen in Table 8.1.

Number of Microphones	**Optimum Tapped-Delay line length for Tones**	**Optimum Tapped-Delay line length for Complex signals**
2	231	231
4	201	201
6	181	181
8	131	131

Table 8.1: Summary of results for the Frost beamformer using single frequency tones and complex signals.

The Generalised Sidelobe Canceller was tested by varying the same parameters as those varied by the Frost beamformer. A summary of the results obtained can be seen in Table 8.2.

Number of Microphones	**Optimum Tapped-Delay line length for Tones**	**Optimum Tapped-Delay line length for Complex signals**
2	91	501
4	71	501
6	71	501
8	51	501

Table 8.2: Summary of results for the GSC beamformer using single frequency tones and complex signals

Although the GSC beamformer performed better than the Frost beamformer when beamforming on signal frequency tones, the Frost beamformer produced a much higher quality output when beamforming on the complex signals.

Response to Objective 6

Chapters Six and Seven also addressed this objective. Chapter six described the setup of the experimental microphone array and the process used to record signals from it.

Each of the three beamformers were evaluated using this data, however, only the Delay and Sum beamformer produced any positive results. These results indicated that the Delay and Sum beamforming does work with the desired signal in each test being reinforced and the undesired signal cancelled to a degree. The performance, however, was not as positive as the simulations.

As mentioned, the Frost and GSC beamformers performed poorly, cancelling out the desired and undesired signals when processing the recoded data. It is suggested that this was the result of steering errors and inaccurate placing of each of the microphones in the array.

8.2 Further Work

There are several areas for possible further work regarding this project. These areas include:

1. As mentioned earlier, source localisation is a significant capability provided by microphone arrays. This area could be investigated with the purpose of combining it with beamforming to create a security or surveillance system capable of locating and tracking a sound source.

2. The results obtained with real data were rather poor, however, this is believed to be the result of discrepancies in the experimental microphone array setup. Future work could focus on constructing a microphone array test bed which would allow accurate placing of the microphones and, possibly, the ability to change the configurations and spacing to evaluate their effects on real data.

3. As mentioned in chapter 5, there are several problems associated with the beamformers implemented. Future work could focus on some of the many other methods of beamforming which aim to overcome the problems of adaptive beamformers, such as computational efficiency and signal cancellation due to steering errors.

4. The work presented in this project is based on receiving Far-field signals. Future work could concentrate on algorithms for Near-field beamforming.

References

Abad, A. & Hernando, J. (2004), 'Integrated adaptive beamforming and wiener filtering for a robust microphone array', *Sensor Array and Multichannel Signal Processing Workshop Proceedings* pp. 367 – 371.

Acoustic Magic (n.d.).
`http://www.acousticmagic.com/`
viewed 27 March, 2006.

APL (n.d.), 'The second array project'.
`http://www.apl.utoronto.ca/projects/oldprojects.html`
viewed 1 May, 2005.

Australian Security Intelligence Organisation (2005), 'REPORT TO PARLIAMENT 2004 - 2005'.
`http://www.asio.gov.au/Publications/comp.htm`.

BBN Technologies (2006), 'Physical Security; Boomerang; Acoustic shooter detection for moving vehicles'.
`http://www.bbn.com/Products_and_Services/Physical_Security/`
`Boomerang_Sniper_Detection.html`
viewed 21 July, 2006.

Beasley, J. & Miller, G. (2002), *Modern Electronic Communication*, seventh edn, Prentice Hall, New Jersey.

Brandstein, M. & Ward, D. (2001), *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, New York.

CBS News (2000), 'The Spy Game'.
`http://www.cbsnews.com/stories/2000/01/31/60II/main155216.shtml`
viewed 21 July, 2006.

CBS News (2003), 'French, German EU Offices Bugged'.

CBS News (2005), 'Nicole Kidman Bugged In Australia'.
`http://www.cbsnews.com/stories/2005/01/25/entertainment/main669062.shtml?CMP=ILC-SearchStories`
viewed 21 July, 2006.

Cermin, S. (2004), 'Ears in the Sky: SwRI accomplishes airborne acoustic surveillance via unmanned aerial vehicles', *Technology Today* .
`http://www.swri.org/3pubs/ttoday/Winter04/Ears.htm`
viewed 15 September, 2006.

Chen, J., Yao, K. & Hudson, R. (2002), 'Source Localization and Beamforming', *Signal Processing Magazine, IEEE* **19**, 30 – 39.

Commonwealth of Australia (2005), 'Surveillance Devices Act 2004: Report for the year ending 30 June 2005'.
`http://www.ema.gov.au/agd/WWW/rwpattach.nsf/VAP/(0C3FC6E500CE9F81CFAAA22DF670141F)~Surveillance.pdf/$file/Surveillance.pdf`
viewed 21 July, 2006.

CyberCollege (2006), 'Microphones'.
`http://www.cybercollege.com/tvp038.htm`
viewed 8 April, 2006.

Defense Devices (n.d.), 'White Noise Generator Portable'.
`http://www.defensedevices.com/white-noise-generator.html`
viewed 15 October, 2006.

Department of Foreign Affairs and Trade (2004), *Transnational Terrorism: The Threat to Australia*, National Capital Printing, Canberra.

Endoacustica (n.d.), 'Miniaturised Bugs'.
http://www.endoacustica.com/english/miniaturised_bugs_en.htm
viewed 1 October, 2006.

Finnigan, J., Rodriguez, E., McPheeters, C. & Bass, J. (2004), Array signal processing, Technical report, Rice University.
http://cnx.org/content/col10255/latest/
viewed 25 March, 2006.

Griffiths, L. & Jim, C. (1982), 'An Alternative Approach to Linearly Constrained Adaptive Beamforming', *Antennas and Propagation, IEEE* **30**, 27 – 34.

