Real-Time Implementation of a Dual Microphone Beamformer

A thesis presented in partial
fulfilment of the requirements for the degree of
Master of Engineering
in
Computer Systems

at Massey University,
Albany,
New Zealand.

Vaitheki Yoganathan

2005

To Whom It May Concern:

This thesis is submitted for the degree of Master of Engineering at Massey University and it is not previously submitted to this or any other institution for any degree, diploma or other qualification. It is the result of my independent work and the information obtained from the published or unpublished source of others have been acknowledged in the text and a list of references is given.

Vaitheki Yoganathan

Vaitheki Yoganathan

Date: 10-03-2005

Abstract

The main objective of this project is to develop a microphone array system, which captures the speech signal for a speech related application. This system should allow the user to move freely and acquire the speech from adverse acoustic environments. The most important problem when the distance between the speaker and the microphone increases is that often the quality of the speech signal is degraded by background noise and reverberation. As a result, the speech related applications fails to perform well under these circumstances. This unwanted noise components present in the acquired signal have to be removed in order to improve the performance of these applications.

This thesis contains the development of a dual microphone beamformer in a Digital Signal Processor (DSP). The development kit used in this project is the Texas Instruments TMS320C6711 DSP Starter Kit (DSK). The switched Griffiths-Jim beamformer was selected as the algorithm to be implemented in the DSK. Van Compernolle developed this algorithm in 1990 by modifying the Griffiths-Jim beamformer structure. This beamformer algorithm is used to improve the quality of the desired speech signal by reducing the background noise. This algorithm requires at-least two input channels to obtain the spatial characteristics of the acquired signal. Therefore, the PCM3003 audio daughter card is used to access the two microphone signals.

The software implementation of the switched Griffiths-Jim beamformer algorithm has two main stages. The first stage is to identify the presence of speech in the acquired signal. A simple Voice Activity Detector (VAD) based on the energy of the acquired

signal is used to distinguish between the wanted speech signal and the unwanted noise signals. The second stage is the adaptive beamformer, which uses the results obtained from the VAD algorithm to reduce the background noise.

The adaptive beamformer consists of two adaptive filters based on the Normalised Least Mean Squares (NLMS) algorithm. The first filter behaves like a beam-steering filter and it's only updated during the presence of speech and noise signal. The second filter behaves like an Adaptive Noise Canceller (ANC) and it is only updated when a noise alone period is present. The VAD algorithm controls the updating process of these NLMS filters and only one of these filters is updated at any given time.

This algorithm was successfully implemented in the chosen DSK using the Code Composer Studio (CCS) software. This implementation is tested in real-time using a speech recognition system. This system is programmed in Visual Basic software using the Microsoft Speech SDK components. This dual microphone system allows the user to move around freely and acquire the desired speech signal. The results show a reasonable amount of enhancement in the output signal, and a significant improvement in the ease of using the speech recognition system is achieved.

Acknowledgments

I would like to express my sincere thanks to my mentor Dr. Tom Moir for his guidance during this research work. Without his invaluable advice, help and suggestions this work would not have been possible. I would also like to thank the late Dr R. Chassaing for his assistance.

Finally I would also like to thank my family and friends for being there and supporting me throughout my studies.

Table of Contents

List of Figures

List of Abbreviations

ADC	Analog-to-Digital Converter
ANC	Adaptive Noise Canceller
C6711	TMS320C6711
CCS	Code Composer Studio
COFF	Common Object File Format
DAC	Digital-to-Analog Converter
DSK	Digital Signal Processor Starter Kit
DSP	Digital Signal Processor
GCC	Generalised Cross Correlation
GSC	Generalised Sidelobe Canceller
HOIT	Home Oriented Informatics and Telematics
LMS	Least Mean Squares
MFLOPS	Million of Floating Point Operations Per Second
ML	Maximum Likelihood
MSC	Magnitude Squared Coherence
NLMS	Normalized LMS
ROM	Read Only Memory
SDK	Software Development Kit
SNR	Signal-to-Noise Ratio
SDRAM	Synchronous Dynamic Random Access Memory
TASI	Time Assigned Speech Interpolation
TDOA	Time Difference Of Arrival
TI	Texas Instrument
VAD	Voice Activity Detector

1. Introduction

During the past 20 or more years, speech acquisition in adverse acoustic environments have received considerable attention due to the increased need for hands-free and voice controlled applications (Krasny & Oraintara, 2002; Martin, 1976). The main difficulty when acquiring speech from this type of environment is that often speech is corrupted by background noise and reverberation. Consequently, this corrupted speech complicates and degrades the performance of these speech related applications.

It is important to improve the quality of the acquired speech signal in order to improve the performance of these applications. This improvement is achieved by suppressing the unwanted noise components present in the acquired signal (without harming the speech signal). This background noise could consist of several components propagating from different sources such as computer fan, engine noise, air conditioner, audio equipments, or competing speech.

Noise reduction in the corrupted speech signal remains an important problem in many speech related applications. Some of these applications include:

- Videoconference and Teleconferencing (Elko, 1996)
- Hands-free telephony (Bouquin-Jeannès, Faucon, & Ayad, 1996; Campbell, 1999)
- Mobile telephony in moving vehicle environment (Cho & Krishnamurthy, 2003; Ezzaidi, Bourmeyster, & Rouat, 1997; Hussain, Campbell, & Moir, 1997; Lin, Lin, & Wu, 2002),

- Hearing aids (Greenberg, Desloge, & Zurek, 2003; Ventura, 1989; Wang et al., 1996; Widrow, 2001; Widrow & Luo, 2003; Wilson, 2003)
- Speech recognition (Chien & Lai, 2004; Moore & McCowan 2003; Van Compernolle, 1992a)
- Speech coding (Collura, 1999; Li & Hoffman, 1999)
- Robotics (Choi, Kong, Kim, & Bang, 2003; Mumolo, Nolich, & Vercelli, 2003; Seabra Lopes & Teixeira, 2000; Valin, Michaud, Rouat, & Letourneau, 2003)

Most of these applications can be categorised as either human-to-human communication (such as communication over the traditional telephone or data networks), or human-to-machine communication (such as communications with robots and computers). In human communication, the most natural and quickest form of high-level language is speech. Over the years, many people have been trying to extend this ability towards human-to-machine communication. However, this procedure has gained some progress only in the recently years. A more detailed discussion on using voice as input for human-to-machine communication can be found in the following literatures (Choen & Oviatt, 1995; Roe & Wilpon, 1994).

When a person's hands are busy and/or are unable to use them due to medical reasons, they could use speech to control the appliances. This is another attractive motivation to use speech communication as an interface to control appliances. For example, it is much safer for a driver to use his voice to dial the phone, rather than dial the phone by hand while driving. Recently, this application has received considerable attention due to the increased number of road accidents happened because the driver was preoccupied with the hand held devices. By using the hands-free technology, these incidents could

have been prevented. However, when acquiring speech from these adverse acoustic environments, the speech signal is more likely to be distorted by noise. As a result, under these circumstances the speech related applications fails to perform as expected.

One effective solution to improve the quality of the received speech signal is to use a microphone near the user, which requires the user to always wear or hold the microphone. However, using wearable microphones is impractical and is not desirable in many applications. Some examples of these situations include fast-food drive through outlets, un-manned service stations and voice pickup in large rooms (Flanagan, Johnston, Zahn, & Elko, 1985).

Directive microphones have been used to overcome this problem of wearable microphones. However, directive microphones capture the desirable speech as well as the background noise. As a result, the quality of the desired speech is degraded. Therefore, it does not achieve our objectives in adverse acoustic environments. To overcome these problems speech enhancement techniques can be used with the directive microphones to achieve a better performance.

Speech enhancement techniques can be divided into two main categories depending on the number of microphones used in the algorithm, they are single microphone system (Cole, Moody, & Sridharan, 1993; Scalart & Filho, 1996) and multi-microphone (microphone array) system (Cao & Sridharan, 1993; Van Compernolle & Van Gerven, 1995; Yan, Du, Wei, & Zeng, 2003). An overview of the available techniques for speech enhancement using single and multi microphone algorithms are given in the

following literatures (Lim, 1983; Ortega-Garcia & Gonzalez-Rodriguez, 1996; Van Compernolle, 1992b).

Single microphone systems have some limitations such as the interference has to be stationary and the input signal-to-noise ratio (SNR) over most of the frequency range has to be positive. (SNR is the ratio between the power of signal and the power of noise, and it is usually given in dB.) On the other hand, microphone array systems can handle non-stationary or very strong interference signals. By using more than one microphone, we can also obtain the spatial information such as the location of the acquired signal. Due to these reasons, this thesis will be focusing on the microphone array system to improve the acquired signal.

The microphone array techniques are a potential replacement for the wearable and the directive microphones to use in the speech related applications. This idea of using the microphone array system to improve the desired speech signal isn't new to this field. However, more affordable solutions became available only after the availability of the inexpensive digital signal processors (DSPs) during the 1980's. In the past decade, many literatures have been published on microphone array techniques and their applications by many authors, such as (Affes & Grenier, 1997; Brandstein & Ward, 2001; Campbell, 1999; Farrell, Mammone, & Flanagan, 1992; Fischer & Simmer, 1996; Kaneda & Ohga, 1986; McCowan, 2001a, 2001b; Silverman, 1987; Van Compernolle, 1990).

The most important objective of a microphone array is to provide a high quality version of the desired speech signal in an adverse acoustic environment. The microphone array

achieves this via beamforming techniques, which are designed to reduce the level of background noise signals, while minimising distortion to speech. A beamformer does spatial filtering by separating the desired signal and the interference signals that originate from different directions but have the same temporal frequency band.

Finding an optimal beamformer algorithm for a particular application depends highly on the available hardware and the computational resources. The Texas Instruments (TIs) TMS320C6711 DSP Starter Kit (DSK) is the hardware platform available for this project. Since the DSK supports only one input channel, the PCM3003 daughter card is used to obtain the two input signals received at the microphones. When choosing an optimal algorithm, more priority is given to less computational cost algorithms; this is due to the limited computational resources in the DSK.

Characteristics of the target application environment such as the type and the level of noise and reverberation are also equally important when choosing an algorithm for a particular application. At present, there is a broad range of speech related applications that require enhancement of speech. The interferences that degrade the quality of the speech signal for each of these applications may differ from application to application. For example, indoor applications may have interference coming from audio equipment, computer fan, etc and outside applications may have interference coming from vehicle engine, birds, etc. Therefore, it is impractical to suggest one beamformer algorithm that is generally applicable to all speech related applications.

This thesis will focus on a specific application in order to choose an optimal algorithm. However, with simple necessary modifications this implementation can work for any

speech related applications. One possible target application for this microphone array beamformer is the Massey University Smart House (Diegel et al., 2005). This smart house is designed for the disable or elderly people to give them independence, quality of life, and the safety they require.

One of the smart technologies in this house is the smart management system. This is a computer based software program called Jeeves, which is programmed to works as a virtual butler for the house. The basic idea behind this system is to interact with the occupants of the smart house, in order to receive commands to control the household appliances. A more detailed discussion about the functions of the smart house can be found in the following literature (Diegel et al., 2005), which is to be presented at the Home Oriented Informatics and Telematics (HOIT) conference in 2005 (copy of this paper is given in the Appendix D).

In order to make it easy as possible for the occupants of the smart house to interact with Jeeves, the medium of communication must be as natural as possible. A natural interface should allow the user to interact with Jeeves directly using their voice to request commands to perform some action. Therefore, a speech recognition system is required in order to analyse the speech and extract the necessary commands to control the household appliances. Commercially available speech recognition software (Visual Basic with Microsoft Speech SDK) is used in this project to interpret the human speech.

For a speech recognition system to work efficiently, it typically requires a SNR of greater than 20dB. In an ideal house, there will be a considerable amount of background noise propagating from different sources such as computer fan, radio, TV,

and other talkers. The speech signal acquired in this environment will be distorted by these background noises. This could lead to poor performance of the speech recognition system.

A simple solution is to use a wearable microphone to acquire the speech from the user. Since the smart house is designed for the disable and elderly people, it will not be convenient for the occupants of the house to use a headset every time they want to issue a command. Moreover, majority of the people wouldn't be fond of wearing a microphone in their house at all times. Therefore, a microphone array system (Chien, Lai, & Lai, 2001) that allows the users to move around freely and interact easily with Jeeves by using their voice is required. This thesis discusses the real-time implementation of such a microphone array beamformer on a DSP.

This thesis is organised as follows: *Chapter 2* discusses the historical context behind the noise reduction techniques and the speech detection algorithms. It gives an overview of the previous approaches done on this area and explains in-depth about the chosen algorithm for this project. The switched Griffiths-Jim beamformer algorithm was selected as the algorithm to be implemented on the TMS320C6711 DSK. Hardware and software implementation of the chosen algorithm in real-time is described in *Chapter 3*. This completed system is tested, and the results can be found in *Chapter 4*. The conclusions and future work are given in *Chapter 5*.

2. Historical Context

This chapter briefly describes the historical context behind the noise reduction techniques and the speech detection algorithms. In addition, the chosen beamformer algorithm to be implemented in the DSK is explained in-depth.

2.1. Noise Reduction Techniques

The problem with acquiring speech from an adverse acoustic environment is that the desired speech signal is often corrupted by background noise and the performance of the speech related application is degraded. The basic idea behind suppression of the background noise has been around for sometimes. However, only recently there is more work is done on research and development; this is due to the availability of low-cost and more rapid hardware platforms. The fundamental technique behind improving the distorted speech is to use an adaptive filter to suppress the noise components, while leaving the desired speech signal unchanged.

Adaptive noise canceller (ANC) scheme was first proposed by Bernard Widrow in 1975. It is a noise reduction techinique, which attempts to use multiple signal sources to eliminate the noise components from the aquired signal (Armbruster, Czernach, & Vary, 1986; Feng, Shi, & Huang, 1993; Sambur, 1978; Widrow, 1975). In order for this technique to work it requires at least two-microphones. A conceptual diagram of the adaptive noise cancellation technique is given in Figure 2.1.

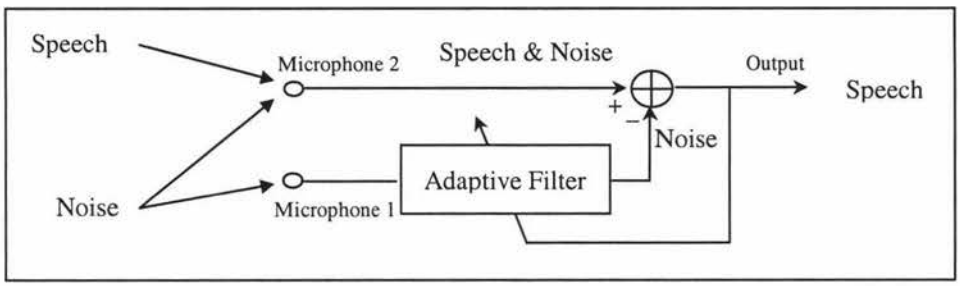


Figure 2.1. Adaptive noise canceller

As shown in the figure, the first microphone is positioned near the noise source and the second microphone is positioned as such to pickup the desired speech signal as well as the noise signals. In an ideal situation, subtracting the second microphone signal from the first microphone signal will eliminate the noise components while leaving the desired speech signal unchanged. In addition, an adaptive filter algorithm is used to delay the signal approximately in order to get the optimum noise reduction. Hence, it is used to minimize the error between a desired signal and the received array output. More detailed discussion on the adaptive filter algorithms is given in the Subsection 2.1.2.

This technique seems to work well in applications where the noise sources have low frequency, such as engine noise. Acquiring speech inside a jet aircraft is a typical example application for this adaptive noise cancellation algorithm (Kang & Fransen, 1987). In order to remove all the noise components in the subtraction process, this technique requires the noise entering the two microphones to be coherent. However, experiments have shown that in many applications the noise entering the two microphones is hardly coherent (thus, this technique fails to reduce the noise).

In order for the noise signal in both channels to be coherent, both microphones have to be kept closer to each other. However, when they are kept close together, it makes it

impossible for one microphone to acquire the speech signal and the other microphone not to. In the subtraction process this speech signal will be removed together with the noise signal, which is the main draw back of this technique. In order for this technique to work, the two microphones have to be kept far enough so only one of them can acquire the desired speech signal, and they also have to be kept close enough, so the acquired noise signals in both microphones are same (coherent).

Noise reduction with microphone array technique is another possibility to enhance the desired speech. A microphone array system is constructed with a number of microphones distributed in a certain area to capture the desired signal. These signals are then processed to suppress the unwanted noise signals. The microphone array system makes use of the beamforming techniques to reduce the effects of the adverse acoustic environment on the speech signal (McCowan, 2001a). A substantial gain in performance can be obtained by using a microphone array beamformer; this is due to the spatial filtering capabilities. An overview of the state of the art microphone array systems can be found in the following literatures (Krim & Viberg, 1996; Mucci, 1984; Van Compernolle & Van Gerven, 1995).

Typical use of beamforming arises from many different backgrounds such as:

- Sonar (Istepanian & Stojanovic, 2001; McHugh, Shaw, & Taylor, 1994)
- Radar (Wirth, 2001; Yongjian, Genmiao, & Shouhong, 2001)
- Communications (Blogh & Hanzo, 2002; Ezzaidi et al., 1997)
- Smart Antennas (Kluwer, 2001; Rappaport, 1998; Tsoulos, 2001)
- Imaging (McHugh et al., 1994)
- Radio astronomy
- Geophysical and Biomedical applications

Although the basic idea of beamforming came from these backgrounds, acoustic beamforming requires considerably different solutions. The following literatures have provided an overview of beamforming techniques from a signal processing perspective (Cox, Zeskind, & Owen, 1987; Fischer & Simmer, 1996; Gannot, Burshtein, & Weinstein, 2001; Van Compernolle & Van Gerven, 1995; Van Veen & Buckley, 1988).

Depending on how the filter weights are chosen, beamformer algorithms can be classified as either fixed beamforming or adaptive beamforming. Fixed beamformer is designed to focus on a targeted direction independently of the interfering signals. This type of system is very robust and requires minimal processing power. However, they need many microphones to achieve high directivity and good noise reduction. Conversely, adaptive beamformer is designed to discard the interfering signals by introducing nulls in the direction of their arrivals. This type of system is able to attain a high noise reduction with a small number of microphones, but at a cost of high processing power.

The simplest method of all microphone array beamforming is the delay-and-sum beamformer (Flanagan et al., 1985). Figure 2.2 illustrates a conceptual diagram of the delay-and-sum beamformer for N number of microphones.

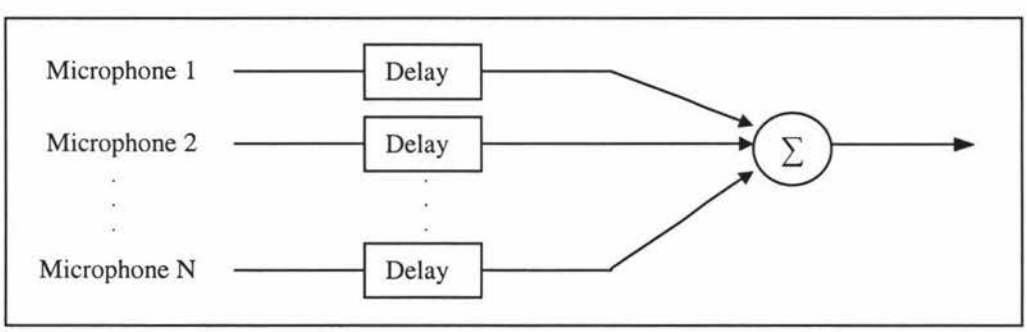


Figure 2.2. Delay-and-sum beamformer

In order to align all the received signals, each microphone's signal is adjusted by a time delay proportional to the distance between a desired source and the microphone. Then all these adjusted signals are added together to obtain a single output signal. The basic idea behind this summing process is that the desired speech signal will have the same phase so they will add together and the interfering signals will have a different phase so they will not add together. This is done by assuming that the noise signal will not be coming from the exact same position as the desired signal, thus the noise signal is not coherent.

The total speech power in the resulting output signal will be multiplied by the number of microphones in the array, while the total noise power in the resulting output signal will remain about the same as the noise power of one microphone. As a result, the SNR will be increased. Major disadvantages of using the delay-and-sum beamformer are that it requires many microphones to obtain a considerable amount of improvement in the SNR, and it only enhances the signal in the direction to which the array is currently steered (it does not reduce the interference itself).

An alternative method to achieve more improved performance is to multiply the received signals with different gain factors (or weights) before summing the received signals. An example of this type of method is the Frost beamformer (Frost, 1972), and it is also known as the linearly constrained minimum variance algorithm. A study done by Raykar shows that using beamformer improves the SNR of the output signal when compared to just using the directive microphone. It also confirms that frost beamformer performs better than the simple delay-and-sum beamformer (Raykar, 2001). Figure 2.3 illustrates a conceptual diagram of the Frost beamformer for N number of microphones.

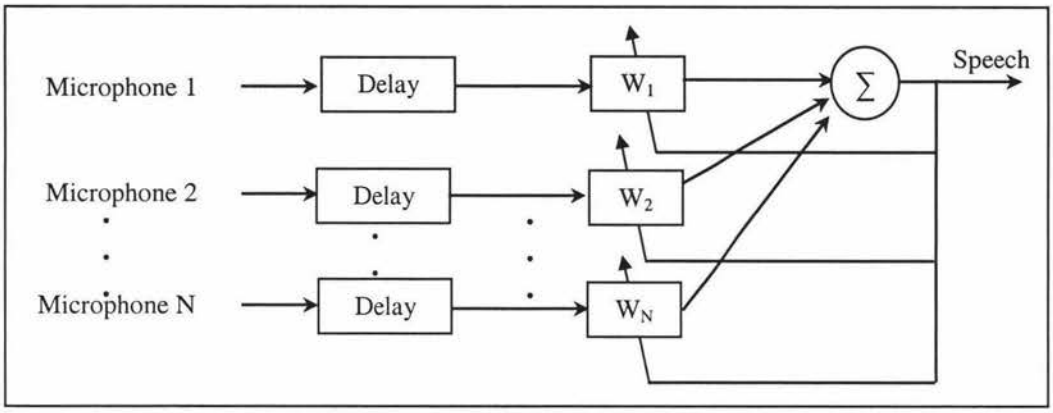


Figure 2.3. Frost beamformer

Frost beamformer uses an adaptive filter algorithm to adjust the weight vector's (W) continuously to minimise the noise, while maintaining a chosen frequency response in the look direction (direction of the desired speech signal). This beamformer algorithm is found to be effective in the presence of strong interferers, provided that the interferers are uncorrelated with the desired source.

An improved version of this structure has been proposed by Griffiths and Jim in the 1980's (Griffiths & Jim, 1982). Their design is known as Generalised Sidelobe Canceller (GSC) or Griffiths-Jim algorithm. Figure 2.4 shows the basic structure of the Griffiths-Jim beamformer for straight-ahead target using two microphone inputs. If the target signal is not straight-ahead then a beam-steering algorithm can be used before this algorithm to focus the microphone in the direction of the desired target signal.

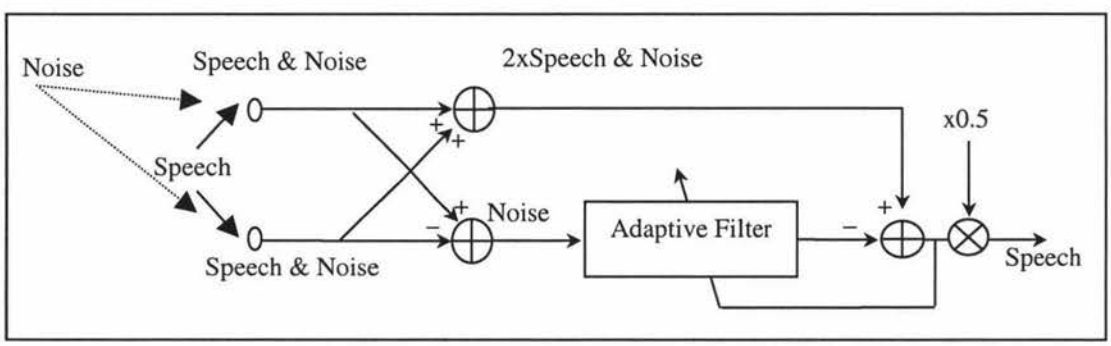


Figure 2.4. Two-channel Griffiths-Jim beamformer

The Griffiths-Jim beamformer operates as an ANC with a pre-processor that does the addition and subtraction of the received microphone signals. This algorithm works well as long as the desired speech signal arriving at the two microphones are aligned. Thus, the speech signals received at the microphones have the same phase and the noise signals have different phases. Therefore, in the addition process, the speech will be added to produce twice the speech signal and noise; and in the subtraction process, the speech will be cancelled out to produce noise alone signal. The resulting signals from the pre-processor, is used as the primary and reference signal for the ANC.

One of the problems raised in the ANC structure is that the noise signal in both input channels was not coherent in order for them to cancel out in the adaptive processes. Conversely, the Griffiths-Jim beamformer structure is designed in such a way to solve this problem. This algorithm is sensitive to target signal leakage and cancellation in the presence of steering vector errors and reverberations (Bitzer, Simmer, & Kammeyer, 1999). These steering vector errors are caused by errors in microphone positions, microphone gains, reverberations, and target direction. These problems have been noticed by some researchers and new improved beamformer algorithms have been proposed in an attempt to solve them (Claesson, Nordholm, & Bengtsson, 1991; Cox et al., 1987; Er & Ng, 1994; Hoshuyama, Sugiyama, & Hirano, 1999; Van Compernolle, 1992a; Yongjian et al., 2001).

One of the improved version of the Griffiths-Jim beamformer is proposed by Hoshuyama et al, which attempts to solve these problems (Hoshuyama et al., 1999). This new robust adaptive beamformer uses an adaptive blocking matrix consists of coefficient constrained adaptive filters in the GSC structure, and a multiple input

canceller using norm-constrained adaptive filters. This technique did prove to cancel interferences by over 30 dB. However, its performance deteriorates in the presence of coloured low-pass interference signals.

Van Compernolle proposed a new speech beamformer called switched Griffiths-Jim beamformer, which attempts to eliminate problems raised in Griffiths-Jim beamformer algorithm (Van Compernolle & Van Gerven, 1995). Van Compernolle showed that signal cancellation can be reduced somewhat by adapting the ANC's filter parameters only during the noise alone regions (when no speech is present in the received signals). Most success has been obtained by this method compared to others. Therefore, this algorithm is selected for the real-time implementation in this project and it will be discussed in the following subsection.

2.1.1. Switched Griffiths-Jim Beamformer

Switched Griffiths-Jim beamformer technique was discovered by Van Compernolle in 1990 (Van Compernolle, 1990) and this is a modified version of the famous Griffiths-Jim beamformer (Griffiths & Jim, 1982). Figure 2.5 illustrates an example conceptual diagram of this technique for two microphone input signals. This design can be easily modified to use more microphones as required. This beamformer algorithm makes use of two adaptive filters, the first filter behaves like a beam-steering filter and the second filter behaves like an ANC. Vectors V and W are adaptive filter tap weights with length of L taps; more information on how it is updated is explained later in the Subsection 2.1.2.

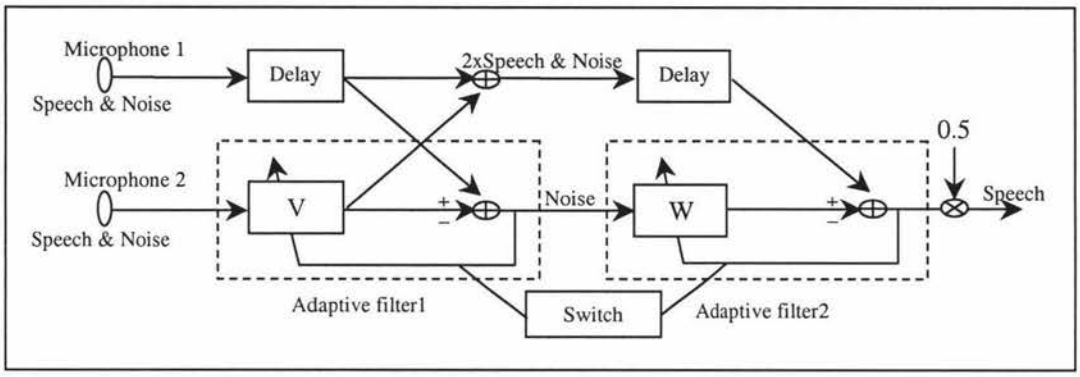


Figure 2.5. Switched Griffiths-Jim beamformer

The purpose of the beam-steering filter is to obtain an optimal phase alignment between the two input channels. This filter creates an adaptive look direction and cues in on the desired speech source. Therefore, this filter (V weight vector) is updated only during the presence of a speech signal. Due to multi-path and strong signal conditions, the reference signal for the ANC is hardly speech free. In order to get around the target signal leakage problem in the Griffiths-Jim beamformer structure, the second adaptive filter (W weight vector) is updated only during the noise alone segment. The resulting signal from the ANC is the estimated clean speech signal.

A lower convergence rate is used for the first filter, because it is not necessary to track all small head movements. Conversely, a faster convergence rate is used for the second filter, since it needs to respond quickly to any changes in the environment. A switch based on speech detection algorithm is used to control these filters, which eliminates the requirement for speech free noise references. The result from the speech detection algorithm allows only one filter to update at any given time. The selection of speech detection algorithms is explained in detail in Section 2.2. In order to prevent the output speech signal from being distorted, this beamformer algorithm requires a reliable speech detector algorithm.

For the real-time implementation of this algorithm, there is a trade off between the number of channel and the taps per channel; this is due to the limited computational power. An uncausal solution would be caused if the microphones were happened to be in the wrong position with respect to each other. In order to prevent this time delay is introduced in one of the input channel to provide a physical reliability.

This beamformer algorithm has proven to improve the overall performance of the system by 5dB for a competing speaker, by 8dB for a wide band semi-stationary noise, and by 20dB for a narrowband interference (Van Compernolle, Van Gerven, Broos, & Weynants, 1991). The next subsection will briefly discuss some of the adaptive filter algorithms that can be used for this adaptive beamformer.

2.1.2. Adaptive Filters

Adaptive filters are used when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters (Simon Haykin, 2002; Kuo, Ranganathan, Gupta, & Chen, 1988). An adaptive filter attempts to find an optimum set of filter parameters based on the time-varying input signals by continually changing their parameters in order to meet a performance requirement. The characteristics of the adaptive filter makes it attractive for signal processing and control applications (Diniz, 1997; Honig & Messerschmitt, 1984). Some applications that use the adaptive filters are:

- Adaptive channel equalization - data transmission through distorted channels

- Adaptive noise cancellation - clean up weak signals buried in heavy interference in medical and other applications (Abutaleb, 1988; Sambur, 1978)
- Adaptive directional antennas
- Echo cancellation in voice and data communications
- Sinusoidal enhancement
- Spectrum analysis
- Coding of speech adaptive beamforming (Collura, 1999; Li & Hoffman, 1999)

Adaptive filters are typically used in four basic configurations to solve problems in these applications (Jenkins, Hull, Strait, Schnaufer, & Li, 1996). They are system identification configuration, adaptive noise cancelling configuration, adaptive linear prediction configuration, and inverse system configuration. Only adaptive noise cancelling configuration will be discussed here, since others are not relevant to this application. Figure 2.6 shows a basic structure of an ANC configuration for the adaptive filter algorithm.

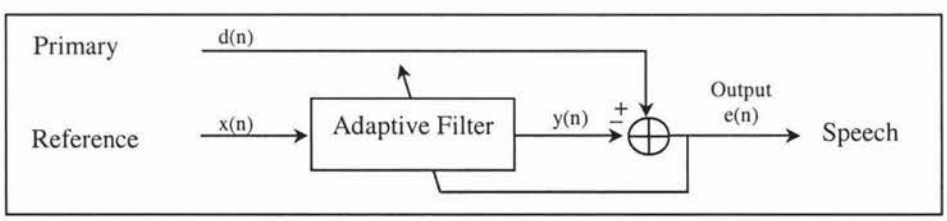


Figure 2.6. Adaptive filter structure

This structure has two inputs [primary input d(n) and reference input x(n)] and one output e(n). Primary input consists of speech and noise signal, and the reference input consists of noise alone signal. The noise present in both of these inputs must be correlated and the speech present in the primary input must be uncorrelated with this noise signal. This structure uses the reference input to reduce the effect of noise in the primary input.

The output error signal measures the difference between the output of the adaptive filter and the primary input, i.e. . $e(n) = d(n) - y(n)$

The filter output is given by: $y(n) = x^T(n)\ w(n)$

where w(n) is the time varying vector of filter coefficients (tap weights), x(n) is the column vector, and superscript "T" denotes vector transpose.

An adaptive algorithm is required in order to adjust the values of tap weights $(w(n))$ in such a way that the filter output $(y(n))$ approximates the primary input. An adaptive algorithm starts with some predefined set of initial conditions, and then updates the tap weights to converge to an optimal result. Initially the tap weight and column vectors are assigned to zero ($x(0) = w(0) = [0...0]^T$). However, if rough values of the tap weights are known then these values can be used.

Widrow and Hoff were one of the first researchers to investigate the adaptive filters in the late 1950s (Widrow, 1975; Widrow & Hoff, 1960). Their work resulted in the famous Least Mean Square (LMS) algorithm (S. Haykin & Widrow, 2002). The LMS algorithm belongs to the family of stochastic gradient algorithms. This algorithm automatically adapts the tap weights of the transversal filter, and it drives its tap weight parameters to values corresponding to minimum mean squared error between the filter output and primary input. The complexity of this algorithm is very low (order of complexity is $O[N]$), and its results are satisfying in many cases.

This algorithm uses the steepest-descent based technique to recursively compute the values of the weight vectors and update them. The following recursive equation is used to update the tap weight vector $(w(n))$ of the LMS filter, computed at iteration (time-step) $n+1$:

$$w(n+1) = w(n) + 2\mu\, e(n)x(n)$$

The term μ is a step-size, which controls the amount of gradient information used to update each coefficient. The value of μ directly affects how quickly the adaptive filter will converge toward the unknown system. If μ is very small, the value of the filter coefficients changes at each update only by a small amount (hence the filter converges slowly). With a large μ, more gradient information is included in each update, and the filter converges more quickly. However, when the step-size is too large, the coefficients may change too quickly and the filter will diverge (the system will go unstable).

This algorithm is simple to implement, and it gives a robust performance against different signal conditions when compared with other adaptive filter algorithms. However, it suffers from a slow convergence rate and causes an unstable system when the convergence rate is too high. Many variations on the LMS coefficient update equation has been proposed in the literature to reduce or eliminate the dependence of the convergence rate of the LMS algorithm (Feng et al., 1993; Orgren, Dasgupta, Rohrs, & Malik, 1991).

Most common solution to this problem is the Normalised LMS (NLMS) algorithm (Yassa, 1987), which utilizes a variable convergence factor that minimizes the instantaneous error. The NLMS algorithm usually converges much faster than the conventional LMS algorithm for both uncorrelated and correlated input signal (An,

Brown, & Harris, 1995). The following equation is used to update the NLMS coefficients:

$$w(n+1) = w(n) + \frac{\mu_n}{\gamma + x^T(n)x(n)} e(n)x(n)$$

In order to lower the influence of the input signal amplitude on the gradient noise, the step-size (μ_n) is divided by the variance of the input signal. The step-size is chosen to be in the range of 0 and 2. In order to prevent situations when division by zero occurs, a small positive constant called gamma (γ) is introduced in the denominator. One of the advantages of using NLMS algorithm is that its convergence rate is insensitive to the power level of the input signal; this is due to the normalization of the step-size.

The switched Griffiths-Jim beamformer algorithm (Subsection 2.1.1) make use of two adaptive filters (Van Compernolle, 1990). Since the LMS algorithm has some drawbacks with stability and selection of the step-size, the NLMS is used in this beamformer algorithm. First NLMS filter is updated during a speech signal and the second NLMS filter is updated during the noise signal. The first one behaves as a beam-steering filter to align both input signals and the second one behaves as an adaptive filter to reduce the noise signal. A voice activity detection algorithm is used to update these two filters and only one of them is updated at any given time.

The Voice Activity Detector (VAD) is used to analyze the acquired signal to determine the presence of speech. The corresponding NLMS filter is updated depending on the result obtained from this VAD. In order to prevent any cancellation of the speech signal, this beamformer algorithm requires a reliable VAD. The possible VAD algorithms that can be used with the switched Griffiths-Jim beamformer algorithm are discussed in detail in the following section.

2.2. Voice Activity Detectors

The process of determining the speech signal from the non-speech (noise alone) signal in the acquired signal is called the voice activity detection. This is an important problem in speech processing systems, since it is very difficult to detect the speech in the presence of the background noise and this could lead to numerous detection errors. In order for the noise reduction scheme to work well, it requires a speech detection algorithm to work as a switch to turn on and off the adaptive filters. A good voice activity detection algorithm is vital in order to precisely reduce the background noise present in the acquired signal. Study done by Krasny and Oraintara (2002) shows that using a Voice Activity Detector (VAD) with noise reduction technique significantly improves the performance of the array processing algorithms (Krasny & Oraintara, 2002).

According to Savoji the essential characteristics of an ideal VAD are reliability, robustness, accuracy, adaptation, simplicity, real-time processing and no prior knowledge of the noise (Savoji, 1989). During the past decade, many researchers have studied different strategies to detect speech in the presence of background noise and its influence on speech related applications (Bouquin-Jeannes & Faucon, 1995; Bouquin-Jeannès & Faucon, 1994; Potamitis & Fishler, 2003; Sangwan et al., 2002; Sohn & Sung, 1998; Tanyer & Ozer, 1998; Wei, Du, Yan, & Zeng, 2003; Woo, Yang, Park, & Lee, 2000).

Voice activity detectors have been used for vast variety of speech communication applications, such as speech recognition (Hirsch & Ehrlicher, 1995; Karray & Martin,

2003), speech coding (Hoffman, Li, & Khataniar, 2001), hands-free telephony (Bouquin-Jeannès et al., 1996; Freeman, Cosier, Southcott, & Boyd, 1989; Krasny & Oraintara, 2002) and echo cancellation, to name a few.

The idea of using VAD was first investigated for the use on TASI (Time Assigned Speech Interpolation) systems (Bullington & Fraser, 1959). The VAD is used in the universal mobile telecommunication systems to achieve silence compression, and also to reduces the average bit rate by using the discontinuous transmission mode (Freeman et al., 1989). Global system for mobile communication (GSM) telephony uses silence detection and comfort noise injection for higher coding efficiency (Srinivasan & Gersho, 1993). In cellular radio system, it reduces co-channel interference and power consumption in portable equipment (Ezzaidi et al., 1997).

The VAD algorithm extracts some measured quantities from the received signal and compares these values with a threshold value to make decision about the received signal. If the calculated value is greater than the threshold then the received signal is considered speech and noise signal. Otherwise, the received signal is considered noise alone signal. The output value from this calculation will be either zero or one, where a one denotes "speech and noise" and a zero denotes "noise alone". This output value is employed to switch on and off the adaptive filters in the switched Griffiths-Jim beamformer (Subsection 2.1.1).

The performance of this beamformer algorithm highly depends on the performance of the VAD algorithm. If the VAD algorithm does not detect the presence of a speech signal, the adaptive filter will assume this signal as noise and suppress this signal.