Hiyane, K. & Iio, J. (2000), *Non-speech sound recognition with microphone array*, Mitsubishi Research Institute, Inc.
http://tosa.mri.co.jp/sounddb/nospeech/system/indexe.htm
viewed 25 March, 2006.

Hoshuyama, O., Sugiyama, A. & Hirano, A. (1999), 'A Robust Adaptive Beamformer for Microphone Arrays with a Blocking Matrix Using Constrained Adaptive Filters', *Signal Processing, IEEE* **47**, 2677 – 2684.

Invasion of Privacy Act 1971 (QLD) (n.d.).

Irwin, M. (2001), Policing Organised Crime, *in* '4^{th} National Outlook Symposium on Crime in Australia, New Crimes or New Responses', Canberra. 21-22 June 2001.

Kissell, J. (2004), *Array Microphones*, Joe Kissells Interesting Thing of the Day.
http://itotd.com/articles/334/
viewed 8 April, 2006.

Leis, J. (2002), *Digital Signal Processing A MATLAB-Based Tutorial Approach*, Research Studies Press Ltd, England.

Liu, W., Weiss, S. & Hanzo, L. (2005), 'A Generalized Sidelobe Canceller Employing Two-Dimensional Frequency Invariant Filters', *Antennas and Propagation, IEEE* **53**, 2339 – 2343.

Liu, W., Wu, R. & Langley, R., eds (2006), *Analysis and a Novel Design of the*

Beamspace Brodband Adaptive Array, Progress In Electomagnetics Research Symposium, UK.

McCowan, I., Pelecanos, J. & Sridharan, S. (n.d.), 'Robust Speaker Recognition using Microphone Arrays'.
`http://www.idiap.ch/~mccowan/publications/mccowan-odyssey.pdf`.

Menegatti, E., Mumolo, E., Nolich, M. & Pagello, E. (n.d.), 'A Surveillance System based on Audio and Video Sensory Agents cooperating with a Mobile Robot'.
`http://www.dei.unipd.it/~emg/papers/emgIAS8-Audio.pdf`
viewed 21 July, 2006.

Microsoft Encarta Encyclopedia Standard (2004), 'Sound'.

Naidu, P. (2001), *Sensor Array Signal Processing*, CRC Press, United States of America.

Nelson, S. & Schreck, D. (1998), 'Directional microphone array processing unit'.
`http://danschreck.com/microphone.pdf`
viewed 25 March, 2006.

Oteri, K. & Waterston, J. (2001), 'Adaptive Antenna Approach for GPS Multipath Mitigation'.

Owens, C. (2004), 'Mounting a Telinga Parabolic Microphone on a Camera Tripod'.
`http://homepage.ntlworld.com/christopher.owens2/TelingaMount.htm`
viewed 3 May, 2006.

Police Powers and Responsibilities Act 1997 (QLD) (n.d.).

Reichgott, M. (2005), 'Chicage Pairing Surveillance Cameras with Gunshot Recognition Systems', *Securityinfowatch* .

Russell, D. (2001), 'Sound Fields Radiated by Simple Sources'.
`http://www.kettering.edu/~drussell/Demos/rad2/mdq.html`
viewed 26 October, 2006.

Sekiguchi, T. & Karasawa, Y. (2000), 'Wideband Beamspace Adaptive Array Utilizing FIR Fan Filters for Multibeam Forming', *Signal Processing, IEEE* **48**, 277 – 284.

Spy Tech Agency (n.d.), 'Super Cardioid Shotgun Mic'.
http://store.yahoo.com/spytechagency/supcarshotmi.html
viewed 20 March, 2006.

Surveillance Device Laws Updated (2004).
http://www.ag.gov.au/agd/www/ministerruddockhome.nsf/page/media_releases_2004_first_quarter_24_march_2004_-_surveillance_device_laws_updated_-_0402004
viewed 21 July, 2006.

The Institution of Engineers Australia (2000), 'CODE OF ETHICS'.
http://www.ieaust.org.au/about_us/res/downloads/Code_of_Ethics_2000.pdf.

Van Veen, B. & Buckley, K. (1988), 'Beamforming: A Versatile Approach to Spatial Filtering', *ASSP Magazine, IEEE* **5**, 4 – 24.

Vanstone, A. (2000), Welcome Address - Transnational crime today, *in* 'Transnational Crime', Canberra. 9-10 March 2000.

Ward, D., Williamson, R. & Kennedy, R. (1998), 'Broadband microphone arrays for speech acquisition'.
http://users.rsise.anu.edu.au/~williams/papers/P97.pdf
viewed 21 July, 2006.

Weinstein, E., Steele, K., Agarwal, A. & Glass, J. (2004), 'LOUD: A 1020-Node Modular Microphone Array and Beamformer for Intelligent Computing Spaces'.
http://cag.csail.mit.edu/mic-array/
viewed 20 March, 2006.

Yun, W. (2002), 'The '123' of Biometric Technology', *Synthesis Journal* .
http://www.itsc.org.sg/synthesis/2002/biometric.pdf
viewed 15 October, 2006.

Zou, Q., Liang Yu, Z. & Lin, Z. (2004), 'A Robust Algorithm for Linearly Constrained Adaptive Beamforming', *Signal Processing Letters, IEEE* **11**, 26 – 29.

Appendix A

Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111/2 Research Project

PROJECT SPECIFICATION

FOR: Benjamin Jon COBB

TOPIC: Directional Microphone Array for Security Applications

SUPERVISOR: Dr John Leis

PROJECT AIM: This project aims to:
- Investigate microphone arrays and their potential within a security setting
- Construct, in software, a directional microphone array model.