Alternatively, if the VAD algorithm does not detect the noise alone signal, the adaptive filter will assume this as speech and will not suppress this signal. This kind of activity will reduce the overall performance of the speech related application. Therefore, it is very important to choose a good VAD algorithm that works well in all situations.

Voice activity detection algorithms fall into two main categories. The first category uses the direction of the received signal as the main criterion to differentiate between speech and background noise. The second category uses the statistics of the received signal to distinguish between speech and background noise. Some existing methods of the VAD algorithms are Itakura LPC distance measure, energy distribution, timing, pitch (Arcienega & Drygajlo, 2002), zero crossing rates (Junqua, Mak, & Reaves, 1994), cepstral features, Automatic variance control (Moir, 2001), spectral information, adaptive noise modelling of voice signals and periodicity measures (Tucker, 1992). These algorithms have some trade offs like computational cost, sensitivity, and accuracy. Brief explanations of some of the VAD algorithms that are suitable for this application are discussed in detail below.

2.2.1. Detection based on direction of the signal

This algorithm uses the direction of the received signal to differentiate between the presence of wanted speech and unwanted noise signals (H. Agaiby & T.J. Moir, 1997). It is assumed that the position of the user is within a predefined area (invisible viewing zone) facing the microphone systems. The speech signal is expected to be present inside this invisible viewing zone and any signal originated outside this zone is

considered background noise. If the user happens to be outside this predefined area, they could easily move themselves into this area to use the microphone system. This type of method restricts the user to stay within this zone of activity, where the speech is expected to be.

Since this algorithm uses the speakers direction as the main factor to distinguish the wanted speech, it has been proved to work under a variety of noise conditions including competing speakers (H. Agaiby & T. J. Moir, 1997). While most of the other VAD algorithms seem to be unsuccessful when there is competing speaker in the room. Since this algorithm makes use of two microphones, it can only estimate the direction of the signal source and not the position. Direction of the signal is approximated by estimating the time delay between the two received signals at the microphones (Strobel & Rabenstein, 1999).

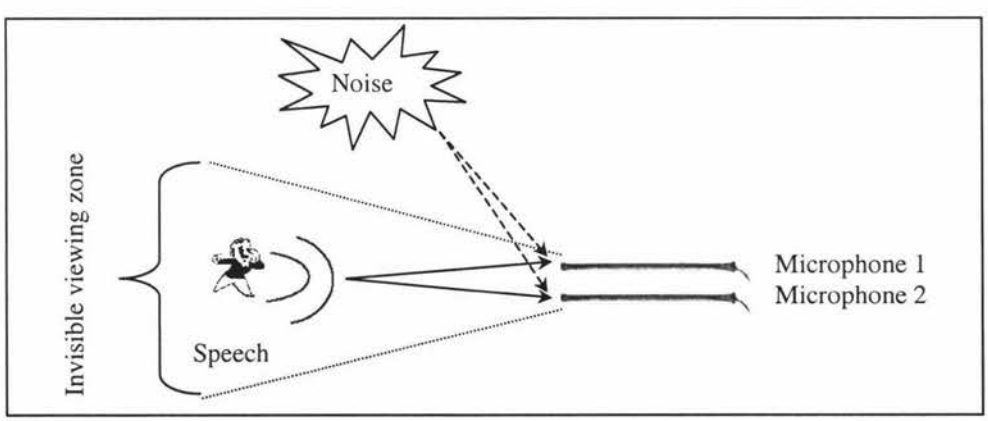


Figure 2.7. Invisible viewing zone

Figure 2.7 shows an ideal situation with a desired speaker and a background noise source. The received signal at the microphones contains the desired speech signal and the unwanted background noise. The two received signals $(x_1(t) and\, x_2(t))$ arriving at the microphones can be mathematically modelled (Schelkunoff, 1943) as:

$$x_1(t) = s(t) + n_1(t)$$
$$x_2(t) = \alpha s(t - D) + n_2(t)$$

where $s(t)$ is the desired speech signal, α is the attenuating factor, D is the time delay and $n_1(t) and\, n_2(t)$ is the noise sources. It is assumed that $s(t)$ is uncorrelated with $n_1(t) and\, n_2(t)$. Estimating the time delay D between the two received signals $x_1(t) and\, x_2(t)$ is the main problem in the above model.

The time delay of the received speech is likely to be close to zero, since the speech is expected to be in front of the microphones. The viewing zone is restricted by a threshold value (This is the maximum time delay allowed as speech signal). The time delay between the two received signals is calculated and it is compared with this threshold value. If the delay is less than the threshold value, then it is considered speech (signal is present inside the viewing zone) otherwise, it is considered noise.

Many authors have proposed various approaches in the past for estimating the time delay between two receives signals (Quazi, 1981). Some of these are Generalised cross correlation (GCC), Parameter estimation (Chan, Riley, & Plant, 1980), Cross-power spectrum phase method (Omologo & Svaizer, 1994), and Higher-order spectra. Out of the above methods, the most common method used for estimating time delay is the well known GCC methods (Knapp & Carter, 1976).

A brief explanation of the GCC method will be given here, however more detail discussion of this method can be found in the following literatures (David & Mordechai, 1985; Knapp & Carter, 1976; Scarbrough, Ahmed, & Carter, 1981). This method consists of two pre-whitening filters followed by cross-correlator. The structure of this algorithm is given in Figure 2.8.

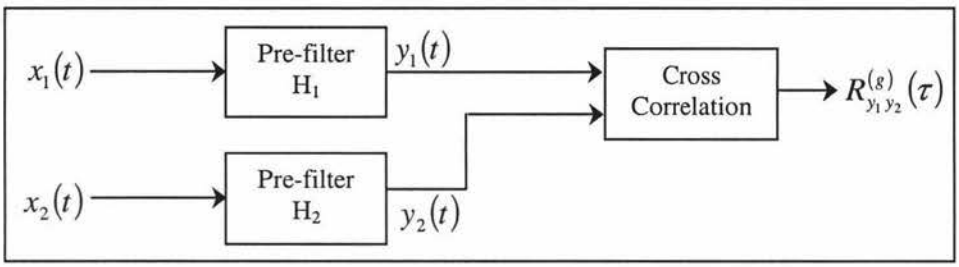


Figure 2.8. Generalised cross correlation method

Generalized correlation between the two inputs ($x_1(t)$ *and* $x_2(t)$) is estimated using the following equation:

$$R_{y_1 y_2}^{(g)}(\tau) = \int_{-\infty}^{\infty} \psi_g(f) G_{x_1 x_2}(f) e^{j2\pi f\tau} df$$

where $\psi_g(f) = H_1(f) H_2^*(f)$, refers to the general frequency weighting. The pre-whitening filters $(H_1(f)$ *and* $H_2(f))$ are used to improve the accuracy of delay estimated by attenuating the signal passed to the cross-correlation in spectral regions where the SNR is the highest. This frequency weighting doesn't affect the location at which the GCC function peaks occur. However, choosing an appropriate weighting function ensures a sharp peak in $R_{y_1 y_2}^{(g)}(\tau)$ rather than a broad one. The argument τ that maximises $R_{y_1 y_2}^{(g)}(\tau)$ is the desired estimate of the time delay D between the two signals.

Various pre-whitening filters, such as the Roth filter, smoothed coherence transform, phase transform, Eckart filter, and Hannan Thomson filter are proposed in the past. The choice of filtering is analysed by Knapp & Carter and a maximum likelihood estimator (ML) is proposed. It has also been shown that performance of the ML estimator is identical to the one proposed by Hannan and Thomson (Hannan & Thomsom, 1981). The ML estimator has the following weight function:

$$\psi_{ML}(f) = \frac{\left|\gamma_{x_1 x_2}(f)\right|^2}{\left|S_{x_1 x_2}(f)\right|\left[1-\left|\gamma_{x_1 x_2}(f)\right|^2\right]}$$

where $\gamma_{x_1x_2}(f)$ is the coherence function of the input signals.

This ML weighting funtion performs well under low reverberations. As the room reverberation increases the performance of this method degrades (Bedard, Champagne, & Stephanne, 1994). Improved method of the GCC is presented in the following literature (Jian, Kot, & Er, 1997), this method uses the symmtry measure of the GCC function to improve the performance of the time delay estimation under reverberations.

Another possibility is to use the magnitude squared coherence (MSC) calculation together with the GCC algorithm to detect the presence of the reverberation signals (Bouquin-Jeannes & Faucon, 1995). The MSC is calculated for the received signal and it is compared with a stored threshold to make a decision on whether echo is present or not. The MSC value computed for the non-reverberation speech signal is normally high (close to 1) and the MSC value computed for a reverberation speech is normally low (close to 0). In general, a threshold value of around 0.5 is chosen. Any MSC value higher than this threshold value is speech and any MSC value lower than this threshold value is reverberation.

The algorithm decides on a valid speech when both of the estimated time delay and the MSC functions detect the presence of speech. More detailed implementation of this technique can be found in the following literatures (H. Agaiby & T. J. Moir, 1997; Moir, 2003). An enhanced version of this algorithm has been implemented for three-microphone, which seems to give better results than the two-microphone algorithm (W.N. Chen & T.J. Moir, 1999; W.N. Chen & T. J. Moir, 1999). The main drawback of this type of algorithms is that it requires lot of computation power.

2.2.2. Detection based on Energy

The energy-based (Rabiner & Sambru, 1975) approach is one of the earlier VAD algorithms where the energy of the received signal is calculated and compared with the stored threshold value. This is the most commonly used algorithm to detect the presences of a speech signal. This algorithm is simple to implement and does not require lot of assumption about the noise characteristics. The only assumption made here is that the speech is sufficiently louder than the noise. This assumption is accurate at high SNR values.

The energy of the m^{th} frame of length N is defined as:

$$E(m) = \left\{ \frac{1}{N} \left[\sum x^2(n) - \frac{1}{N} \left[\sum x(n)^2 \right] \right] \right\}$$

where n is the time index.

The formula given above is to calculate a single frame, to obtain a more smother results the energy can be calculated using the sliding mean calculations. $\hat{E}(m)$ is defined as:

$$\hat{E}(m) = \beta \times E(m-1) + (1-\beta) \times \hat{E}(m)$$

where β is the forgetting factor (it varies between zero and one). If a faster convergence is required then β is chosen to be close to one. Otherwise, it is chosen to be close to zero for a slower convergence. In general, the value of β is chosen to be close to 0.9.

The calculated energy value is compared with the stored threshold value. If the calculated value is less then the threshold value the received signal is considered noise alone signal. Otherwise, it is considered speech and noise signal. This algorithm is sensitive to noise as a result of this variation in the noise statistics can reduce this

algorithms performance. There is also a problem of start and end of the speech segments not being detected by this algorithm when the threshold value is chosen higher than the energy of the speech signal. This can be avoided by choosing an acceptable threshold value, which can be estimated by trial and error process.

To improve the accuracy of the energy based VAD, it can be used with other VAD algorithms like the zero-crossing rate calculation to obtain more accurate detection (Lau & Chan, 1985). The zero-crossing rate is generally considered as a simple measure of the frequency content of speech. The average zero crossing rate is a good frequency estimate of the narrowband signals. However, speech signals are broadband signals, so it is less accurate. But when zero-crossing rate is combined with the energy measure for detection, it has proved to give marginally better results than using only energy (Bush, Ganapathiraju, Kornman, Trimble, & Webster, 1995).

2.2.3. Detection based on Entropy

The main feature of entropy-based detection is that it is less sensitive to the changes in the amplitude of the speech signal. The only assumption made in this algorithm is that the signal spectrum is more organised during speech segments than during noise segments (Shen, Hung, & Lee, 1998), therefore it is sensitive to spectral nature of the noise. Shannon's entropy is used to measure the organization of the signal. It measures the average length of bit code per symbol under optimal coding, and it is defined as:

$$H(s) = \sum_{i=1}^{N} P(s(i)) \times \log_2 (P(s(i)))$$

where $s = [s(1),...,s(N)]$ represents a source of N symbols, and P (s(i)) is the probability of emission of symbol i. The entropy H(S) is maximal when all the symbols have equal probability and it is minimal when one symbol has a probability of one and the others of zero.

The measure of entropy is defined in the spectral energy domain as:

$$H\left(\left|Y(\omega,t)\right|^2\right) = \sum_{\omega=1}^{\Omega} P\left(\left|Y(\omega,t)\right|^2\right) \times \log_2\left(P\left(\left|Y(\omega,t)\right|^2\right)\right)$$

where $P\left(\left|Y(\omega,t)\right|^2\right) = \frac{\left|Y(\omega,t)\right|^2}{\sum_{\omega=1}^{\Omega}\left|Y(\omega,t)\right|^2}$ is the probability of the frequency band ω for the magnitude spectrum for frame t.

When the received signal is white noise the entropy is maximum, and when the received signal is pure tone the entropy is minimum. The above calculation is quite appropriate for white or quasi white noise; however, it will perform poorly under coloured noise. For this algorithm to work under coloured noise, the spectrum of each frame needs to be divided by the average spectrum computed over all frames. Also before computing the spectrum, a white noise with small amplitude is added to the signal.

More detail implantation of this method can be found in the following literatures (Renevey & Drygajlo, 2001; Shen et al., 1998; Waheed, Weaver, & Salam, 2002). Detection based on entropy of the magnitude spectrum has been proved to work in stationary, non-stationary, white and coloured noise conditions at SNR from 10 dB down to -10 dB and below (Renevey & Drygajlo, 2001). Some research proves that entropy based VAD performs better than energy based VAD.

3. Real-time Implementation

The first part of this chapter describes the hardware products used in this project and their setup for the project, and the second part of the chapter describes the software implementation of the chosen beamformer algorithm. The chosen algorithm has been implemented in real-time for a two-microphone setup.

3.1. Hardware

Hardware Equipments required for this project are:

- 2-microphones
- Pre-amplifier
- TMS320C6711 DSP Starter Kit (DSK) board
- PCM3003 stereo codec daughter board
- PC equipped with software package to communicate with the DSK

Texas Instruments TMS320C6711 (C6711) DSK is used in this project as a hardware platform for the development of a microphone array beamformer. The C6711 DSK saves time and expense of building prototype development boards and simplifies rapid implementation of the target DSP applications. This DSK is intended for desktop operation while connected to the parallel port of the PC. The PC can control the DSK via the development environment called "Code Composer Studio" (CCS), which is created by Texas Instrument (TI).

The DSK board includes the C6711 floating-point digital signal processor (DSP) and a 16-bit codec AD535 for accessing the input and output channels. This board also has a special input filter for anti-aliasing to eliminate erroneous signals, and an output filter to smooth or reconstruct the processed output signal. This platform allows the user to efficiently develop and test applications with the C6711 DSPs. This onboard codec provides a fixed sampling rate of 8 kHz.

In a basic system, the received input signal is send to an analog-to-digital converter (ADC) and the C6711 DSP processes this resulting digital representation of the received signal. Later, this processed signal is sent to the digital-to-analog converter (DAC) as the final output of this system. This DSP is based on a very-long-instruction-word (VLIW) architecture, which is well suited for numerically intensive calculations. It is capable of performing up to 900 million of floating point operations per second (MFLOPS) at a clock rate of 150 MHz.

The DSK board has 16 Mbytes of synchronous dynamic random access memory (SDRAM) and 128 Kbytes of flash programmable and erasable read only memory (ROM). Since this board has the flash programmable and erasable ROM, an external power supply needs to be connected to the board at all time during the operation. More details about this development environment can be found in the following literatures (Chassaing, 2002; *TMS320C62x/C67x Programmer's Guide*, 1999; *TMS320C6000 DSP Platform Tools Documentation*, ; *TMS320C6000 Peripherals Reference Guide (SPRU190D)*, 2001; *TMS320C6000 Technical Brief (SPRU197D)*, 1999).

The PCM3003 stereo codec daughter board is used in this project to obtain the input signals (Chassaing, 2002; *PCM3002/PCM3003 16-/20-Bit Single-Ended Analog Input/Output Stereo Audio Codec (SBAS079)*, 2000). The PCM3003 stereo codec provides an alternative to the AD535 codec. This daughter board is attached to the C6711 DSK via the header connection. Figure 3.1 shows a picture of the C6711 DSK and the PCM3003 audio daughter board.

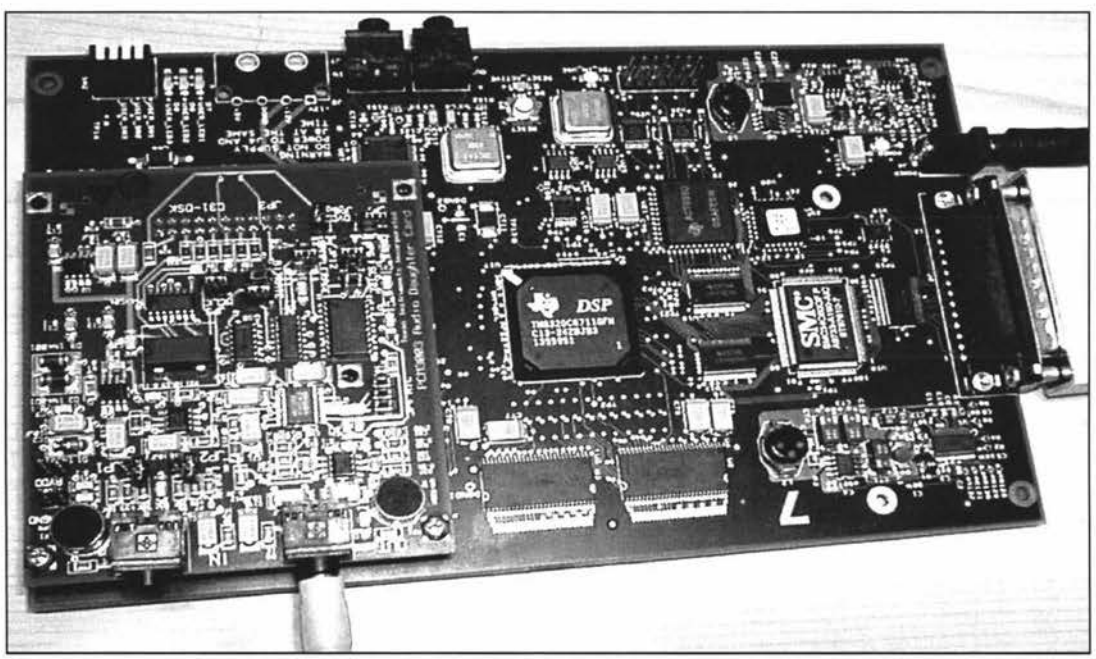


Figure 3.1. C6711 DSK and PCM3003 daughter card

The PCM3003 audio daughter board provides 16-bit stereo analog input and output channels at variable sampling rate of up to 73 kHz. This daughter board allows the user to choose between two sampling rate settings, one is to have the sampling rate at fixed 48 kHz, and other is to set the required sampling rate in the program. Although a variable sampling rate of 73 kHz can be used, TI recommends not to use any sampling rate over 48 kHz.

This daughter board also has two microphones built on the board and by leaving the input channels unconnected it can be accessed. However, these microphones will not be used here since the microphones need to be close to each other for our application. Two identical cardioid microphones are used in this project to obtain a better quality speech signal from the user. Cardioid microphones are designed to pick up sound that originates from the front and reduce the pickup levels at the sides, while rejecting sounds that originate from the back.

Figure 3.2 (*The ABC's of AKG: Microphone Basics & Fundamentals of Usage*) shows a picture of the microphone components: GN30 gooseneck mounting module and the CK31 condenser capsule (made by AGK Acoustics). Two of these microphones are used in this project.

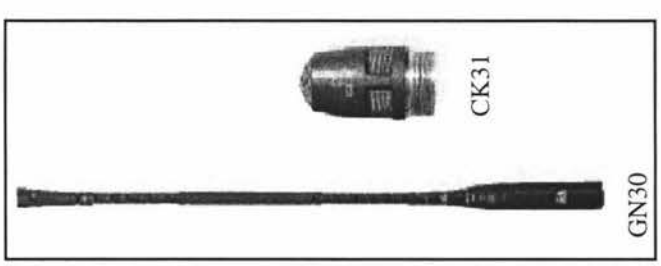


Figure 3.2. GN30 Gooseneck and CK31 capsule

These microphones are placed next to each other in a linear array facing the speaker. In order to avoid spatial aliasing, the distance (d) between these microphones should be chosen to be $d < \frac{\lambda_{min}}{2}$, where λ_{min} is the minimum wavelength. For a Nyquist frequency of 8 kHz and the speech of sound 330m/s, the minimum wavelength calculated as $\lambda_{min}=(1/8000)*330=4$cm. Therefore, the distance between the microphones should be less than 2cm. The microphones should be placed as close as possible to each other. An external dual microphone preamplifier (made by MIDIMAN audio buddy) is used to add additional gain to the microphone signals.

3.1.1. Hardware setup

Figure 3.3 shows the setup of the hardware equipments, which will be used in this project. Two identical microphones are used to acquire the signal from the environment. These two microphones are connected to the dual microphone preamplifier. This preamplifier is connected to the analog input of the PCM3003 audio daughter card. This signal is then passed to the C6711 DSK for further processing.

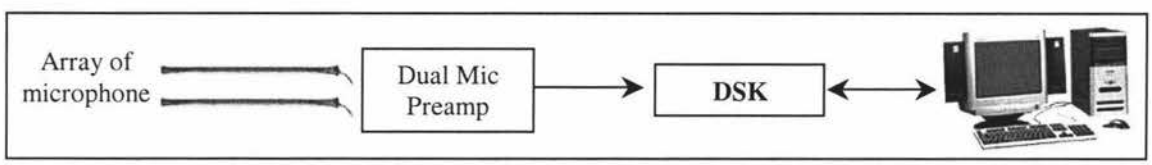


Figure 3.3. Overview of the project

The C6711 DSK is powered by the power supply and it is connected to the PC via a parallel port cable. The PC is equipped with Code composer studio (version 2.1) or Matlab (version 7) software for programming the DSK. Only one of this software can be used to program the DSK at any given time. When the programming is done in the chosen software environment, it is then loaded in to the DSK through the parallel port cable. While the program is loading to the DSK, it is essential for the DSK to be connected to the PC.

The signals received at the input channels will be processed in the C6711 DSK and an improved speech signal will be sent to the output channel. This processed signal could be used as an input for a speech recognition system. However, this signal can be passed to any speech related application that requires speech as input. The following section will describe the software packages used to program the DSK and how the chosen algorithm is implemented in real-time to reduce the noise from the received signal.

3.2. Software

There are two main software programs used in this project to communicate with the DSK; they are Code Composer Studio (CCS) and Matlab. Only one of these software packages can be used at any given time to communicate with the DSK. The following sections will discuss how the programming was carried out in different software platforms.

3.2.1. Matlab

Matlab is a software program used for numerical computations. It offers cutting-edge algorithms, enormous data handling abilities and powerful programming tools. There are two ways to program the DSK using the Matlab software package; first way is to use an interface file with Matlab m-file and the second way is to use Simulink. These methods are very simple and require less time to do the programming compared to CCS. Therefore, the choice was made to program in Matlab.

Simulink contains block sets that provide physical pathways to access the input and output connectors of the DSK and PCM3003 daughter card. Furthermore, Simulink library has a variety of toolbox's that can be used for many different applications. Once the Simulink model is created, using the build option this implementation can be automatically loaded into the DSK.

The main advantage of using Simulink to do the programming is that it reduces the implementation cycle. However, Simulink does not support real-time algebraic loops. Since switched Griffiths-Jim beamformer algorithm has two algebraic loops, it was not feasible to carry out the rest of the implementation in Simulink. Therefore, programming is carried out in m-file and some files are used to interface with the DSK.

A Matlab m-file can be used to communicate with the DSK by using the interface files c6x_DAQ.dll and pcm3003.out (Morrow, 2003a). These interface files were developed using Matlab's "mex" facility and Microsoft Visual C++, and they are freely accessible from the author's website (Morrow, 2003b). These files are needs to be in the same folder as the Matlab m-file in order for them to communicate with the DSK.

This method obtains the data from the input channels and transfers the digital representation of this data to the PC via the parallel port cable. Then this data will be processed in the Matlab environment and the analog representation of this result will be sent back to the output channel. In this method, this DSK board is used as a data acquisition toolbox, where the data passes through. This method does not use CCS like the Simulink method and it takes less time to implement the target application compared to other methods.

The Griffiths-Jim beamformer and VAD based on the time delay estimation has been implemented in Matlab m-file (this code can be found in the Appendix A). During the process of this implementation, there was a setback in this method of programming. The DSK required more bandwidth to move stereo data to and from the DSK, and the bandwidth of the parallel port connection was not able to support this. Because of this,

the frequency of the signal sent back to the output channel has been modified during this process. Therefore, the experiment done on this beamformer algorithm was not able to confirm the reduction of noise. However, the analysis carried out on the implemented VAD program has proved to identify the location of the speaker. Since none of these methods seems to give the desirable solution for our application, the next potential method is to program the DSK in CCS software.

3.2.2. Code composer studio

The CCS software is a fully integrated development environment supporting the Texas Instruments TMS320C6x DSP platforms (*Code Composer Studio Getting Started Guide (SPRU509C)*, 2001). It allows the user to edit, compile, link, and download the implemented program to the DSK board. It has graphical capabilities and it supports real-time debugging. CCS gives several options of programming languages to program the DSK; they are C/C++ or assembler. In this project, C language is used to implement the chosen algorithms.

The C complier in the CCS complies a C source program (with extension .c) to produce an assembly source file (with extension .asm). The assembler in the CCS assembles this assembly source file to produce a machine language object file (with extension .obj). Then the linker in the CCS combines the object files and object libraries as input to produce an executable file (with extension .out). This executable file represents a linked common object file format (COFF). This file can be loaded in to the DSK and run directly on the DSP.

The real-time analysis capabilities of CCS allow the user to probe, trace, and monitor the data while the program is running. These utilities are based on a real-time link and it produces awareness between the CCS environment and the target DSK. One of the main drawbacks of programming in CCS is that it is very time-consuming compared to other methods. In CCS, all the required algorithms have to be programmed by the user but in Matlab, they are already available. The chosen algorithm was implemented using the CCS software, since it seems to be the best option available. The following subsection will discuss how this was carried out.

3.2.3. The algorithm implementation

The DSK can be programmed in two ways depending on how the input values are read; they are the interrupt driven or the pooling technique. An interrupt driven technique reads the input samples every time an interrupt takes place, and an interrupt occurs after every sampling period ($T_s=1/F_s$). The polling technique uses a continuous procedure of testing to check when the data is ready to read. Polling is much simpler than the interrupt technique. Therefore, the polling technique is used for programming the DSK in this project.

The beamforming technique used for this project is the well-known switched Griffiths-Jim Beamformer, which was explained earlier in Subsection 2.1.1. In summary, the particular case of two-microphone system is shown in Figure 3.6. The received signal at the microphone is send to the VAD function for speech detection. This function returns a value "1" for speech and noise signal and value "0" for noise alone signal.

This result is used in the adaptive filters NLMS1 and NLMS2 to update the weight coefficients. When the VAD returns a value "1", the first adaptive filter is turned on; and when it returns a value "0", the second adaptive filter is turned on. Only one of these adaptive filters is updated at any given time.

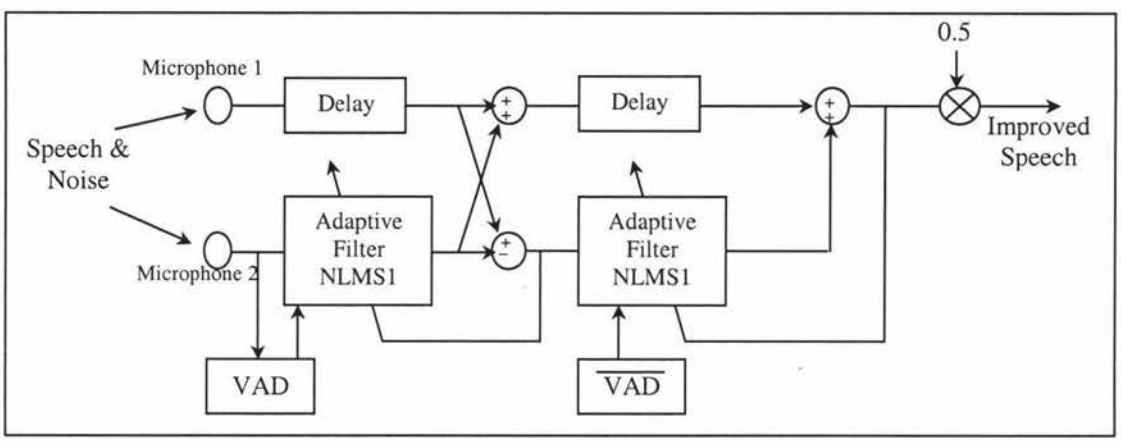


Figure 3.4. Beamformer structure

A simple VAD based on the energy value of the received signal is used with the switched Griffiths-Jim beamformer to switch on and off the adaptive filters for this project. This algorithm requires less computational power and it is simple and easy to implement than other VAD algorithms (explained earlier in the Subsection 2.2). Since the application of this project needs a real-time processor to perform the calculations, the selection of the algorithm was restricted to the algorithm that requires less computation cost.

The voice activity detector function is used to detect the presences of speech in the acquired signal. The acquired signal from one microphone is passed as the input to the VAD function. Then the energy of this signal is calculated and it is compared with a threshold value. If the calculated value is higher than the threshold value, it returns "1" (speech and noise) otherwise, it returns "0" (noise alone). This threshold value and the forgetting factor are chosen by trial and error process.

The following pseudo code illustrates the voice activity detector function based on energy of the received signal.

```
short vad(float input) {
        short temp=0;
        float b=0.9, vari=0.0;        //b-forgetting factor

        vari=varf;                    //stores the old value for further calculation

        //calculate energy of the received signal using a recursive formula
        varf=(b*vari)+((1-b)*input*input);

        //compares the calculated value with the threshold value
        if (varf>=15E7) {             //received signal contains speech &noise
                temp=1;
        } else {                      //received signal contains noise alone
                temp=0;
        }
        return temp;                  //output of the VAD algorithm
}
```

Energy based VAD program is given in Appendix B.1. A good VAD is very crucial in this beamformer algorithm. If the VAD does not detect the desired speech, then the adaptive filter will try to suppress this received signal assuming it as noise or visa versa. This kind of activity will reduce the reliability of the whole beamformer system.

The following pseudo code illustrates the adaptive filter based on NLMS algorithms (Subsection 2.1.2). The initialisations of the weight vector (we) and the input vector (xv) have been carried out in the main function. Initially they are made equal to zero. Adaptive filter program based on NLMS algorithm is given in Appendix B.2.

```
float nlms(float prim, float ref, float sw, int nh, float mu, float gamma,float
*we,float *xv) {
        float yn=0.0, xvt=0.0, E=0.0;
        xv[0]=ref;
        for(i=0; i<nh; i++)     {
                yn+=(we[i]*xv[i]);              //filter output
                xvt+=(xv[i]*xv[i]);             //input vector
        }
        if (nh==Nw1) m_yn=yn;                   //stores the first filter output
        E=prim-yn;                              //error output
        for(i=nh-1; i>=0; i--)                  //weight vector update
                we[i] = we[i] + ((mu*E*xv[i]*sw)/(gamma+xvt));
        for(i=nh-1; i>0; i--)                   //input vector update
                xv[i]=xv[i-1];
        return E;                               //returns the output
}
```

This beamformer algorithm makes use of two adaptive filters based on NLMS algorithm. The result from the VAD function is used to update these filters. The first filter is updated during the presence of speech and noise signal (when VAD function returns "1"). The second filter is updated during the presence of noise alone signal (when VAD function returns "0").

The following pseudo code is part of the C program given in Appendix B.3, which is used to perform the beamforming process.

```
//Right input signal is passed to the VAD function
//and the result is returned to switch1 variable
//switch1 contains "1" for speech and noise signal and "0" for noise signal
switch1=vad(right);
if (switch1==1) {                    //speech and noise
        switch2=0;
}else if (switch1==0) { //noise
        switch2=1;
}
//Implementation of the switched Griffiths-Jim beamformer
//fist filter is used as beam-steering filter
//switch1 is use to turn on/off the update of the NLMS1 filter coefficients
op_one=nlms(right,left,switch1,Nw1,1E-1,1E-2,we1,xv1);   //NLMS1
add=right+m_yn;       //m_yn contains the results from the first filter output
//second filter is used as ANC
//switch2 is use to turn on/off the update of the NLMS2 filter coefficients
op_two=nlms(add,op_one,switch2,Nw2,5E-1,1E-2,we2,xv2);        //NLMS2
output=(op_two)*5E-1;                    //beamformer output
```

A software base switch is used to turn on and off the beamformer algorithm programmed here. A slider GEL (General Extension Language) file is programmed as a software switch to slide through different output values while the processor is running. This method allows the user to compare the effects of two different algorithms without having to halt the processor and reprogram the DSK. All the variables for the slider

have to be defined in the program. The following is an example GEL file that slides between two variables zero and one to turn on and off a particular algorithm.

```
/*Adaptswitch.gel Slider for outputting different values*/
//parameters   0 - the average of the received signals goes to the output
//                1 – algorithm output
menuitem "A_sw"                          /*printed on the slider objective */
slider A_sw(0,1,1,1,outtype)  {
                                         /*increment by 1,from 0 up to 1*/
      out_type = outtype;       /*vary type of output*/
}
```

The following support files are used together with the source files given in Appendix B to build a project in CCS.

- Cdxdsk.cmd - sample linker command file
- C6xdsk.h - header file that defines addresses of external memory interface, the serial ports, etc
- C6xinterrupts.h - contains init functions for interrupts
- C6xdskinit_pcm.h - header file with the function prototypes
- C6xdskinit_pcm.c - communication support file
- Vectors.asm - handles interrupts, INT4 through INT15

Figure 3.4 shows the CCS file view of the all the necessary source codes used to create one project in CCS.

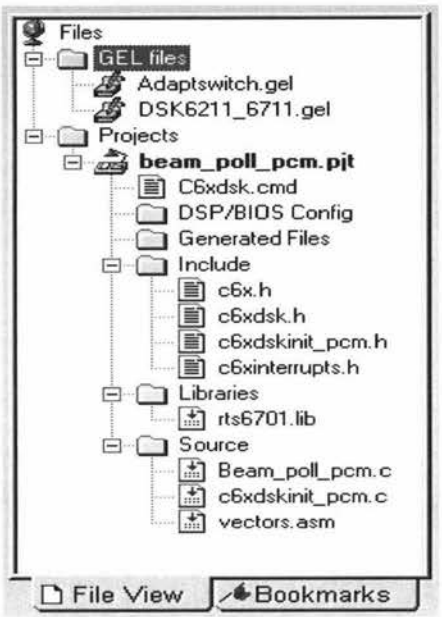


Figure 3.5. File view window

For all the programs implemented in the CCS software (programs in the Appendix B), the build options for linker and compiler were chosen as shown in the Figure 3.6.

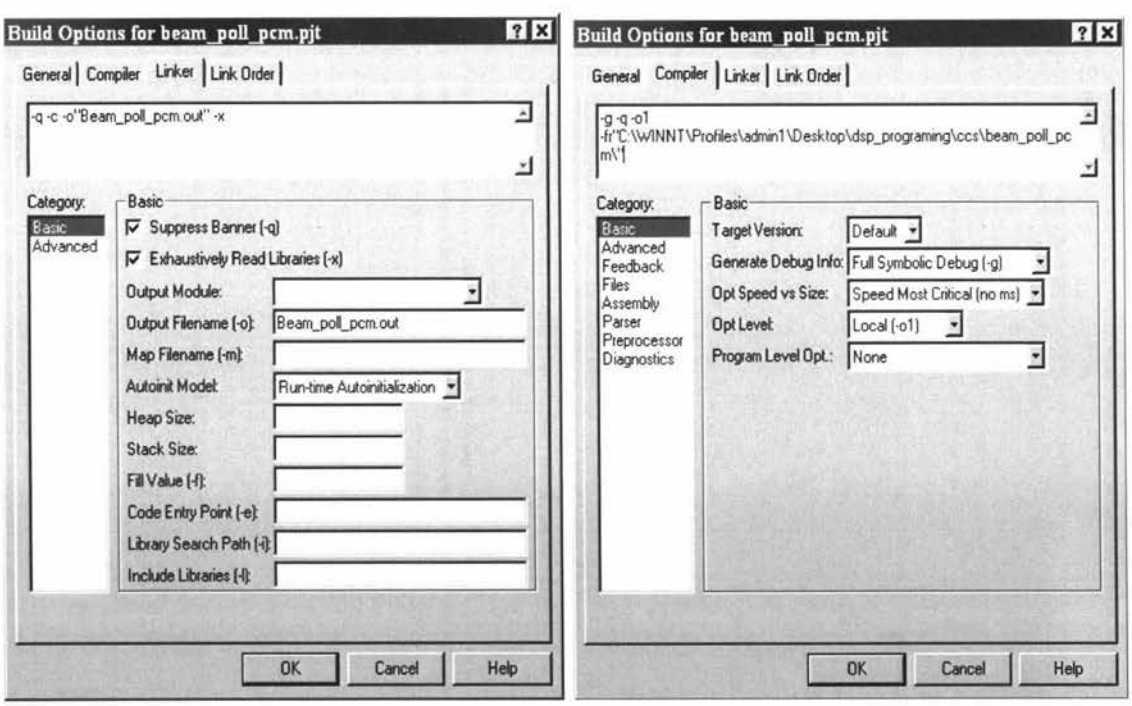


Figure 3.6. Linker and Compiler options

More information about programming the DSK using the CCS can be found in the following book (Chassaing, 2002). This book was referred during the implementation process to get help on programming this DSK and to obtain the support files, which are used in this project.

3.2.4. Speech Recognition

A simple speech recognition system was designed, in order to get a comparison of using a headset and a speech beamformer microphone. This is only an example of the speech recognition system, which could be used in the smart house project. However, there are more functions exits in the original speech recognition system that was designed for the smart house. This speech recognition system is not the focus of this project. (This work is part of another person's work so it will not be discussed in detail in this thesis.)

At present, two well-known companies dominate the market place in the speech recognition field; they are Dragon Naturally Speaking and Microsoft Speech SDK. Dragon naturally speaking requires the user to train the system before it can be used. Microsoft Speech SDK is speaker independent, which means that the user does not have to go through any training and it works well for any user. In commercial applications, it is not desirable for the user to go through a long process of training the system. Since Microsoft SDK 5.1 offers the flexibility of speaker independent, it was chosen for this project.

Microsoft Speech SDK version 5.1 and Visual Basic 6 are the software packages used in this project to develop a speech recognition system. More information about the Microsoft Speech SDK can be obtained from the company website (http://www.microsoft.com/speech/). Microsoft Speech SDK supports two different grammar types, dictation grammar and command and control grammar. The advantage of using the dictation grammar is that it allows all phrases in a language (unlimited grammar). However, when the grammar file is not defined, the speech recognition

system seems to fail more often. The advantage of using the command and control grammar is that it gives more accurate speech recognition results since the user has to specify the grammar file. Given that our application has limited phrases that need to be recognized, command and control grammar type is chosen for this project.