PROGRAMME: **Issue B, 26th October, 2006**

1. Establish the potential need for directional microphone arrays in a security context and suggest examples of possible roles for which they can be used.

2. Investigate the legal implications of using microphone arrays in security applications.

3. Investigate the concept of beamforming using a directional microphone array.

4. Develop software models of several beamformers and a microphone array, to supply data to those beamformers, in Matlab™.

5. Evaluate the software models by varying the parameters of their operation and of the test data.

6. Setup a microphone array using off the shelf studio microphones and record test files to evaluate the beamformers.

As time permits

7. Design and construct a two to four microphone array test bed and compare it to the software model

AGREED:

Student: _____________ Supervisor: _____________

Date: ___ / ___ / ___ Date: ___ / ___ / ___

Appendix B

Included Files

B.1 Recorded Data

The directory "BenCOBB_appendices/Recorded Data/" on the included CD contains the record test files from the experimental microphone array. Descriptions of these files can be seen in Table B.1.

B.2 Frost Complex Signal Results

The directory "BenCOBB_appendices/Frost Results/" on the included CD contains the Frost beamforming test results on complex signals. Descriptions of these files can be seen in Table B.2.

B.3 GSC Complex Signal Results

The directory "BenCOBB_appendices/GSC Results/" on the included CD contains the GSC beamforming test results on complex signals. Descriptions of these files can be seen in Table B.3.

ID	Spacing (m)	Filenames	Signal 1	Angle 1	Signal 2	Angle 2
1	0.150	MIC *_01.wav	1000Hz	45^o	800Hz	-45^o
2	0.150	MIC *_02.wav	900Hz	0^o	600Hz	-60^o
3	0.100	MIC *_03.wav	1600Hz	45^o	800Hz	-60^o
4	0.100	MIC *_04.wav	1600Hz	60^o	1400Hz	-60^o
5	0.100	MIC *_05.wav	900Hz	-30^o	600Hz	-70^o
6	0.100	MIC *_06.wav	600Hz	45^o	900Hz	-45^o
7	0.055	MIC *_07.wav	3000Hz	45^o	2000Hz	-45^o
8	0.055	MIC *_08.wav	900Hz	45^o	600Hz	-60^o
9	0.055	MIC *_09.wav	1000Hz	60^o	1200Hz	-60^o

Table B.1: Summary of recorded data. Note: all files were sampled at a rate of 96000Hz.

Filename	Number of Microphones	Tapped-delay length	Look direction
FROST_TEST_MIC1.wav	-	-	No beamforming
FROST_2Mics_231Tap_-60Look.wav	2	231	-60^o
FROST_2Mics_231Tap_+60Look.wav	2	231	60^o
FROST_2Mics_501Tap_-60Look.wav	2	501	-60^o
FROST_2Mics_501Tap_+60Look.wav	2	501	60^o
FROST_4Mics_201Tap_-60Look.wav	4	201	-60^o
FROST_4Mics_201Tap_+60Look.wav	4	201	60^o
FROST_4Mics_501Tap_-60Look.wav	4	501	-60^o
FROST_4Mics_501Tap_+60Look.wav	4	501	60^o
FROST_6Mics_181Tap_-60Look.wav	6	181	-60^o
FROST_6Mics_181Tap_+60Look.wav	6	181	60^o
FROST_6Mics_501Tap_-60Look.wav	6	501	-60^o
FROST_6Mics_501Tap_+60Look.wav	6	501	60^o
FROST_8Mics_131Tap_-60Look.wav	8	131	-60^o
FROST_8Mics_131Tap_+60Look.wav	8	131	60^o
FROST_8Mics_501Tap_-60Look.wav	8	501	-60^o
FROST_8Mics_501Tap_+60Look.wav	8	501	60^o

Table B.2: Summary of results for Frost algorithm on complex signals. Note: microphone array in all cases employed four microphones with a spacing of 0.055m.

Filename	**Number of Microphones**	**Tapped-delay length**	**Look direction**
GSC_TEST_MIC1.wav	-	-	No beamforming
GSC_2Mics_231Tap_-60Look.wav	2	231	-60^o
GSC_2Mics_231Tap_+60Look.wav	2	231	60^o
GSC_2Mics_501Tap_-60Look.wav	2	501	-60^o
GSC_2Mics_501Tap_+60Look.wav	2	501	60^o
GSC_4Mics_201Tap_-60Look.wav	4	201	-60^o
GSC_4Mics_201Tap_+60Look.wav	4	201	60^o
GSC_4Mics_501Tap_-60Look.wav	4	501	-60^o
GSC_4Mics_501Tap_+60Look.wav	4	501	60^o
GSC_6Mics_181Tap_-60Look.wav	6	181	-60^o
GSC_6Mics_181Tap_+60Look.wav	6	181	60^o
GSC_6Mics_501Tap_-60Look.wav	6	501	-60^o
GSC_6Mics_501Tap_+60Look.wav	6	501	60^o
GSC_8Mics_131Tap_-60Look.wav	8	131	-60^o
GSC_8Mics_131Tap_+60Look.wav	8	131	60^o
GSC_8Mics_501Tap_-60Look.wav	8	501	-60^o
GSC_8Mics_501Tap_+60Look.wav	8	501	60^o

Table B.3: Summary of results for GSC algorithm on complex signals. Note: microphone array in all cases employed four microphones with a spacing of 0.055m.

Appendix C

Source Code

C.1 Microphone Array Simulation

C.1.1 arraySim.m

```
% Function:      arraySim.m
% [mics] = ARRAYSIM(Fs, numMics, spacing, c, sourceSignals, inputAnglesArray)
% Written By:   Benjamin Cobb - 2006
% Description:  Simulate a microphone array
% Inputs:       Fs                = the sampling frequency
%               numMic            = the number of microphones to be in the array
%               spacing           = the distance between the microphones in the array
%               sourceSignals     = a N-by-M matrix of source signals to be received by
%                                   the array, where N is the number of signals and M is
%                                   the length of the signals.
%               inputAnglesArray  = an N length vector of angles from which the
%                                   source signals arrive from.
% Outputs:      mics              = N-by-M matrix of values that represent the signals received
%                                   by the array. Each row represents a microphone.

function [mics] = ARRAYSIM(Fs, numMics, spacing, c, sourceSignals, inputAnglesArray);

%Check that the length of source signals and angles are the same
sourceDim = size(sourceSignals);
if sourceDim(1) ~= length(inputAnglesArray)
    error('Argument lengths dont match', ...
        'The number of given frequencies does not match the number of given angles');
end


%                       |
%                       |
%        - ve           |          + ve
%                       |
%                       |
%-----------------------|----------------------
%           Mic1    Mic2    Mic3 ... MicN

%Create matrix to store the microphone array signals
mics = zeros(numMics, sourceDim(2));

%Find the time delay between each microphone for each angle
timeDelay = abs((spacing / c) .* sin(inputAnglesArray));
%Convert the time delay to delay in samples
sampleDelay = round(timeDelay ./ (1 / Fs));
%Find the maximum delay that will be experienced by a microphone in the
%array
maxDelay = sampleDelay * (numMics - 1);

%For every source signal
for x = 1:length(inputAnglesArray)
    %For every microphone in the array
    for m = 1:numMics
        if inputAnglesArray(x) == 0
            %Signal coming from 0 degrees
            mics(m,:) = mics(m,:) + sourceSignals(x,:);

        elseif inputAnglesArray(x) > 0
            %Signal coming from a positive angle
            delay = maxDelay(x) - ((m - 1) * sampleDelay(x));
            mics(m,:) = mics(m,:) + [zeros(1,delay) sourceSignals(x,1:length(sourceSignals) - delay)];

        else
            %Signal coming from a negative angle
            delay = (m - 1) * sampleDelay(x);
            mics(m,:) = mics(m,:) + [zeros(1,delay) sourceSignals(x,1:length(sourceSignals) - delay)];
        end
    end
```

```
end
```