A complete speech recognition program developed using the Visual Basic software and Microsoft Speech SDK is given in the Appendix C.1. This program uses a grammar file, which contains all the possible commands expected from the user. This grammar file is given in the Appendix C.2 and this should be saved under the name of "newnums.xml" in the same folder as the Visual Basic program. At present, the grammar file contains commands such as one to thirty, light on, light off, radio on and radio off. However, this file can also be extended to contain more command as required by the user. It is also possible to program this example without using a grammar file.

The following pseudo code was declared when limited grammar is required:

```
'Must declare this in the program for grammar file use
myGrammar.CmdLoadFromFile App.Path & "\newnums.xml", SLOStatic
myGrammar.DictationSetState SGDSInactive
myGrammar.CmdSetRuleIdState 1, SGDSActive
If fRecoEnabled = True Then
   myGrammar.CmdSetRuleIdState 1, SGDSInactive
   cmdStart.Caption = "Start"
   fRecoEnabled = False
Else
   myGrammar.CmdSetRuleIdState 1, SGDSActive
   cmdStart.Caption = "Pause"
   fRecoEnabled = True
End If
```

The following pseudo code can be replaced with the above code, when free dictation is required:

```
'Must declare this in the program for free dictation grammar use
myGrammar.DictationSetState SGDSActive
If fRecoEnabled = True Then
   myGrammar.DictationSetState SGDSInactive
   cmdStart.Caption = "Start"
   fRecoEnabled = False
Else
   myGrammar.DictationSetState SGDSActive
   cmdStart.Caption = "Pause"
   fRecoEnabled = True
End If
```

This speech recognition system was created as a basic example of the required functions for the smart house. However, the user can easily add more functions as required to this system. Figure 3.7 shows the user interface of the created speech recognition system. As you can see there are three buttons shown on the user interface, they are Pause, Info, and Exit. The 'Pause' button stops the speech recognition system from working; this can be used when this system is not in use so it does not pickup any unnecessary words.

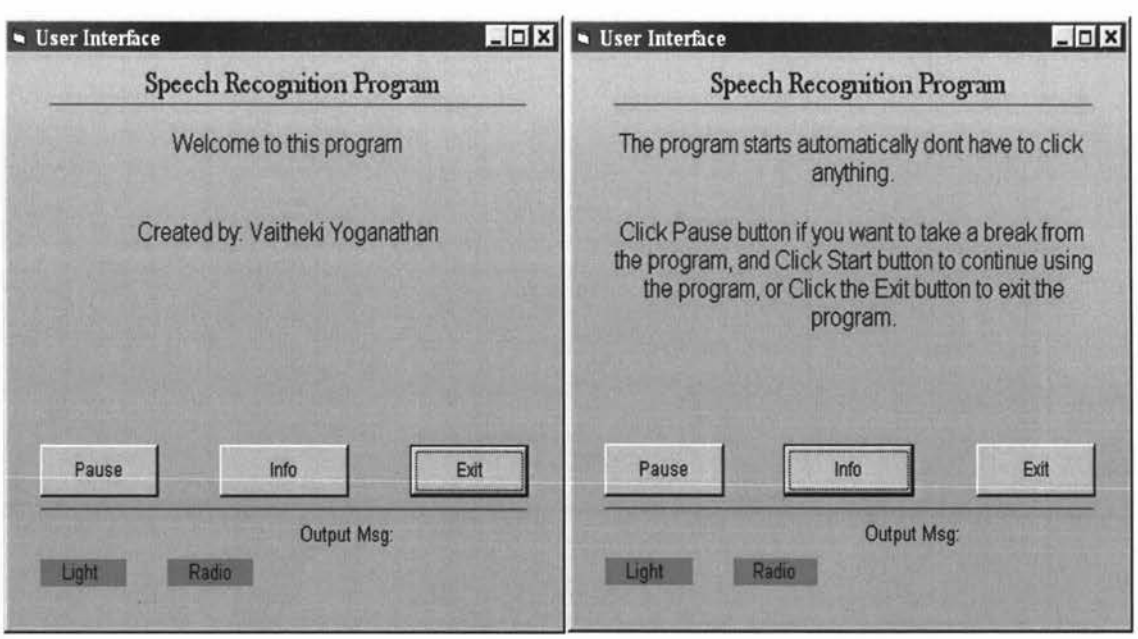


Figure 3.7. User Interface

The Info button provides the user the necessary information about how to use the system. Figure 3.7 also shows information presented to the user when the Info button is pressed. Exit button is used to exit the program. It should be noted that this speech recognition system is not connected to any external appliances to perform the requested commands, but with the necessary hardware equipments, this software can be used to perform the commands.

There were two labels created in the user interface, which represent Light and Radio appliances. In addition, they have been programmed to change the colour of the labels when the user requests a specific command. For example, when the user says "light on" command, the light label turns green and the output message confirms this action is completed. In addition, when the user says "light off" command, the light label turns red and the output message confirms this action is completed. Figure 3.8 shows an example of such a commands being executed in the speech recognition system.

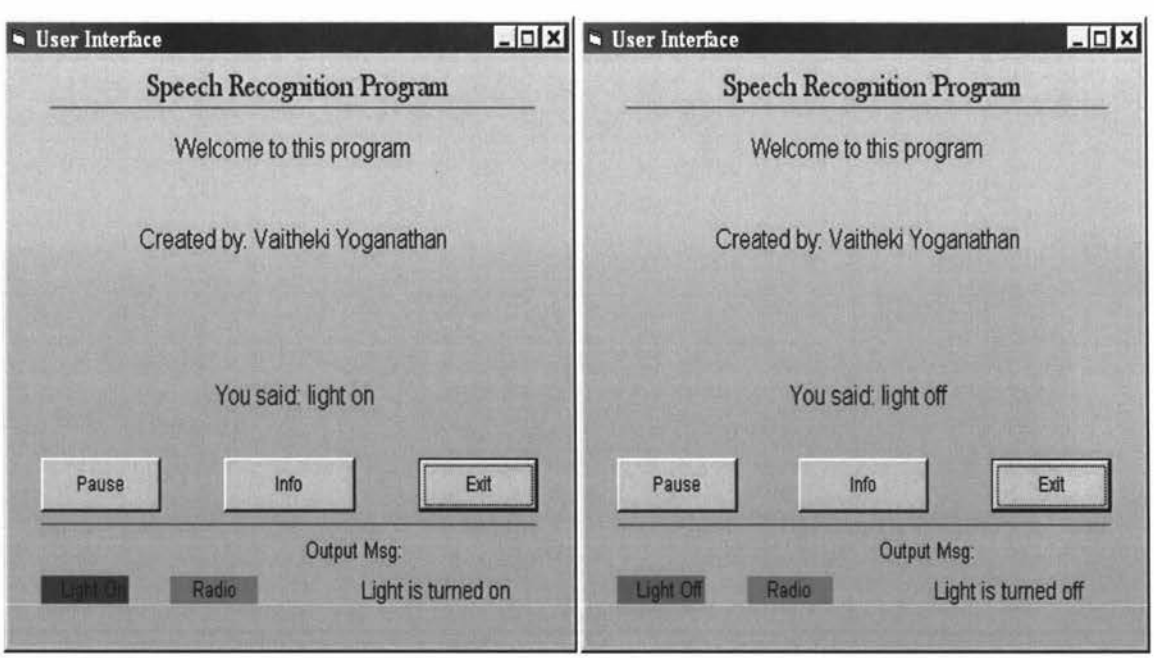


Figure 3.8. "Light on" and "Light off" commands

When the user says a command that is not in the grammar file, the system responds with "no recognition". This speech recognition system have been programmed to test for situations when the light is already on and the user commands to turn the Light on or visa versa, when the light is already off and the user commands to turn the Light off. In these occasions, the system is programed to respond with a message saying, "the light is already on" or "the light is already off", respectively.

Figure 3.9 shows an example of such situation being executed on the speech recognition system.

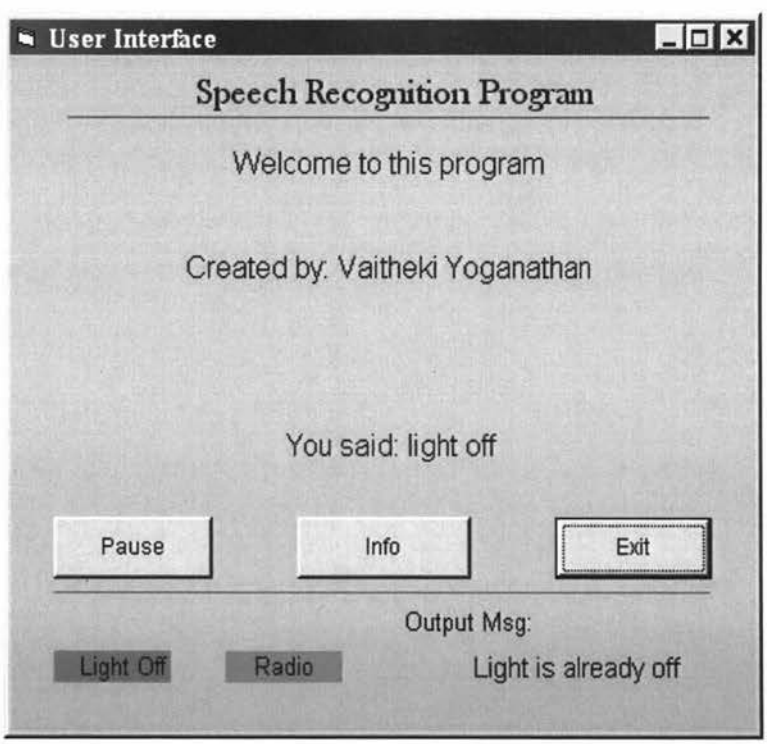


Figure 3.9. "Light off" command is said when the light is already off

Similar operations implemented on the Light appliance are also programmed for the Radio. Since this is only a prototype, only two common appliances were chosen and implemented in the system. However, more appliances can be added to this system, as required by the user.

4. Experimental Results

In this chapter, the implemented programs (discussed in Subsection 3.2.3) have been analysed and the results obtained from these experiments will be presented. A number of experiments were carried out to test whether the implemented program is working effectively.

The sampling frequency was chosen to be 16 kHz for all the experiments. The two microphones are placed side by side in a linear array facing the desired speaker. The environment consists of one desired source, a radio, computer fans, and ambiguous noise. In all the experiments, the desired speaker is about 1m away from the microphones. A radio was placed at a similar distance from the microphone system to act as interference to the speech. Figure 4.1 shows the room and the contents of the room where the experiment was carried out.

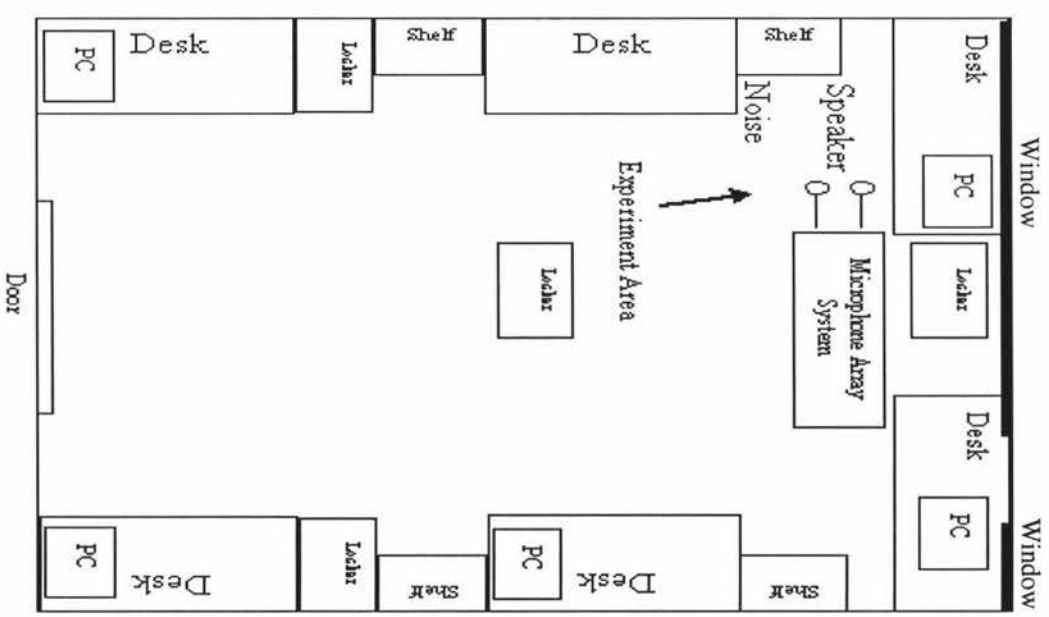


Figure 4.1. Experiment room

4.1. VAD Experiment

The purpose of this experiment is to determine the performance of the VAD algorithm based on the energy of the received signal. This algorithm is implemented using the CCS software and it can be found in the Appendix B.1. A switch implemented using CCS software component is used to turn on/off the VAD algorithm. For testing purposes when the VAD detects the noise alone segment, it is programmed to replace this segment with silence (zero) to monitor the difference.

The experiment was performed using the speech utterance "one, two, three", and the interference signal is created by a radio (it was playing some music). The threshold value to distinguish between speech and noise signal is chosen by trial and error process. A threshold value between 19e6 and 15e7 seems to give the best result. An example of the original speech and noise signal received at the microphone before the VAD algorithm processes is given in Figure 4.2.

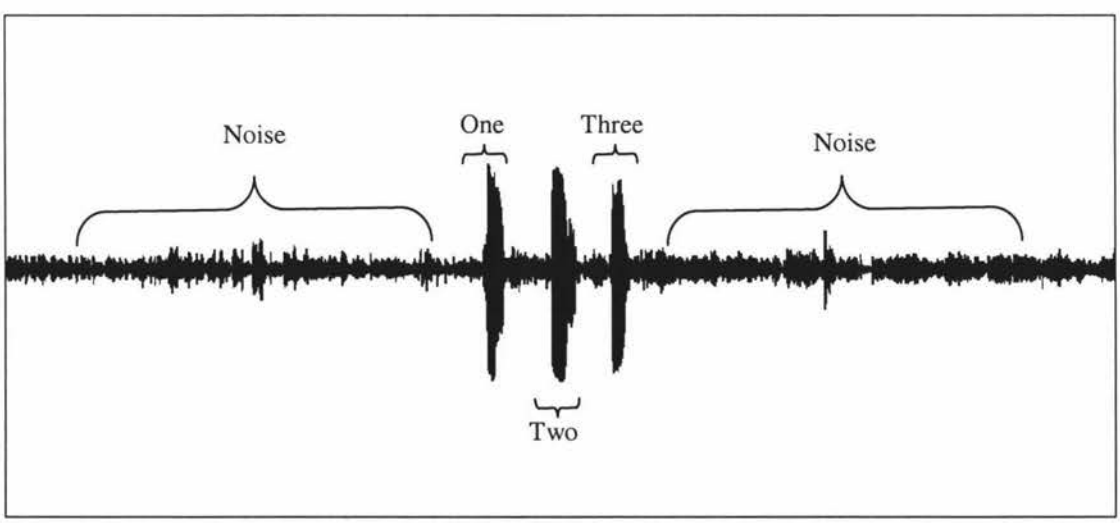


Figure 4.2. Speech and noise signal before VAD

For this experiment, a threshold value of 15e7 is chosen so it does not cut-off any parts of the speech signal. Figure 4.3 shows the resulting speech signal after the VAD algorithm has been applied.

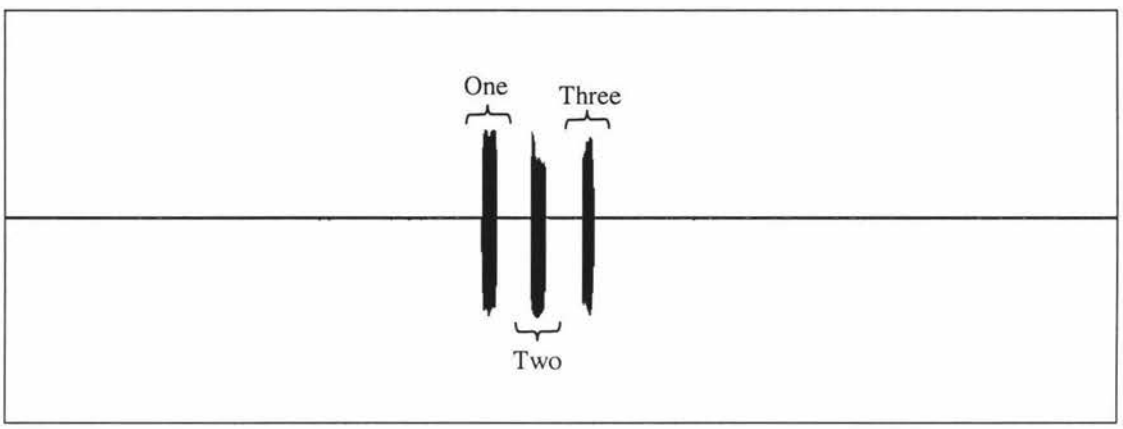


Figure 4.3. Resulting speech signal after VAD

The speech signal has been identified successfully and there are no incorrectly identified speech or noise segments. All the previous noise alone signal periods has been successfully replaced with silence periods. The noise present in the identified speech signal is not filtered, since the VAD algorithm is not a filter.

Another experiment was carried out to demonstrate the effect of choosing an appropriate threshold value. For this experiment, the sound level of the radio was increased and Figure 4.4 shows the resulting speech signal after the VAD algorithm.

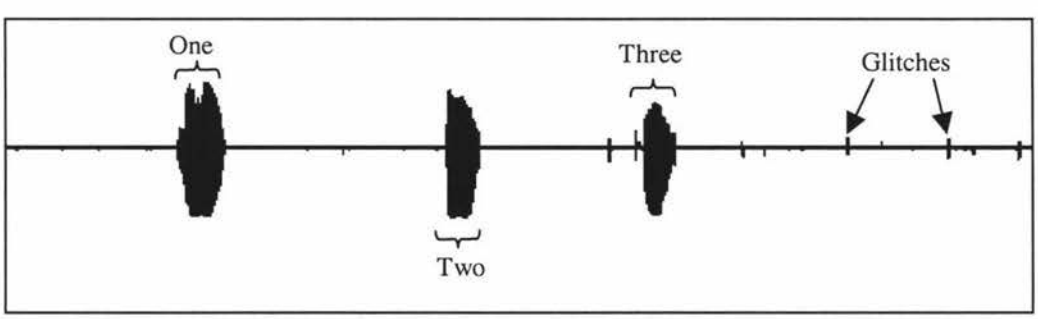


Figure 4.4. Glitch in the VAD output

The speech utterance "one, two, three" has been detected correctly but there is also a small amount of noise detected as speech in some situations. This is happening because the energy of the noise is higher than the chosen threshold value. This is one of the main constrain in the energy-based algorithm and it was expected to happen. In order to avoid this problem the threshold value can be increased, so that the energy of the noise will be lower than this value. Then there is also another problem of start and end of the speech been cut-off. By choosing an appropriate threshold value, this algorithm can be used successfully. Since this algorithm is easy to implement and takes less computational time, this is the best choice for this hardware platform.

4.2. Adaptive filter experiment

This experiment was conducted to test the performance of the adaptive filter based on NLMS algorithm. The adaptive filter program implemented using the CCS software is given in Appendix B.2. In order to verify the performance of the filter this experiment was performed off-line (not using speech), by using some predefined values as input signals for the adaptive filter.

Input channels of the adaptive filter were assigned the value "1" and the program is used to calculate the output error values. Other variable such as the number of filter coefficient is chosen to be "3", the step-size (μ) is chosen to be "0.5" and gamma (γ) is chosen to be "0.1". A manual calculation of the error output was performed using these values and it was compared with the results obtained from this program.

Figure 4.5 shows a comparison of the error output values for both of these calculations, for the first 10 iterations.