C.2 Beamformer Implementations

C.2.1 delayAndSumMethod.m

```
% Procedure:    delayAndSumMethod.m
% Written By:   Benjamin Cobb - 2006
% Description: Perform delay and sum beamforming on a set of signals that
%               are defined by the program

clc;                %Clear the command window
clear;              %Clear all variables
% close all;          %Close all open figures

%Angle in which to point the beamformer
lookDirection = 60;
%Convert the look direction to radians
lookDirection = (pi*lookDirection)/180;


%Simulation variables
numMics = 4;        %The number of microphones in the array
spacing = 0.05;     %The spacing between the microphone (meters)
time = 1;           %The length, in seconds, for which to run the simulation (modes 0 and 1)
Fs = 48000;         %The sample rate
c = 344;            %The speed of Sound at 20 degrees

%Find Microphone weights
applyHammingWindow = 1; %Apply a Hamming window the inputs
if applyHammingWindow
    Wn = hammingWeights(numMics);
else
    Wn = ones(1,numMics)*(1/numMics);
end

%=====================================================================
%Generate array
%=====================================================================

%Mode of simulation:
%   0 = Frequency sources
%   1 = Recorded sources
%   2 = Array recorded data
mode = 2;

%Find the desired sources
if mode == 0    %Frequency sources

    %The frequency(s) of the sound sources
    inputFreqsArray = [700, 900, 330, 1350, 2100];
    %The position(s) of the sound sources
    inputAnglesArray = [-60, 60, -60, -60, -60];
    %Convert input angles to radians
    inputAnglesArray = (inputAnglesArray * pi) / 180;

    %Time for which to run simulation
    t = 0:1/Fs:time;

    %Matrix to store source signals
    sourceSignals = zeros(length(inputFreqsArray), length(t));

    %Declare originalSignal - stores the original signal
    originalSignal = zeros(1,length(t));

    %Create signals
    for x = 1:length(inputFreqsArray)
        sourceSignals(x,:) = sin(2*pi*t*inputFreqsArray(x));

        %Capture the original signal to calculate the SNR once finished
```

```
        if inputAnglesArray(x) == lookDirection
            originalSignal = originalSignal + sourceSignals(x,:);
        end
    end

    %Generate microphone array data
    micarray = ARRAYSIM(Fs, numMics, spacing, c, sourceSignals, inputAnglesArray);

    %Clear unneeded variables
    clear inputFreqsArray inputAnglesArray t time sourceSignals;

elseif mode == 1    %Recorded Sources

    %Load in the test files
    [y1, Fs] = wavread('C:\Documents and Settings\Ben\Desktop\test4.wav');
    [y2, Fs] = wavread('C:\Documents and Settings\Ben\Desktop\test5.wav');
    [y3, Fs] = wavread('C:\Documents and Settings\Ben\Desktop\tv1.wav');

    %Ensure that each signal has the same length
    len = min([length(y1) length(y2) length(y3)]);
    len = len/2;
    y1 = y1(1:len);
    y2 = y2(1:len);
    y3 = y3(1:len);

    %Assign each loaded signal to sourceSignals
    sourceSignals= zeros(3,length(y1));
    sourceSignals(1,:) = y1';
    sourceSignals(2,:) = y2';
    sourceSignals(3,:) = y3';

    %The positions of the sound sources
    inputAnglesArray = [-80, 80, 0];
    %Convert input angles to radians
    inputAnglesArray = (inputAnglesArray * pi) / 180;

    %Declare originalSignal - stores the original signal
    originalSignal = zeros(1,len);

    %Capture the original signal to calculate the SNR once finished
    for x = 1:length(inputAnglesArray)
        if inputAnglesArray(x) == lookDirection
            originalSignal = sourceSignals(x,:);
        end
    end

    % Generate microphone array data
    micarray = ARRAYSIM(Fs, numMics, spacing, c, sourceSignals, inputAnglesArray);

    %Clear unneeded variables
    clear y1 y2 y3 len inputAnglesArray sourceSignals;

else    %Array recorded data

    %Load in data
    %Track number
    Track = '08';
    %Relative address of files
    File1 = 'Data\MIC_1_';
    File2 = 'Data\MIC_2_';
    File3 = 'Data\MIC_3_';
    File4 = 'Data\MIC_4_';
    [y1,Fs]=WAVREAD(strcat(File1,Track,'.wav'));
    [y2,Fs]=WAVREAD(strcat(File2,Track,'.wav'));
    [y3,Fs]=WAVREAD(strcat(File3,Track,'.wav'));
    [y4,Fs]=WAVREAD(strcat(File4,Track,'.wav'));

    len = length(y1);
    len = len / 1;

    y1 = y1(1:len);
    y2 = y2(1:len);
    y3 = y3(1:len);
    y4 = y4(1:len);

    %Assign each signal to micarray
    micarray = zeros(4,length(y1));
    micarray(1,:) = y4';
```

```
    micarray(2,:) = y3';
    micarray(3,:) = y2';
    micarray(4,:) = y1';

    %Clear unneeded variables
    clear Track File1 File2 File3 File4 len y1 y2 y3 y4
end

%Store the signal from the first mic a signal from the lookDirection
%will encounter
if lookDirection >= 0
    originalMic = micarray(numMics,:);
else
    originalMic = micarray(1,:);
end

%==========================================================================
%Perform Delay and Sum Beamforming
%==========================================================================

%Find the time delay between each microphone for the look direction
timeDelay = abs((spacing / c) .* sin(lookDirection));
%Convert the time delay to delay in samples
sampleDelay = round(timeDelay / (1 / Fs));
%Find the maximum delay that will be experienced by a microphone in the
%array
maxDelay = sampleDelay * (numMics - 1);

steeredSignals = zeros(numMics, length(micarray));
for x = 1:numMics

    if lookDirection == 0
        %Look direction of 0 degrees
        steeredSignals(x,:) = micarray(x,:);

    elseif lookDirection > 0
        %Look direction positive degrees
        delay = (x - 1) * sampleDelay;
        steeredSignals(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-
            delay)];

    else
        %Look direction negative degrees
        delay = maxDelay - ((x - 1) * sampleDelay);
        steeredSignals(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-
            delay)];

    end
end

%Add each individual microphone together
steeredSum = zeros(1,length(steeredSignals));
for x = 1:numMics
    steeredSum = steeredSum + Wn(x)*steeredSignals(x,:);
end


%Play back the result
% sound(steeredSum, Fs);     %UNCOMMENT THIS!!!!!!!!!!!!!!!!!!!!!!!!!

%==========================================================================
%Analyse Results
%==========================================================================

%Plot the original signal, the microphone signal and the steeredSum and
%their corresponding frequency spectrums
%Only for mode 0 and 1 because no original signal for the array recorded
%data. Plot the fft as well
if mode == 0 || mode == 1

    %Normalise each signal by the energy in the original signal so that they
    %can be compared

    %Energy in original signal
    ein = sqrt(sum(originalSignal.*originalSignal)/length(originalSignal));

    %Normalise energy in microphone signal
    eout = sqrt(sum(originalMic.*originalMic)/length(originalMic));
    originalMic = originalMic / eout;
    originalMic = originalMic * ein;

    %Normalise energy in steeredSum signal
```

```
    eout = sqrt(sum(steeredSum.*steeredSum)/length(steeredSum));
    steeredSum = steeredSum / eout;
    steeredSum = steeredSum * ein;

    figure

    %Plot three signals - The original, the signal seen that the microphone
    %and the signal after beamforming

    %Plot the original signal
    subplot(3,1,1);
    set(gca, 'fontsize', 12)
    plot(originalSignal);
    title('Original signal');
    xlabel('Sample number');
    ylabel('Magnitude');

    %Plot the microphone signal
    subplot(3,1,2)
    set(gca, 'fontsize', 12)
    plot(originalMic);
    title('Signal received at first microphone');
    xlabel('Sample number');
    ylabel('Magnitude');

    %Plot the recovered signal
    subplot(3,1,3)
    set(gca, 'fontsize', 12)
    plot(steeredSum);
    title('Result after Delay and Sum Beamforming');
    xlabel('Sample number');
    ylabel('Magnitude');

    %Plot  the waveforms and frequency spectrums of the original signal,
    %the signal received at the microphones and the output of the
    %beamformer
    figure;

    block = 500:1523;   %block size = 1024
    step = 1/length(block);
    freqLabel = 0:step:1-step;  %Find increments of x
    freqLabel = freqLabel * Fs;        %Multiply by the sample frequency
    %The amount of the spectrum to see, i.e. 3 for 1/3
    amountSpec = 20;

    %Plot the originalSignal
    subplot(3,2,1);
    set(gca, 'fontsize', 12)
    plot(block,originalSignal(block));
    title('Original Signal');
    xlabel('Sample number');
    ylabel('Magnitude');
    grid on;

    subplot(3,2,2);
    set(gca, 'fontsize', 12)
    freq = abs(fft(originalSignal(block)));
    stem(freqLabel(1:length(block)/amountSpec), freq(1:length(block)/amountSpec));
    title('FFT of Original Signal');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Plot the originalMic
    subplot(3,2,3);
    set(gca, 'fontsize', 12)
    plot(block, originalMic(block));
    title('Waveform seen at Microphone');
    xlabel('Sample number');
    ylabel('Magnitude');
    grid on;

    subplot(3,2,4);
    set(gca, 'fontsize', 12)
    freq = abs(fft(originalMic(block)));
    stem(freqLabel(1:length(block)/amountSpec), freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at Microphone');
```

```
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Plot steeredSum
    subplot(3,2,5);
    set(gca, 'fontsize', 12)
    plot(block, steeredSum(block));
    title('Waveform after Delay and Sum Beamforming');
    xlabel('Sample number');
    ylabel('Magnitude');
    grid on;

    subplot(3,2,6);
    set(gca, 'fontsize', 12)
    freq = abs(fft(steeredSum(block)));
    stem(freqLabel(1:length(block)/amountSpec), freq(1:length(block)/amountSpec));
    title('FFT of Waveform after Delay and Sum Beamforming');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');
else
    %Normalise the energy in the output of the beamformer by the average in
    %the array

    %Energy in original signal
    einMic1 = sqrt(sum(micarray(1,:).*micarray(1,:))/length(micarray(1,:)));
    einMic2 = sqrt(sum(micarray(2,:).*micarray(2,:))/length(micarray(2,:)));
    einMic3 = sqrt(sum(micarray(3,:).*micarray(3,:))/length(micarray(3,:)));
    einMic4 = sqrt(sum(micarray(4,:).*micarray(4,:))/length(micarray(4,:)));
    ein = (einMic1 + einMic2 + einMic3 + einMic4) / 4;

    %Normalise energy in steeredSum signal
    eout = sqrt(sum(steeredSum.*steeredSum)/length(steeredSum));
    steeredSum = steeredSum / eout;
    steeredSum = steeredSum * ein;

    %Plot the frequency spectrums of each of the four microphones and of
    %the output of the beamformer.

    figure;
    block = 500:2547;   %block size = 2048
    step = 1/length(block);
    freqLabel = 0:step:1-step;  %Find increments of x
    freqLabel = freqLabel * Fs;         %Multiply by the sample frequency
    %The amount of the spectrum to see, i.e. 3 for 1/3
    amountSpec = 40;

    %Microphone 1
    subplot(3,2,1);
    set(gca, 'fontsize', 12)
    freq = abs(fft(micarray(1,block)));
    stem(freqLabel(1:length(block)/amountSpec),freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at Microphone 1');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Microphone 2
    subplot(3,2,2);
    set(gca, 'fontsize', 12)
    freq = abs(fft(micarray(2,block)));
    stem(freqLabel(1:length(block)/amountSpec),freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at Microphone 2');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Microphone 3
    subplot(3,2,3);
    set(gca, 'fontsize', 12)
    freq = abs(fft(micarray(3,block)));
    stem(freqLabel(1:length(block)/amountSpec),freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at Microphone 3');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Microphone 4
```

```
    subplot(3,2,4);
    set(gca, 'fontsize', 12)
    freq = abs(fft(micarray(4,block)));
    stem(freqLabel(1:length(block)/amountSpec),freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at Microphone 4');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Output of beamformer
    subplot(3,2,5);
    set(gca, 'fontsize', 12)
    freq = abs(fft(steeredSum(block)));
    stem(freqLabel(1:length(block)/amountSpec),freq(1:length(block)/amountSpec));
    title('FFT of Waveform seen at output of beamformer');
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');

    %Waveform out of beamformer
    subplot(3,2,6);
    set(gca, 'fontsize', 12)
    plot(block,steeredSum(block));
    title('Waveform ouput of beamformer');
    xlabel('Sample');
    ylabel('Magnitude');
end
```

C.2.2 frostBeamformer.m

```
% Function:      frost.m
% function [result] = frostBeamformer(c, Fs, numMics, d, tapOrder, lookDirection, micarray)
% Written By:  Benjamin Cobb - 2006
% Description: Perform Frost beamforming on a set of signals that are
%              defined by the program
% Inputs:      c                = the speed of sound
%              Fs               = the sampling frequency
%              numMics          = the number of microphones in the array
%              d                = the distance between each mic in the
%                                 array
%              tapOrder         = the desired order of the tapped-delay
%                                 line
%              lookDirection    = the desired look direction of the beamfomer
%              micarray         = the microphone signals
% Outputs:      result          = the output of the beamformer


function [result] = frostBeamformer(c, Fs, numMics, d, tapOrder, lookDirection, micarray)

%Paramaters for the filter
fLow = 100;
fHigh = c/(d*2); %High frequency is constrained by the spacing of the array


%==========================================================================
%Find the filter coefficients and initialise tapped-delay weights
%==========================================================================

%Find constraints to be placed on the LMS algorithm
constraintVector = filterCoeffs(fLow,fHigh,Fs,tapOrder);

%Create coefficient vector for tapped delay line
W = zeros(numMics, tapOrder);

%Apply condition Transponse (W)(l)1 = f(l)
for l = 1:tapOrder
    W(:,l) = constraintVector(l)/numMics;
end

%==========================================================================
%Perform FROST Beamforming - Delay appropriate microphone signals
%==========================================================================

%Find the time delay between each microphone for the look direction
timeDelay = abs((d / c) .* sin(lookDirection));
```

```
%Convert the time delay to delay in samples
sampleDelay = round(timeDelay / (1 / Fs));
%Find the maximum delay that will be experienced by a microphone in the
%array
maxDelay = sampleDelay * (numMics - 1);

steered = zeros(numMics, length(micarray));
for x = 1:numMics

    if lookDirection == 0
        %Look direction of 0 degrees
        steered(x,:) = micarray(x,:);

    elseif lookDirection > 0
        %Look direction positive degrees
        delay = (x - 1) * sampleDelay;
        steered(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-delay)];

    else
        %Look direction negative degrees
        delay = maxDelay - ((x - 1) * sampleDelay);
        steered(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-delay)];

    end
end

%==========================================================
%Perform FROST Beamforming
%==========================================================

%Vector to store the result
result = zeros(1,length(steered));
%Matrix to buffer the data to perform the filtering
buffer = zeros(size(W));

%Set the alpha value for the LMS Algorithm
alpha = 0.05;

%Pre-calculating values
ONES = ones(numMics,1); %Vector of ones
OneOnMics = 1/numMics;  %1/numMics

%Matrix of the filter values. Each row is a copy
constraintMatrix = ONES*constraintVector;
bufferDim = size(buffer);
for x = 1:length(steered)

    %Copy a set of steered inputs into the buffer
    buffer = [steered(:,x) buffer(:,1:bufferDim(2)-1)];
    %Multiply buffer by filter coefficients
    temp = buffer .* W;
    %Sum and store the results
    result(x) = sum(sum(temp));


    %Calculate power in the buffer
    Pk = sum(sum(buffer.^2));

    %Calculate step size
    if Pk ~= 0
        mu = alpha / Pk;
    else
        mu = 0;
    end

    qx = sum(OneOnMics * buffer);

    qal = sum(OneOnMics * W);

    %Calculate amount to change the FIR fiter coefficients by
    deltaL = mu*result(x)*(ONES*qx - buffer) - ONES*qal + OneOnMics*
        constraintMatrix;

   W = W + deltaL;

%     fprintf('Sample %d of %d\n', x, length(steered));
end

%Normalise the energy of the output with average energy of the input
ein = 0;
for x = 1:numMics
    ein = ein + sqrt(sum(micarray(x,:).*micarray(x,:))/length(micarray(x,:)));
end
```

```
ein = ein / numMics;

eout = sqrt(sum(result.*result)/length(result));
result = result / eout;
result = result * ein;
```