No of Iterations	Calculated error output values	Actual error output values
1	1	1
2	0.545455	0.545455
3	0.285714	0.285714
4	0.147465	0.147465
5	0.076111	0.075531
6	0.039283	0.038506
7	0.020275	0.01963
8	0.010465	0.010008
9	0.005401	0.005102
10	0.002788	0.002601

Figure 4.5. Results from the error output calculations

There is a slight difference between the manually calculated values and the actual values from the program, which might have been caused by the round off error between calculations. These values were plotted in a graph and it is given in Figure 4.6.

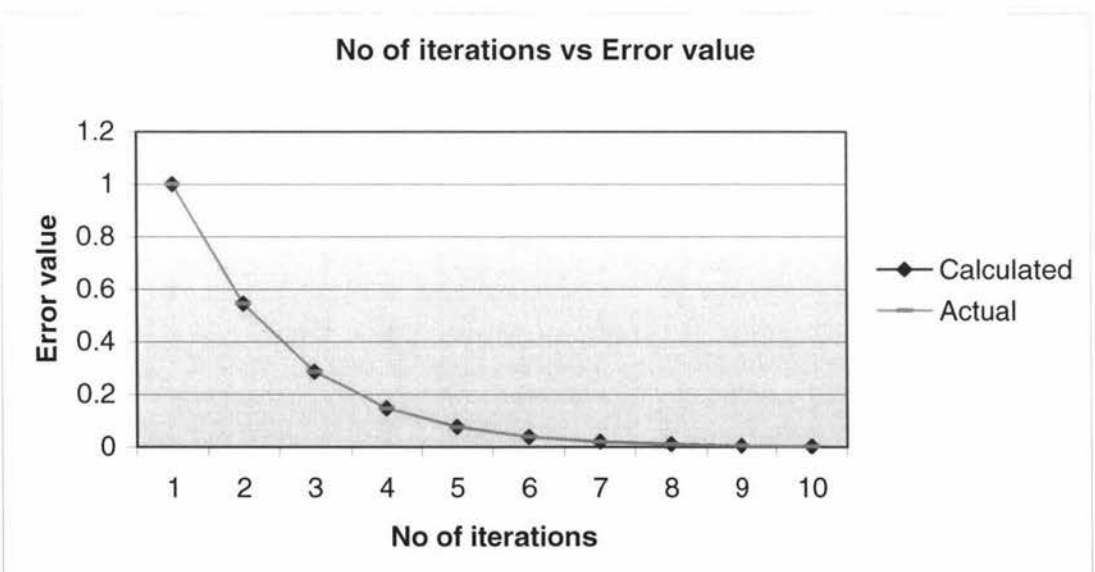


Figure 4.6. Graph of the error output calculations

As you can see, both graphs are identical and they are converging towards zero, which proves that this adaptive filter program is working.

4.3. Switched Griffiths-Jim beamformer experiment

This experiment is executed to test the performance of the switched Griffiths-Jim beamformer. The implemented software program for this experiment is given in Appendix B.3. A switch implemented using the CCS software components is used to turn on and off the beamformer algorithm. It allows the user to compare the effect of using the beamformer algorithm without having to reprogram the processor. A Labview program created by Dr. Tom Moir was used to evaluate the performance of this beamformer program. This Labview program is programmed to calculate the average power spectrum of a given signal. In addition, this program allows the user to store an old spectrum so it can be compared with the newly created spectrum.

For all the experiments, some speech was required at the beginning of the program to steers toward the speaker. The first experiment was carried out to identify the reduction of the ambiguous noise. In this experiment the adaptive filter coefficients of 50 is chosen for the first filter (NLMS1) and 100 is chosen for the second filter (NLMS2). The step-size (μ) of 0.1 was chosen for the first filter and 0.5 was chosen for the second filter. The gamma (γ) value of 0.01 was chosen for both filters to avoid division by zero at the denominator.

Figure 4.7 shows the calculated power spectrum of the input signal before the beamformer (switch = 0) and the output signal after the beamformer (switch = 1). As you can see on the graph, a considerable amount of improvement is achieved at low frequencies less than 1 kHz, where the ambiguous noises suppose to be.

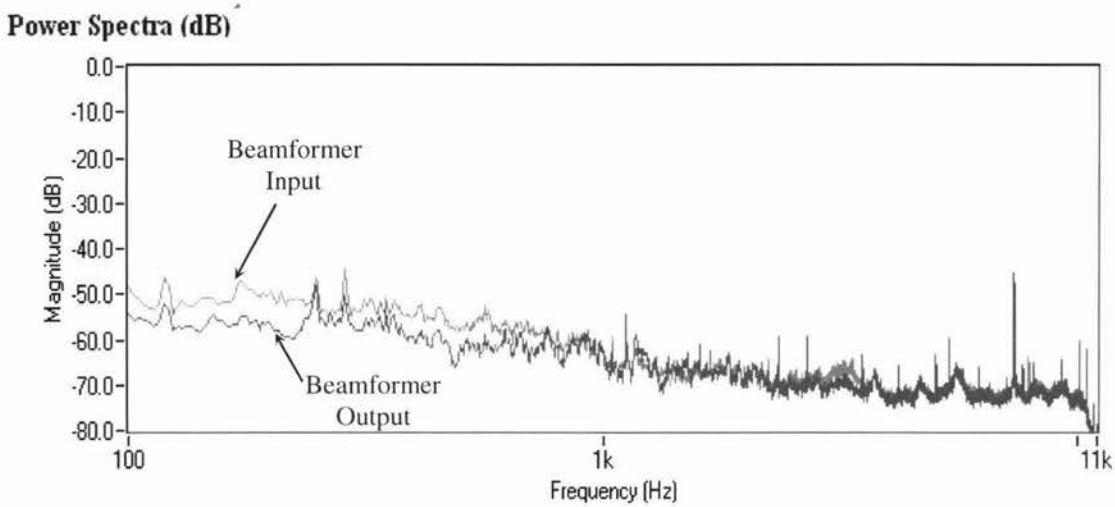


Figure 4.7. Ambiguous noise reduction

Another experiment was carried out to compare the reduction of the radio noise signal. All the variables were kept same as the above experiment. Figure 4.8 shows the power spectrum of the input signal before the beamformer and the output signal after the beamformer. There was a reduction of about 2dB was noticed in some situation in the output signal. The power spectrum of the beamformer output signal is more organised than the input signal.

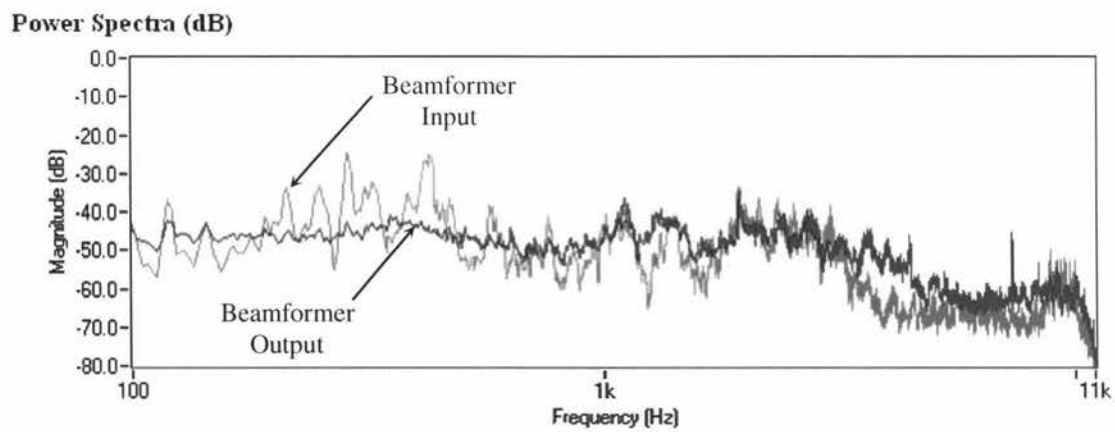


Figure 4.8. Radio noise reduction with low filter coefficients

Another experiment was carried out to observe the effects of using more filter coefficients on the beamformer algorithm. In this experiment the adaptive filter

coefficients of 100 is chosen for the first filter (NLMS1) and 400 is chosen for the second filter (NLMS2). All the other variables were kept same as the other experiments. The interference signal is created from a radio. Figure 4.9 shows the power spectrum of the input and output signal to the beamformer algorithm.

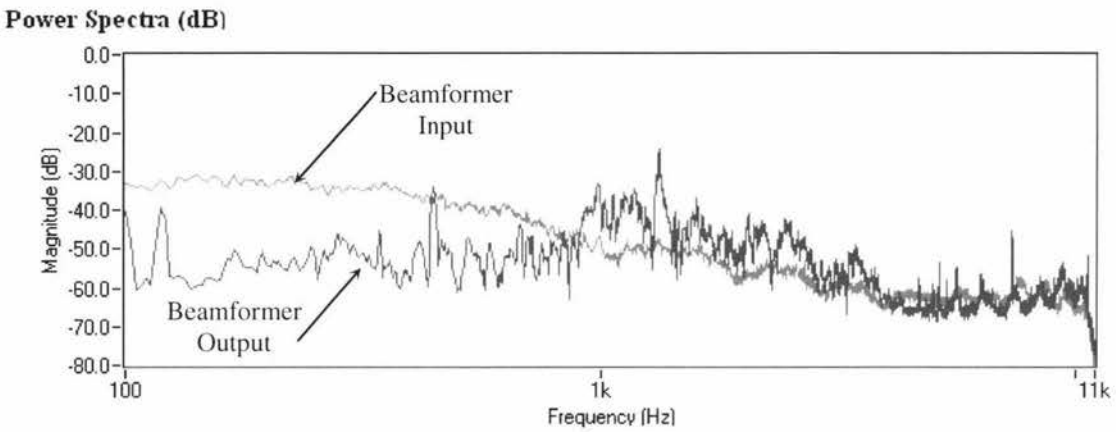


Figure 4.9. Noise reduction with high filter coefficients

As you can see on the graph, a significant amount of noise was reduced in the lower frequencies. However, when a big number of filter coefficients are used, the output signal is distorted. One possible reason for this problem is that the C6711 DSK is unable to handle large amount of calculations that were required by this algorithm. Because of this, the following input signal is missed, which causes the distortion in the output speech signal. Filter coefficients of 50 and 100 were the appropriate numbers that could be used for the adaptive filters, before the output speech signal is distorted.

This beamformer program was tested with the speech recognition system created in Visual Basic software. Although a moderate amount of noise reduction was obtained, the speech recognition system performed reasonably well. This dual microphone beamformer allowed the user to move around freely and issue commands to the speech recognition system, with the same performance as using a wearable microphone.

5. Conclusions and Future work

The switched Griffiths-Jim beamformer and the VAD based on the energy of the received signal were implemented in the DSK for real-time processing. The program has been designed in such a way that it can be modified to work for a range of applications. Using varies types of distorted speech signals the performance of this implemented system was tested. A noise reduction of about 2dB was observed in some situations while using the adaptive filter coefficients of 50 and 100. The choice of using larger filter coefficients in the adaptive filters was restricted by the limited computational power of the hardware. Due to this reason, only a small amount of noise reduction was achieved when compared to the true potential of this algorithm.

This program was tested with the speech recognition system created in Visual Basic software. Although the output of the beamformer was not improved significantly, a more user-friendly speech recognition system was obtained. The user of this dual microphone system can have the same feeling as the face-to-face conversation, without having to use hand-held or wearable microphones. This system can be used as a front-end device for the current speech technologies to provide hands-free operations and more human machine interfaces.

Although the performance of the present VAD algorithm is quite acceptable, there is scope for future improvements. One of the problems raised in the energy-based approach is that it fails abruptly when the energy of the noise signal is higher than the chosen threshold value. Because of this, the performance of the beamformer algorithm

is affected greatly. The accuracy of the energy-based approach can be improved by combining several different VAD techniques to detect the presence of the speech signal.

Even though the chosen beamformer algorithm has been successfully implemented in real-time, the processor was not able to handle the required processing power for the adaptive filter algorithm. The adaptive filter based on NLMS required more coefficients to achieve good noise reduction in the beamformer output. There are several ways that this problem can be solved. One solution is to implement the algorithm in ASM language instead of using C language. Using ASM language had proven to reduce the computational cost in some situations. However, it is best to use a much faster DSP to improve the performance of this beamformer algorithm. If a faster processor is not available then one alternative is to use another adaptive filter algorithm that requires fewer coefficients (less computational cost).

Using multi-processor boards is another potential solution that has been used by many developers these days. This allows the user to choose algorithms that give the best results rather than being restricted by the computational power. Since the chosen algorithm can be split into two different parts, it is ideal for multi-processor implementations. The VAD algorithm can be implemented in one DSP and the adaptive filters can be implemented in another DSP. Then the results from the first DSP can be sent to the second DSP to update the filters to produce the resulting signal. However, this method complicates the implementation process from a programming point of view. It is also possible to use more microphones to obtain a better result in the output but this will greatly increases the computational power.

References

The ABC's of AKG: Microphone Basics & Fundamentals of Usage. Retrieved 01, September, 2004, from http://www.akgusa.com/pages/data.html or http://www.musiciansbuy.com/mmMBCOM/html/akg/abcs_mic_basics.pdf

Abutaleb, A. S. (1988). An adaptive filter for noise cancelling. *Circuits and Systems, IEEE Transactions on, 35*(10), 1201-1209.

Affes, S., & Grenier, Y. (1997). A signal subspace tracking algorithm for microphone array processing of speech. *IEEE Trans. Speech Audio Processing, 5*, 425-437.

Agaiby, H., & Moir, T. J. (1997). *Knowing the wheat from the Weeds in Noisy Speech.* Paper presented at the ESCA Eurospeech 97, Rhodes, Greece.

Agaiby, H., & Moir, T. J. (1997). *A robust word boundary detection algorithm with application to speech recognition.* Paper presented at the 13th International Conference on Digital Signal Processing Proceedings,, Santorini, Greece.

An, P. E., Brown, M., & Harris, C. J. (1995). On the Convergence Rate Performance of Normalized Least-Mean-Square Adaptation. *Ieee Transactions On Neural Networks, 6*(6), 1549--1552.

Arcienega, M., & Drygajlo, A. (2002, September 16-20). *Robust voiced-unvoiced decision associated to continuous pitch tracking in noisy telephone speech.* Paper presented at the International conference on spoken language processing (ICSLP 2002), Denver, Colorado.

Armbruster, W., Czernach, R., & Vary, P. (1986). *Adaptive noise cancelling with reference input - possible applications and theoretical limits.* Paper presented at the EURASIP European Signal Processing Conference (EUSIPCO).

Bedard, S., Champagne, B., & Stephanne, A. (1994). *Effects of room reverberation on time-delay estimation performance.* Paper presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing.

Bitzer, J., Simmer, K. U., & Kammeyer, K.-D. (1999). *Theoretical noise reduction limits of the generalized sidelobe canceller (GSC) for speech enhancement.* Paper presented at the Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on.

Blogh, J. S., & Hanzo, L. (2002). *Third-generation systems and intelligent wireless networking : smart antennas and adaptive modulation.* England: John Wiley & Sons, Ltd.

Bouquin-Jeannes, R. L., & Faucon, G. (1995). Study of a voice activity detector and its influence on a noise reduction system. *Speech Communication, 16*(3), 245-254.

Bouquin-Jeannès, R. L., & Faucon, G. (1994). Proposal of a voice activity detector for noise reduction. *Electronics Letters, 30*(12), 930-932.

Bouquin-Jeannès, R. L., Faucon, G., & Ayad, B. (1996). How to improve acoustic echo and noise cancelling using a single talk detector. *Speech Communication, 20*(3-4), 191-202.

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

Bullington, K., & Fraser, J. M. (1959). Engineering aspects of TASI. *Bell System Technical Journal*, 353-364.

Bush, K., Ganapathiraju, A., Kornman, P., Trimble, J., & Webster, L. (1995). *A Comparison of energy-based endpoint detectors for speech signal processing.* Paper presented at the MS State DSP Conference.

Campbell, D. K. (1999). *Adaptive Beamforming Using a Microphone Array for Hands-Free Telephony.* Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Cao, Y., & Sridharan, S. (1993, December). *Multi-Microphone speech enhancement system.* Paper presented at the Proceedings of Workshop on Signal Processing and its applications, Brisbane.

Chan, Y. T., Riley, J. M., & Plant, J. B. (1980). A parameter estimation approach to time-delay estimation and signal detection. *IEEE Trans on Acoust., Speech, and Signal Processing, ASSP-28*(1), 8-16.

Chassaing, R. (2002). *DSP Applications Using C and the TMS320C6x DSK*. New York: John Wiley & Sons, Inc.

Chen, W. N., & Moir, T. J. (1999). *Active word boundary detection using three microphones.* Paper presented at the Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on.

Chen, W. N., & Moir, T. J. (1999). Adaptive noise cancellation for non-stationary real data background noise using three microphones. *Electronics Letters, 35*(23), 1991-1992.

Chien, J.-T., & Lai, J.-R. (2004). Use of Microphone Array and Model Adaptation for Hands-Free Speech Acquisition and Recognition. *Journal of VLSI Signal Processing Systems, 36*(2-3), 141-151.

Chien, J.-T., Lai, J.-R., & Lai, P.-Y. (2001). *Microphone array signal processing for far-talking speech recognition.* Paper presented at the Wireless Communications, 2001. (SPAWC '01). 2001 IEEE Third Workshop on Signal Processing Advances in.

Cho, J., & Krishnamurthy, A. (2003). *Speech enhancement using microphone array in moving vehicle environment.* Paper presented at the Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.

Choen, P. R., & Oviatt, S. L. (1995). *The role of voice input for human-machine communication.* Paper presented at the In Proceedings of the National Academy of Science.

Choi, C., Kong, D., Kim, J., & Bang, S. (2003). *Speech enhancement and recognition using circular microphone array for service robots.* Paper presented at the

Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.

Claesson, I., Nordholm, S. E., & Bengtsson, B. A. (1991). A Multi-DSP Implementation of a Broad-Band Adaptive Beamformer for Use in a Hand-Free Mobile Radio Telephone. *IEEE Transactions on vehicular technology, 40*(1), 194-202.

Code Composer Studio Getting Started Guide (SPRU509C). (2001, November). Retrieved 1 june, 2004, from www.ti.com

Cole, D., Moody, M., & Sridharan, S. (1993, December). *Enhancement of single microphone recordings in small highly reverberant rooms.* Paper presented at the Proceedings of Workshop on Signal Processing and Applications, Brisbane.

Collura, J. S. (1999). *Speech enhancement and coding in harsh acoustic noise environments.* Paper presented at the Speech Coding Proceedings, 1999 IEEE Workshop on.

Cox, H., Zeskind, R., & Owen, M. (1987). Robust adaptive beamforming. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 35*(10), 1365-1376.

David, H., & Mordechai, A. (1985). Time delay estimation between two phase shifted signals via generalized cross-correlation methods. *Signal Processing, 8*(2), 235-257.

Diegel, O., Lomiwes, G., Messom, C., Moir, T., Ryu, H., Thomsen, F., et al. (2005). *A Bluetooth Home Design @ NZ: Four Smartness.* Paper presented at the HOIT 2005 (Home-Oriented Informatics and Telematics) International Working Conference, University of York, UK.

Diniz, P. S. R. (1997). *Adaptive filtering algorithms and practical implementation.* USA: Kluwer Academic Publishers.

Elko, G. W. (1996, Jun 1995). *Microphone array systems for hands-free telecommunication.* Paper presented at the Speech Communication International workshop, Norway.

Er, M. H., & Ng, B. C. (1994). A new approach to robust beamforming in the presence of steering vector errors. *IEEE Transactions Speech Processing*, 1826-1829.