C.2.3 GSCBeamformer.m

```
% Function:    gsc.m
% function  [result] = GSCBeamformer(c, Fs, numMics, d, JTapOrder, lookDirection, micarray)
% Written By:  Benjamin Cobb - 2006
% Description: Perform GSC beamforming on a set of signals that are
%              defined by the program
% Inputs:      c                    = the speed of sound
%              Fs                   = the sampling frequency
%              numMics              = the number of microphones in the array
%              d                    = the distance between each mic in the
%                                     array
%              tapOrder             = the desired order of the tapped-delay
%                                     line
%              lookDirection        = the desired look direction of the beamfomer
%              micarray             = the microphone signals
% Outputs:     result               = the output of the beamformer

function [result] = GSCBeamformer(c, Fs, numMics, d, JTapOrder, lookDirection, micarray)

%Select the blocking matrix
if numMics == 2
    blockingMatrix = [1 -1];
end
if numMics == 4
    blockingMatrix = [ 1  1 -1 -1;
                       1 -1 -1  1;
                      -1  1 -1  1];
elseif numMics == 6
    blockingMatrix = [ 0  0  1 -1  1 -1;
                       0  1 -1  1 -1  0;
                       1 -1  1 -1  0  0;
                      -1  1 -1  0  0  1;
                       1 -1  0  0  1 -1];
elseif numMics == 8
    blockingMatrix = [ 0  0  0  0  1 -1  1 -1;
                       0  0  0  1 -1  1 -1  0;
                       0  0  1 -1  1 -1  0  0;
                       0  1 -1  1 -1  0  0  0;
                       1 -1  1 -1  0  0  0  0;
                      -1  1 -1  0  0  0  0  1;
                       1 -1  0  0  0  0  1 -1];
end

%Determine paramaters for the fixed filter
fLow = 100;
fHigh = c/(d*2); %High frequency is constrained by the spacing of the array
JTapOrderFixed = 1001; %Order of Fixed Filter

%Find Microphone weights
Wc = hammingWeights(numMics);

%Blocking martix dimensions
blockingMatrixDimension = size(blockingMatrix);


%==========================================================
%Find the Fixed Filter coefficients
%==========================================================

fixedFilterCoeffs = filterCoeffs(fLow,fHigh,Fs,JTapOrderFixed);

%Apply Hamming window to fixed filter coefficients.
fixedFilterCoeffs = fixedFilterCoeffs .* hammingWeights(JTapOrderFixed);

%=====================================================================
%Perform GSC Beamforming - Delay appropriate microphone signals
%=====================================================================

%Find the time delay between each microphone for the look direction
timeDelay = abs((d / c) .* sin(lookDirection));
```

```
%Convert the time delay to delay in samples
sampleDelay = round(timeDelay / (1 / Fs));
%Find the maximum delay that will be experienced by a microphone in the
%array
maxDelay = sampleDelay * (numMics - 1);

steered = zeros(numMics, length(micarray));
for x = 1:numMics

    if lookDirection == 0
        %Look direction of 0 degrees
        steered(x,:) = micarray(x,:);

    elseif lookDirection > 0
        %Look direction positive degrees
        delay = (x - 1) * sampleDelay;
        steered(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-delay)];

    else
        %Look direction negative degrees
        delay = maxDelay - ((x - 1) * sampleDelay);
        steered(x,:) = [zeros(1,delay) micarray(x,1:length(micarray)-delay)];

    end
end

%==========================================================
%Upper Path
%==========================================================

%Add each individual microphone together
steeredSum = zeros(1,length(steered));
for x = 1:numMics
    steeredSum = steeredSum + Wc(x) * steered(x,:);
end

%Clear unneeded variables
clear H i wHigh wLow nhigh nlow w dw h sampleDelay timeDelay

% Filter the signal
buffer = zeros(1, JTapOrderFixed);
filteredSignal = zeros(size(steeredSum));
for x = 1:length(steeredSum)
    buffer = [steeredSum(x) buffer(1:length(buffer)-1)];    %Copy a set of
        steered inputs into the buffer
    temp = buffer.*fixedFilterCoeffs;
    filteredSignal(x) = sum(temp);
end

%==========================================================
%Lower Path
%==========================================================


%Apply Blocking Matrix
blockedSignal = zeros(blockingMatrixDimension(1),length(steeredSum));
for x = 1:length(steeredSum)
    blockedSignal(:,x) = blockingMatrix*steered(:,x);
%     fprintf('Blocking sample %d of %d\n', x, length(steered));
end

buffer = zeros(blockingMatrixDimension(1),JTapOrder);
result = zeros(1, length(blockedSignal));
alpha = 0.05;

A = zeros(blockingMatrixDimension(1), JTapOrder);
bufferDim = size(buffer);

%Apply multiple-input canceller to blocked signal.
for x = 1:length(result)

    buffer = [blockedSignal(:,x) buffer(:,1:bufferDim(2)-1)];    %Copy a set of
        blocked inputs into the buffer
    temp = buffer.*A;
    result(x) = filteredSignal(x) - sum(sum(temp));

    %Now Adapt
    Pk = sum(sum(buffer.^2));

    %Calculate mu
    if Pk ~= 0
        mu = alpha / Pk;
```

```
    else
        mu = 0;
    end

    %Adjust the tapped-delay weights to their new values
    A = A + mu*result(x)*buffer;

end

%Normalise the energy of the output with average energy of the input
ein = 0;
for x = 1:numMics
    ein = ein + sqrt(sum(micarray(x,:).*micarray(x,:))/length(micarray(x,:)));
end
ein = ein / numMics;

eout = sqrt(sum(result.*result)/length(result));
result = result / eout;
result = result * ein;
```