Ezzaidi, H., Bourmeyster, I., & Rouat, J. (1997). *A new algorithm for double talk detection and separation in the context of digital mobile radio telephone.* Paper presented at the Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on.

Farrell, K., Mammone, R. J., & Flanagan, J. L. (1992). *Beamforming microphone arrays for speech enhancement.* Paper presented at the Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on.

Feng, Z., Shi, X., & Huang, H. (1993). *An improved adaptive noise cancelling method.* Paper presented at the Electrical and Computer Engineering, 1993. Canadian Conference on.

Fischer, S., & Simmer, K. U. (1996). *Beamforming microphone arrays for speech acquisition in noisy environments.* Paper presented at the Speech Communication, International workshop; 4th -- 1995 Jun : Roros; Norway.

Flanagan, J. L., Johnston, J. D., Zahn, R., & Elko, G. W. (1985). Computer-steered microphone arrays for sound transduction in large rooms. *The Journal of the Acoustical Society of America, 78*(5), 1508-1518.

Freeman, D. K., Cosier, G., Southcott, C. B., & Boyd, I. (1989). *The voice activity detector for the PAN-European digital cellular mobile telephone service.* Paper presented at the International Conference on Acoustic Speech Signal Processing.

Frost, O. L. (1972). An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE, 60*, 926-935.

Gannot, S., Burshtein, D., & Weinstein, E. (2001). Signal enhancement using beamforming and nonstationarity with application to speech. *IEEE Trans. Signal Processing, 49*, 1614-1626.

Greenberg, J., Desloge, J., & Zurek, P. (2003). Evaluation of array-processing algorithms for a headband hearing aid. *The Journal of the Acoustical Society of America, 113*(3), 1646-1657.

Griffiths, L. J., & Jim, C. W. (1982). An alternative approach to linearly constrained adaptive beamforming. *IEEE Trans. Antennas Propagation, 30*, 27-34.

Hannan, E. J., & Thomsom, P. J. (1981). Delay estimation and the estimation of coherence and phase. *IEEE Trans on Acoust., Speech, and Signal Processing, ASSP-29*(3), 485-490.

Haykin, S. (2002). *Adaptive Filter Theory* (4 ed.). Upper Saddle River, New Jersey: Prentice Hall, Inc.

Haykin, S., & Widrow, B. (2002). *Least-mean-square adaptive filters.* New York: John Wiley & Sons Inc.

Hirsch, H. G., & Ehrlicher, C. (1995). *Noise estimation techniques for robust speech recognition.* Paper presented at the Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on.

Hoffman, M. W., Li, Z., & Khataniar, D. (2001). GSC-based spatial voice activity detection for enhanced speech coding in the presence of competing speech. *Speech and Audio Processing, IEEE Transactions on, 9*(2), 175-178.

Honig, M. L., & Messerschmitt, D. G. (1984). *Adaptive Filters Structures, Algorithms, and Applications.* USA: Kluwer Academic Publishers.

Hoshuyama, O., Sugiyama, A., & Hirano, A. (1999). A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters. *IEEE Transactions on signal processing, 47*(10), 2677-2684.

Hussain, A., Campbell, D. R., & Moir, T. J. (1997). *Multi-sensor sub-band adaptive noise cancellation for speech enhancement in an automobile environment.* Paper presented at the Adaptive Signal Processing for Mobile Communication Systems (Ref. No. 1997/383), IEE Colloquium on.

Istepanian, R. S. H., & Stojanovic, M. (2001). *Underwater acoustic digital signal processing and communication systems.* Boston: Kluwer Academic Publishers.

Jenkins, W. K., Hull, A. W., Strait, J. C., Schnaufer, B. A., & Li, X. (1996). *Advanced concepts in adaptive signal processing.* USA: Kluwer academic publishers.

Jian, M., Kot, A. C., & Er, M. H. (1997). *TIme delay estimation at multi-path based on symmetry measure of GCC function.* Paper presented at the International Conference on Information, Communications and Signal Processing (ICICS 97), Singapore.

Junqua, J., Mak, B., & Reaves, B. (1994). A Robust Algorithm for Word Boundary Detection in Presence of Noise. *IEEE Trans. on speech and audio processing, 2*(3), 406-412.

Kaneda, Y., & Ohga, J. (1986). Adaptive microphone-array system for noise reduction. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 34*(6), 1391-1400.

Kang, G., & Fransen, L. (1987). Experimentation with an adaptive noise-cancellation filter. *Circuits and Systems, IEEE Transactions on, 34*(7), 753-758.

Karray, L., & Martin, A. (2003). Towards improving speech detection robustness for speech recognition in adverse environment. *Speech communication, 40*(3), 261-276.

Kluwer, T. (2001). *Development of a test-bed for smart antenna, using digital beamforming.* Unpublished M. Sc. Thesis, University of Twente.

Knapp, C., & Carter, G. (1976). The generalized correlation method for estimation of time delay. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 24*(4), 320-327.

Krasny, L., & Oraintara, S. (2002). *Voice activity detector for microphone array processing in hand-free systems.* Paper presented at the Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2002.

Krim, H., & Viberg, M. (1996). Two Decades of Array Signal Processing Research - The Parametric Approach. *IEEE signal processing Magazine.*

Kuo, S., Ranganathan, G., Gupta, P., & Chen, C. (1988). Design and implementation of adaptive filters. *IEEE 1988 International Conference on Circuits and Systems.*

Lau, Y.-K., & Chan, C.-K. (1985). Speech recognition based on zero crossing rate and energy. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 33*(1), 320-323.

Li, Z., & Hoffman, M. W. (1999). Evaluation of microphone arrays for enhancing noisy and reverberant speech for coding. *Speech and Audio Processing, IEEE Transactions on, 7*(1), 91-95.

Lim, J. S. (1983). *Speech Enhancement.* Englewood Cliffs, NJ: Prentice Hall.

Lin, C.-T., Lin, J.-Y., & Wu, G.-D. (2002). A robust word boundary detection algorithm for variable noise-level environment in cars. *IEEE Transactions on Intelligent Transportation Systems, 3*(1), 89-101.

Martin, T. B. (1976). Practical applications of voice input to machines. *Proceedings of the IEEE, 64*, 487-501.

McCowan, I. A. (2001a). *Microphone Arrays : A Tutorial.* Australia: Queensland University of Technology.

McCowan, I. A. (2001b). *Robust Speech Recognition using Microphone Arrays.* Queensland University of Technology, Australia.

McHugh, R., Shaw, S., & Taylor, N. (1994). *A general purpose digital focused sonar beamformer.* Paper presented at the OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings.

Moir, T. J. (2001). Discrete-time variance tracking with application to speech processing. *Research Letters in the Information and Mathematical Sciences, 2*, 71-80.

Moir, T. J. (2003). Cancellation of noise from speech using Kepstrum analysis. *Research Letters in the Information and Mathematical Sciences, 4*, 101-111.

Moore , D., & McCowan , I. (2003). *Microphone Array Speech Recognition : Experiments on Overlapping Speech in Meetings.* Paper presented at the In Proceedings of ICASSP 2003.

Morrow, M. G. (2003a). C6X DSK - Matlab (Version 3.1). Madison, WI USA: College of Engineering, University of Wisconsin-Madison.

Morrow, M. G. (2003b). *TMS320C6X11 DSK to Matlab direct interface software - (C6X DSK - Matlab version 3.1).* Retrieved January, 2005, from http://eceserv0.ece.wisc.edu/~morrow/software/

Mucci, R. A. (1984). A Comparison of Efficient Beamforming Algorithms. *IEEE Trans. on Acoust., Speech, Signal Processing, 32*(3), 548-557.

Mumolo, E., Nolich, M., & Vercelli, G. (2003). Algorithms for acoustic localization based on microphone array in service robotics. *Robotics and Autonomous Systems, 42*(2), 69-88.

Omologo, M., & Svaizer, P. (1994). *Acoustic event localization using a cross-power spectrum phase based technique.* Paper presented at the ICASSP - 1994, Adelaide, Australia.

Orgren, A. C., Dasgupta, S., Rohrs, C. E., & Malik, N. R. (1991). Noise cancellation with improved residuals. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], 39*(12), 2629-2639.

Ortega-Garcia, J., & Gonzalez-Rodriguez, J. (1996). *Overview of speech enhancement techniques for automatic speaker recognition.* Paper presented at the Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on.

PCM3002/PCM3003 16-/20-Bit Single-Ended Analog Input/Output Stereo Audio Codec (SBAS079). (2000, 2000). Retrieved 1 June, 2004, from www.ti.com

Potamitis, I., & Fishler, E. (2003). Speech activity detection of moving speaker using microphone arrays. *Electronics Letters, 39*(16), 1223-1225.

Quazi, A. H. (1981). An overview on the time delay estimate in active and passive systems for target localization. *IEEE Trans on Acoust., Speech, and Signal Processing, ASSP-29*(3), 527-533.

Rabiner, L. R., & Sambru, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell System Technical Journal, 54*(2), 297-315.

Rappaport, T. S. (1998). *Smart antennas : adaptive arrays, algorithms, & wireless position location.* Piscataway, NJ: The Institute of Electrical and Electronics Engineers.

Raykar, V. C. (2001). *A study of a various Beamforming Techniques and Implementation of the Constrained Least Mean Squares algorithm for Beamforming.* College Park: Department of Electrical and Computer Engineering, University of Mayland.

Renevey, P., & Drygajlo, A. (2001, September 3-7). *Entropy based voice activity detection in very noisy conditions.* Paper presented at the 7th European conference on speech communication and technology (EUROSPEECH-2001), Aalborg, Denmark.

Roe, D. B., & Wilpon, J. G. (1994). *Voice communication between humans and machines.* Washington, D.C., USA: National Academy of Science.

Sambur, M. (1978). Adaptive noise canceling for speech signals. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 26*(5), 419-423.

Sangwan, A., Chiranth, M. C., Jamadagni, H. S., Sah, R., Prasad, R. V., & Gaurav, V. (2002). *VAD techniques for real-time speech transmission on the Internet.* Paper presented at the International Conference on High-Speech Networks and Multimedia Communication.

Savoji, M. H. (1989). A robust algorithm for accurate end-pointing of speech signals. *Speech Communication, 8*(1), 45-60.

Scalart, P., & Filho, J. V. (1996). *Speech enhancement based on a priori signal to noise estimation.* Paper presented at the Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on.

Scarbrough, K., Ahmed, N., & Carter, G. C. (1981). On the simulation of a class of time delay estimation algorithms. *IEEE Trans on Acoust., Speech, and Signal Processing, ASSP-29*(3), 534-540.

Schelkunoff, S. A. (1943). A mathematical theory of linear arrays. *Bell System Technical Journal, 22*, 80-107.

Seabra Lopes, L., & Teixeira, A. J. S. (2000). *Human-Robot Interaction through Spoken Language Dialogue.* Paper presented at the Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000), Japan.

Shen, J.-l., Hung, J.-w., & Lee, L.-s. (1998). *Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments.* Paper presented at the Proc. Int. Conf. on Spoken Lang. Processing, Sydney ICSLP-98.

Silverman, H. (1987). Some analysis of microphone arrays for speech data acquisition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 35*(12), 1699-1712.

Sohn, J., & Sung, W. (1998). *A voice activity detector employing soft decision based noise spectrum adaptation.* Paper presented at the International conference on acoustic speech signal processing.

Srinivasan, K., & Gersho, A. (1993). *Voice activity detection for cellular networks.* Paper presented at the Speech Coding for Telecommunications, 1993. Proceedings., IEEE Workshop on.

Strobel, N., & Rabenstein, R. (1999). *Classification of time delay estimates for robust speaker localization.* Paper presented at the IEEE Int. Conf. on Acoustics, Speech & Signal Processing (ICASSP), Phoenix, USA.

Tanyer, S. G., & Ozer, H. (1998). *Voice activity detection in nonstationary Gaussian noise.* Paper presented at the Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on.

TMS320C62x/C67x Programmer's Guide. (1999, May). Retrieved 1 June, 2004, from www.ti.com

TMS320C6000 DSP Platform Tools Documentation. Retrieved 1 June, 2004, from www.dspvillage.ti.com/freetools and www.ti.com/sc/knowledgebase

TMS320C6000 Peripherals Reference Guide (SPRU190D). (2001, Feb). Retrieved 1 june, 2004, from www.ti.com

TMS320C6000 Technical Brief (SPRU197D). (1999, Feb.). Retrieved 1 June, 2004, from www.ti.com

Tsoulos, G. V. (2001). *Adaptive antennas for wireless communications.* New York: IEEE Press.

Tucker, R. (1992). Voice activity detection using a periodicity measure. *Communications, Speech and Vision, IEE Proceedings I, 139*(4), 377-380.

Valin, J.-M., Michaud, F., Rouat, J., & Letourneau, D. (2003). *Robust sound source localization using a microphone array on a mobile robot.* Paper presented at the Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.

Van Compernolle, D. (1990). *Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings.* Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., Albuquerque.

Van Compernolle, D. (1992a). *Acoustically Robust Speech Recognition.* Katholieke Universiteit Leuven, Belgium.

Van Compernolle, D. (1992b, November). *DSP Techniques for Speech Enhancement.* Paper presented at the In Proc. ESCA Workshop on Speech Processing in Adverse Conditions.

Van Compernolle, D., & Van Gerven, S. (1995). *Beamforming with Microphone Arrays.* Paper presented at the Proceedings of the COST 229 : Applications of Digital Signal Processing to Telecommunications, E.U.

Van Compernolle, D., Van Gerven, S., Broos, W., & Weynants, L. (1991, April 3-4). *A Real-Time Griffiths-Jim Beamformer for Speech Applications.* Paper presented at the In Proceeding of the IEEE and ProRISC Symposium on Circuits, Systems and Signal Processing, Veldhoven (The Netherlands).

Van Veen, B. D., & Buckley, K. M. (1988). Beamforming: a versatile approach to spatial filtering. *IEEE ASSP magazine : a publication of the IEEE Acoustics, Speech, and Signal Processing Society, 5*(2), 4-24.

Ventura, J. C. (1989). *Digital audio gain control for hearing aids.* Paper presented at the Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on.

Waheed, K., Weaver, K., & Salam, F. M. (2002, August 4-7). *A robust algorithm for detecting speech segments using an entropic contrast.* Paper presented at the 45th IEEE international midwest symposium on circuits and systems, Oklahoma.

Wang, A., Yao, K., Hudson, R. E., Korompis, D., Lorenzellii, F., Soli, S., et al. (1996). *Microphone array for hearing aid and speech enhancement applications.* Paper presented at the Application Specific Systems, Architectures and Processors, 1996. ASAP 96. Proceedings of International Conference on.

Wei, J., Du, L., Yan, Z., & Zeng, H. (2003). *A new algorithm for voice activity detection.* Paper presented at the Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on.

Widrow, B. (1975). Adaptive noise cancellation: Principles and applications. *Proceedings of the IEEE, 63*, 1692-1716.

Widrow, B. (2001). A microphone array for hearing aids. *Circuits and Systems Magazine, IEEE, 1*(2), 26-32.

Widrow, B., & Hoff, M. E. (1960). *Adaptive switching circuits.* Paper presented at the IRE WESCON Convention Rececord, New York.

Widrow, B., & Luo, F. (2003). Microphone arrays for hearing aids: an overview. *Speech communication, 39*, 139-146.

Wilson, B. (2003). Digital signal processing applications for hearing accessibility. *Signal Processing Magazine, IEEE, 20*(5), 14-18.

Wirth, W.-D. (2001). *Radar techniques using array antennas.* London: The Institution of Electrical Engineers.

Woo, K.-H., Yang, T.-Y., Park, K.-J., & Lee, C. (2000). Robust voice activity detection algorithm for estimating noise spectrum. *Electronics Letters, 36*(2), 180-181.

Yan, Z., Du, L., Wei, J., & Zeng, H. (2003). *Two-channel microphone array processing for speech enhancement.* Paper presented at the Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on.

Yassa, F. F. (1987). Optimality in the choice of convergence factor for gradient based adaptive algorithms. *IEEE Trans on Acoust., Speech, and Signal Processing, ASSP-35*(Jan.), 48-59.

Yongjian, L., Genmiao, Y., & Shouhong, Z. (2001). *A fast and robust adaptive beamformer.* Paper presented at the Radar, 2001 CIE International Conference on, Proceedings.

Appendices

Appendix A: Matlab Source Code

The Griffiths-Jim beamformer and the VAD based on the time delay are implemented using the Matlab m-file and it is given here. VAD uses the GCC algorithm to calculate the time delay between two microphones. The adaptive filter uses the NLMS algorithm to reduce the noise signal. The updates of the filter coefficients are controlled by the VAD algorithm.

This program requires the pcm3003.out file and c6x_dap.dll in the same directory as this m-file to interface with the DSK. These support files (c6x dsk Matlab version 3.1) can be obtained from author's website:
(http://eceserv0.ece.wisc.edu/~morrow/software/)

```
c6x_daq('Reset');                          % reset DSK
FrameSize = 64;                            %buffer size
c6x_daq('Init', 'pcm3003.out');            %use for TI PCM3003 codec board
c6x_daq('FrameSize', FrameSize);
c6x_daq('QueueSize', 2*FrameSize);
Fs = c6x_daq('SampleRate', 22000);         %use -1 for fixed sample rate
numChannels = c6x_daq('NumChannels', 2);
c6x_daq('TriggerMode', 'Auto');
c6x_daq('TriggerSlope', '+');
c6x_daq('TriggerValue', 0.0);
c6x_daq('TriggerChannel', 1);
c6x_daq('GetSettings');
c6x_daq('Version');

data = c6x_daq('GetFrame');                %read the input from the DSK
P1=plot(data(:,1),'b');
hold on
P2=plot(data(:,2),'m');
hold off
legend('Rnm');
set(gcf,'doublebuffer','on')
grid minor
B=0.9;                                     %forgetting factor
Cmin=0.3;      %Cut off value for the MSC to check if its reverberation or speech
dmax=5;        %if the Max delay is less than this delay then its speech
Snn_old=1;
Smm_old=1;
```

```
Snm_old=1;

while 1 > 0
  data = c6x_daq('GetFrame');                    %signal form each of the two microphones

  %**************************************************************************
  %Step 1: Estimating the spectra of the 2 signals and cross-spectrum
  w1=hamming(FrameSize);
  copy1=data(:,1);
  copy2=data(:,2);
  data(:,1)=data(:,1).*w1;
  data(:,2)=data(:,2).*w1;
  data = fft(data);
  %smothly updating the spectrum recursively at each FFT frame
  Snn_new=(B*Snn_old)+((1-B)*(data(:,1).*conj(data(:,1))));
  Smm_new=(B*Smm_old)+((1-B)*(data(:,2).*conj(data(:,2))));
  Snm_new=(B*Snm_old)+((1-B)*(data(:,1).*conj(data(:,2))));
  Snm_new_temp=Snm_new;
  Snm_new=abs(Snm_new);

  %**************************************************************************
  %Step 2: Estimating the Magnitude Squared Coherence
  MSC=(Snm_new.*Snm_new)./(Snn_new.*Smm_new);
  a_msc=0;
  no=(FrameSize/2)+1;
  for I=1:no,
     a_msc=a_msc+MSC(I);
  end
  a_msc=a_msc/no;

  %**************************************************************************
  %Step 3: Estimating the weight term
   weight=MSC./(Snm_new.*(1-MSC));

   %**************************************************************************
   %Step 4: Estimate the time-delay of arrival d from the GCC
   temp=ifft(weight.*Snm_new_temp);
   temp=real(temp);
   %calculating the maximum value!!
   val=0;
   m_index=0;
   for J=1:FrameSize,
      if val<temp(J)
         m_index=J;
         val=temp(J);
      end
   end
   if m_index>(FrameSize/2)
      m_index=FrameSize-m_index;
   end
```

```
%checking to see if the data is noise or speech signal
if (m_index<=dmax)
   if a_msc>=Cmin
      constant=1;                          %speech
   else
      constant=0;                          %echo
   end
else
   constant=0;                             %noise
end

%*************************************************************************
%adaptive filter based on NLMS
alpha=0.01;
mu=0.5;
norder=8;
nwe=norder+1;
xv=zeros(nwe, 1);
we=zeros(nwe,1);
errv=zeros(FrameSize,1);
for k=1:FrameSize,
   for l=nwe:-1:2
      xv(l)=xv(l-1);
   end
   xv(1)=copy2(k);
   sums=xv'*we;
   err=copy1(k)-sums;
   errv(k)=err;
   for m=1:nwe
      we(m)=we(m)+((mu/(alpha+(xv(m)'*xv(m))))*err*xv(m)*constant);
   end
end
%*************************************************************************
%sending the speech signal to the output
data(:,1)=0;
data(:,2)=errv;
%*************************************************************************
%copying the old data for reference
Snn_old=Snn_new;
Smm_old=Smm_new;
Snm_old=Snm_new_temp;
try
  set(P1,'ydata',data(:,2)) %b
  set(P2,'ydata',data(:,1)) %m
catch
  break;
end
drawnow
end
%*************************************************************************
```

Appendix B: CCS Source Codes

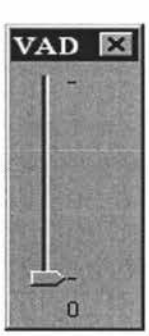
The programs used to carryout the experiments are given here in this appendix. The following support files were used with the programs given here to build a project in the CCS software to communicate with the DSK.