C.3 Helper Functions

C.3.1 filterCoeffs.m

```
% Function:     filterCoeffs.m
% [weights] = hammingWeights(lengthOfWindow)
% Written By:   Benjamin Cobb - 2006
% Description:  Calculate FIR filter coefficients
% Inputs:       fLow       = the lowest end of the pass band
%               hHigh      = the highest end of the pass band
%               Fs         = the sampling frequency
%               JTapOrder  = the desired length of the filter
% Outputs:      coeffs     = the calculated coefficients of the filter

function [coeffs] = filterCoeffs(fLow, fHigh, Fs, JTapOrder)

dw = pi/400;
w = -pi:dw:pi;

H = zeros(size(w));
wLow = pi/((Fs/2)/fLow);
wHigh = pi/((Fs/2)/fHigh);

% 0 to pi
i = find( (w >= wLow) & (w <= wHigh) );
H(i) = ones(size(i));

% -pi to 0
i = find( (w <= -wLow) & (w >= -wHigh) );
H(i) = ones(size(i));

nlow = - (JTapOrder-1)/2;
nhigh = (JTapOrder-1)/2;

K = 1/(2*pi);
for n = nlow:nhigh
    h(n-nlow+1) = K*sum(H.*exp(j*w*n)*dw);
end

%Cater for rounding errors
coeffs = real(h);
```

C.3.2 hammingWeights.m

```
% Function:     hammingWeigths.m
% [weights] = hammingWeights(lengthOfWindow)
% Written By:   Benjamin Cobb - 2006
% Description:  Calculate a Hamming window of lenght lengthOfWindow
% Inputs:       lenghtOfWindow = the length of the Hamming window required
% Outputs:      weights        = the Hamming window coefficients

function [weights] = hammingWeights(lengthOfWindow)

n = 0:1:(lengthOfWindow-1);
```

```
weights = 0.54 - 0.46 * cos(2*pi*n/(lengthOfWindow-1));
```

C.4 Other Functions

C.4.1 delayAndSumResponse.m

```
% Procedure: test.m
% Written By: Benjamin Cobb - 2006
% Description: Function that calculates the Delay and Sum beamformer
% response.

clc;                %Clear the command window
clear;              %Clear all variables
close all;          %Close all open figures

%Angle in which to point the beamformer
lookDirection = 60;
%Convert the look direction to radians
lookDirection = (pi*lookDirection)/180;

%Apply Hamming weights
hammingWeightEnabled = 0;

%Parameters
d = 0.055;
numMics = 2;
c = 344;

%Find Microphone weights
if hammingWeightEnabled
    amp = hammingWeights(N);
else
    amp = ones(1,numMics) ./ numMics;
end

%Find response of Delay and Sum beamformer
[X, Y] = meshgrid(300:3000, -90:90);
n = 1:numMics;
results = zeros(181, 2701);
for freq = 300:3000

    fprintf('Calculating Frequency Response for frequency %d Hz\n', freq);
    wavlen = c / freq;

    %Find response of at frequency at angle
    for angle = -90:90
        thetaR = (pi*angle)/180;

        temp = amp .* exp(j*2*pi*(sin(lookDirection)/wavlen)*(n-1)*d) .* (exp(j * 2 * pi * (sin(thetaR)/wavlen) * (n-1) * d));

        results(angle+91, freq-299) = sum(temp);

    end
end

%Normalise the response to one
results = abs(results);

results = results./max(max(results));

%Plot the Response
figure;
set(gca, 'fontsize', 12);
mesh(X,Y, results);
xlabel('Frequency (Hz)');
ylabel('Angle (Degrees)');
zlabel('Spatial Response');
```