- Cdxdsk.cmd
- C6xdsk.h
- C6xinterrupts.h
- C6xdskinit_pcm.h
- C6xdskinit_pcm.c
- Vectors.asm

B.1. Voice activity detector program

```
//Program used for VAD experiments
//vad_poll_pcm.c VAD program with polling using PCM3003 codec
//using the slider can choose to either turn on or off the VAD.
// 0 - VAD is off the output is the average of both inputs
// 1- VAD is on the output is the average speech inputs, and noise is zero.
float Fs = 16000.0;
//declaration of VAD function and
short vad(float input);
float varf=0.0;                 //a global variable used to store the previous energy value
//slider variable
short out_type = 1;                                     //output type for slider

void main()
{
        float left, right, output;
        float add;
        short switch1=0;                                //voice activity detector switch

        comm_poll();                                    //init DSK,codec,McBSP
        while(1) {                                      //infinite loop
                left = (float) input_left_sample();
                right = (float)input_right_sample();
                add=(right+left)*5E-1;          //contains the addition of the input values
                if (out_type == 0) {
                        output=add;             //average of the input goes to output
                }
                else if (out_type == 1) {
```

```
                    switch1=vad(right);                 //voice activity detector
                    if (switch1==1) {                   //speech
                            output=add;
                    } else if(switch1==0) {             //noise
                            output=0;
                    }
            }
            output_sample((short)(output));             //overall output result
        }
}

//*****************************************************
//voice activity detector
//*****************************************************
short vad(float input) {
        short temp=0;
        float b=0.9, vari=0.0;

        vari=varf;
        varf=(b*vari)+((1-b)*input*input);   %energy calculation
        if (varf>=15E7) temp=1;              %comparing with the threshold
        else temp=0;

        return temp;
}

//*****************************************************
GEL file
//*****************************************************
//VADswitch.gel Slider for outputting the speech
//after or before the Voice activity detector

menuitem "VAD" //printed on the slider objective

slider VAD(0,1,1,1,outtype)  /*increment by 1,from 0 up to 1*/
{
                //parameters 0 - VAD off, and 1 - VAD on
    out_type = outtype;    /*vary type of output*/
}
```



Switch at 0 - without VAD

Switch at 1 - with VAD algorithm

B.2. Adaptive filter program

```
//Program used for Adaptive filter experiments
//nlms_poll_pcm.c adaptive filter program with polling using PCM3003 codec
//switch is there to turn on or off the adaptive filter process

/* Definitions */
#define Nw1 3                                   //Number of weights

/* Declarations of Global Variable*/
float Fs = 16000.0;                             //sampling rate
float we1[Nw1], xv1[Nw1];
short out_type = 0;                             //output type for slider
short i;

/* Function Prototype */
//declaration of Adaptive filter function
float nlms(float prim, float ref, int nh, float mu, float alpha,float *we,float *xv);
void initial(void);

void main()
{
        float left, right, output;
        float op_one;

        initial();
        comm_poll();                            //init DSK,codec,McBSP
        while(1) {                              //infinite loop
                left = 1;                       //(float) input_left_sample();
                right = 1;                      //(float)input_right_sample();

                //switch for adaptive output or just average output?
                if (out_type == 0) {
                        output=(right+left)*5E-1;
                }
                else if (out_type == 1) {
                        op_one=nlms(right,left,Nw1,5E-1,1E-1,we1,xv1);
                        output=(op_one)*5E-1;
                }
                output_sample((short)(output));         //overall output result
        }
}

//*****************************************************
//Adaptive filtering - NLMS
//*****************************************************
float nlms(float prim, float ref, int nh, float mu, float alpha,float *we,float *xv)
{
        float yn=0.0, xvt=0.0, E=0.0;
```

```
        xv[0]=ref;
        for(i=0; i<nh; i++)     {
                yn+=(we[i]*xv[i]);
                xvt+=(xv[i]*xv[i]);
        }
        E=prim-yn;
        for(i=nh-1; i>=0; i--)  {
                we[i] = we[i] + ((mu*E*xv[i])/(alpha+xvt));
        }
        for(i=nh-1; i>0; i--)   {
                xv[i]=xv[i-1];
        }
        return E;
}

void initial(void) {

        for (i=0; i<Nw1; i++) {
                we1[i]=0;
                xv1[i]=0;
        }
}

/*nlms_switch.gel Slider for outputting the speech after/ before the adaptive filter part*/

menuitem "NLMS" //printed on the slider objective

slider NLMS(0,1,1,1,outtype)  /*increment by 1,from 0 up to 1*/
{
        //parameter 1 - after adaptive filter and 0 - before adaptive filter
        out_type = outtype;     /*vary type of output*/
}
```

Switch at 0 - no adaptive filter (just average of the two input signals)

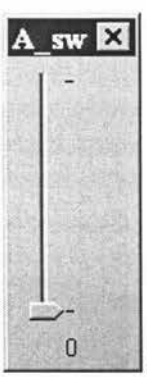
Switch at 1 - adaptive filter output

B.3. Switched Griffiths-Jim beamformer program

```
//Program used for switched Griffiths-Jim beamformer experiment
//Beam_poll_pcm.c Beamformer program with polling using PCM3003 codec
/* Definitions of the no of weight for the adaptive filters */
#define Nw1 30                          //Number of weights coefficients for NLMS1
#define Nw2 50                          //Number of weights coefficients for NLMS2
/* Declarations of Global Variable*/
float Fs = 16000.0;                             //sampling rate
float varf=0.0;
//weight and x vector for NLMS1 and NLMS2
float we1[Nw1], we2[Nw2], xv1[Nw1], xv2[Nw2];
float m_yn=0;
short out_type = 0;                             //output type for slider
short i;                                        //general purpose index variable

/* Function Prototype */
void initial(void);                     // initialisation of weight and x vectors function
short vad(float input);                 //declaration of VAD function
//Adaptive filter function declaration
float nlms(float prim, float ref, float sw, int nh, float mu, float alpha,float *we,float *xv);

void main()
{
        float left, right, output;
        float op_one, op_two, add;
        short switch1=0, switch2=1;             //voice activity detector switch

        initial();
        comm_poll();                            //init DSK,codec,McBSP
        while(1) {                              //infinite loop
                left = (float) input_left_sample();     //left input channel
                right = (float)input_right_sample();    //right input channel

                switch1=vad(right);             //voice activity detector
                if (switch1==1) {               //speech
                        switch2=0;
                } else if(switch1==0) {         //noise
                        switch2=1;
                }
                //switch for beamformer output or just average output
                if (out_type == 0) {
                        output=(right+left)*5E-1;
                }
                else if (out_type == 1) {
                        //speech beamformer: fist filter is used as beam-steering filter
                        //and the second filter is used as adaptive noise canceller
                        op_one=nlms(right,left,switch1,Nw1,9E-1,1E-2,we1,xv1);
                        add=right+m_yn;
```

```
                op_two=nlms(add,op_one,switch2,Nw2,5E-1,1E-2,we2,xv2);
                output=(op_two)*5E-1;
            }
            output_sample((short)(output));          //overall output result
        }
}


//*****************************************************
//initialisation of weight and x vectors
//*****************************************************
void initial(void) {

        for (i=0; i<Nw1; i++) {
                we1[i]=0;
                xv1[i]=0;
        }
        for (i=0; i<Nw2; i++) {
                we2[i]=0;
                xv2[i]=0;
        }
}


//*****************************************************
//voice activity detector- based on energy of the received signal
//*****************************************************
short vad(float input) {
        short temp=0;
        float b=0.9, vari=0.0;

        vari=varf;                              //stores the previous value
        varf=(b*vari)+((1-b)*input*input);
        if (varf>=15E7) temp=1;
        else temp=0;
        return temp;
}


//*****************************************************
//Adaptive filtering - NLMS
//*****************************************************
float nlms(float prim, float ref, float sw, int nh, float mu, float alpha,float *we,float *xv)
{
        float yn=0.0, xvt=0.0, E=0.0;

        xv[0]=ref;
        for(i=0; i<nh; i++)     {
                yn+=(we[i]*xv[i]);
                xvt+=(xv[i]*xv[i]);
```

```
        }
        if (nh==Nw1) m_yn=yn;
        E=prim-yn;
        for(i=nh-1; i>=0; i--)  {
                we[i] = we[i] + ((mu*E*xv[i]*sw)/(alpha+xvt));
        }
        for(i=nh-1; i>0; i--)    {
                xv[i]=xv[i-1];
        }
        return E;
}


//*****************************************************
/*Adaptswitch.gel Slider for outputting the speech
after or before the beamforming algorithm*/
//*****************************************************

menuitem "A_sw"                 //printed on the slider objective

slider A_sw(0,1,1,1,outtype)    /*increment by 1,from 0 up to 1*/
{
                //parameters 0 - the average of the received signal goes to the output
                //           1 - beamformer output
    out_type = outtype;         /*vary type of output*/
}
```



Switch at 0 – without beamformer algorithm (average of the 2input signals)

Switch at 1 – with beamformer algorithm

Appendix C: Speech Recognition

C.1. Visual Basic Source Code

```
'VB application sample : Dictation recognition
'It uses a grammer file, but can also be used to do free dictaion.
'this program can be used to turn on or off two devices (light and radio)
' and can count up to 30
' the grammar file can be modified to have more words as the user required
Public WithEvents RC As SpSharedRecoContext
Public myGrammar As ISpeechRecoGrammar
'boolean variable to turn on or off the speech recognition process
Public fRecoEnabled As Boolean
'boolean variable to handle errors
Public gSAPIPresent As Boolean
'these two boolean variables store the status of the devices.
Public radio_st As Boolean
Public light_st As Boolean

Private Sub cmdExit_Click()
   'Exit the project
   'MsgBox "Thanks for using this program!"
   End
End Sub

Private Sub cmdHelp_Click()
   Label2.Caption = "Click Pause button if you want to take a break from the program, and
Click Start button to continue using the program, or Click the Exit button to exit the program."
   Label4.Caption = "The program starts automatically dont have to click anything."
End Sub

Private Sub cmdStart_Click()
   If fRecoEnabled = True Then
      'must declare for free dictation program
      'myGrammar.DictationSetState SGDSInactive
      'must declare for grammar file program
      myGrammar.CmdSetRuleIdState 1, SGDSInactive
      cmdStart.Caption = "Start"
      fRecoEnabled = False
   Else
      'must declare for free dictation program
      'myGrammar.DictationSetState SGDSActive
      'must declare for grammar file program
      myGrammar.CmdSetRuleIdState 1, SGDSActive
      cmdStart.Caption = "Pause"
      fRecoEnabled = True
   End If
End Sub
```

```
Private Sub Form_Load()
   On Error GoTo SAPINotFound
   'seting the background colors for the form and the label's
   Form1.BackColor = RGB(180, 180, 200)
   Label1.BackColor = RGB(180, 180, 200)
   Label2.BackColor = RGB(180, 180, 200)
   Label3.BackColor = RGB(180, 180, 200)
   Label4.BackColor = RGB(180, 180, 200)
   Label5.BackColor = RGB(180, 180, 200)
   Label6.BackColor = RGB(0, 180, 200)
   Label7.BackColor = RGB(0, 180, 200)
   Label9.BackColor = RGB(180, 180, 200)
   Set RC = New SpSharedRecoContext

   Set myGrammar = RC.CreateGrammar
   'must declare for free dictation program
   'myGrammar.DictationSetState SGDSActive
   'must declare for grammar file program
   myGrammar.CmdLoadFromFile App.Path & "\newnums.xml", SLOStatic
   myGrammar.DictationSetState SGDSInactive
   myGrammar.CmdSetRuleIdState 1, SGDSActive
   fRecoEnabled = True
   gSAPIPresent = True
   radio_st = False
   light_st = False
   cmdStart.Caption = "Pause"
   Exit Sub
'error message handling for the user
SAPINotFound:
   If Err.Number = 459 Then
      MsgBox "SAPI not found"
   Else
      MsgBox "Error encountered : " & Err.Number
   End If

   gSAPIPresent = False
End Sub

Private Sub RC_FalseRecognition(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant, ByVal Result As SpeechLib.ISpeechRecoResult)
   Label1.Caption = "(no recognition)"
End Sub

'displays the speech on a label
Private Sub RC_Recognition(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant, ByVal RecognitionType As SpeechLib.SpeechRecognitionType, ByVal Result As
SpeechLib.ISpeechRecoResult)
   Dim inSpeech As String
   Dim opSpeech As String

   inSpeech = Result.PhraseInfo.GetText
   Select Case inSpeech
   Case Is = "light on"
   '  Green background color
      If light_st = True Then
```

```
        Label9.Caption = "Light is already on"
      End If
      If light_st = False Then
        Label6.BackColor = RGB(0, 200, 0)
        Label6.Caption = "  Light On"
        Label9.Caption = "Light is turned on"
        light_st = True
      End If

    Case Is = "light off"
      'Red backgound color
      If light_st = False Then
        Label9.Caption = "Light is already off"
      End If
      If light_st = True Then
        Label6.Caption = "  Light Off"
        light_st = False
        Label9.Caption = "Light is turned off"
        Label6.BackColor = RGB(200, 0, 0)
      End If
    Case Is = "radio on"
    '  Green background color
      If radio_st = True Then
        Label9.Caption = "Radio is already on"
      End If
      If radio_st = False Then
        Label7.BackColor = RGB(0, 200, 0)
        Label7.Caption = "  Radio On"
        Label9.Caption = "Radio is turned on"
        radio_st = True
      End If

    Case Is = "radio off"
      'Red backgound color
      If radio_st = False Then
        Label9.Caption = "Radio is already off"
      End If
      If radio_st = True Then
        Label7.Caption = "  Radio Off"
        radio_st = False
        Label9.Caption = "Radio is turned off"
        Label7.BackColor = RGB(200, 0, 0)
      End If
    End Select
    opSpeech = "You said: " & inSpeech
    Label1.Caption = opSpeech

    'if Result.PhraseInfo.GetText=="radio on"
      'open "C:\Program Files\Winamp\winamp.exe"
End Sub
```

C.2. Grammar file

The following xml code is used as the grammar file for the speech recognition system, and it has to be saved under the name of "newnums.xml" in the same directory as the Visual Basic program given in Appendix C.1.

```
- <GRAMMAR>
  - <RULE ID="1" Name="number" TOPLEVEL="ACTIVE">
    - <L PROPNAME="number">
        <P VALSTR="radio on">radio on</P>
        <P VALSTR="light on">light on</P>
        <P VALSTR="radio off">radio off</P>
        <P VALSTR="light off">light off</P>
        <P VAL="0">zero</P>
        <P VAL="1">one</P>
        <P VAL="2">two</P>
        <P VAL="3">three</P>
        <P VAL="4">four</P>
        <P VAL="5">five</P>
        <P VAL="6">six</P>
        <P VAL="7">seven</P>
        <P VAL="8">eight</P>
        <P VAL="9">nine</P>
        <P VAL="10">ten</P>
        <P VAL="11">eleven</P>
        <P VAL="12">twelve</P>
        <P VAL="13">thirteen</P>
        <P VAL="14">fourteen</P>
        <P VAL="15">fifteen</P>
        <P VAL="16">sixteen</P>
        <P VAL="17">seventeen</P>
        <P VAL="18">eighteen</P>
        <P VAL="19">nineteen</P>
        <P VAL="20">twenty</P>
        <P VAL="21">twenty one</P>
        <P VAL="22">twenty two</P>
        <P VAL="23">twenty three</P>
        <P VAL="24">twenty four</P>
        <P VAL="25">twenty five</P>
        <P VAL="26">twenty six</P>
        <P VAL="27">twenty seven</P>
        <P VAL="28">twenty eight</P>
        <P VAL="29">twenty nine</P>
        <P VAL="30">thirty</P>
      </L>
    </RULE>
  </GRAMMAR>
```

Appendix D: Paper to be presented

Full paper submitted to Home Oriented Informatics and Telematics 2005 conference.

A Bluetooth Home Design @ NZ: Four Smartness

Olaf Diegel[1], Grettle Lomiwes[2], Chris Messom[2], Tom Moir[2], Hokyoung Ryu[2], Federico Thomsen[2], Vaitheki Yoganathan[2], Liu Zhenqing[2]

[1]Institute of Technology and Engineering
[2]Institute of Information and Mathematical Sciences
Massey University, Auckland, New Zealand

Abstract

Much of the work in the smart house technology has been done on individual technologies, but little has been done on their integration into a cohesive whole. The Bluetooth house project at Massey University in New Zealand, which was initiated in 2002, embraced a systems engineering approach to design a usable smart house, aiming at a complete and integrated solution, which can be customised, based on individual needs, to give elderly people independence, quality of life, and the safety they require. This paper presents how the Massey Bluetooth smart house design project has been carried out and what the smart home may look like in the near future.

Considering current technical feasibility and the advances in other research, it is suggested that for a house to be considered as truly 'smart', four levels of smartness are imperative: *smart sensors*, *smart management*, *smart control*, and *smart appliances*. The Bluetooth house at Massey University incorporates these four smart technologies and allows all these individual technologies to be integrated into a seamless whole. For smart sensing, the project employed Bluetooth technology to connect the whole house, and to locate the user's position. In order to coordinate all the technologies, a smart management system was developed, that is capable of coordinating the information for commands, feedback from smart appliances, and user's location information. It can make intelligent decisions on what to do, or relay necessary information to individual intelligent devices throughout the house. In addition, the medium of communication with the house must be as natural as possible, in order to make it as easy as possible for the occupants of the smart house to interact with and the various smart appliances. A voice-activated universal remote control and a new microphone system are being developed to this end. Finally, the smart house has to provide an enjoyable experience that can promote the uptake of smart house technology by users in the future. An interactive TV environment is being developed to this end.

The Massey Bluetooth house project is not so much aimed at a cutting-edge technology in smart house design, but at integrating technologies into a seamless, cohesive whole through the application of four levels of smartness.

Contact:
Hokyoung Ryu
Institute of Information and Mathematical Sciences
Massey University
Auckland, 1311, New Zealand
Tel: +64 9 4140800 ext. 9140, Fax: +64 9 441 8181
Web: http://www.massey.ac.nz/~hryu
mailto: h.ryu@massey.ac.nz

1. Introduction

Since Mark Weiser of PARC has coined the phrase 'ubiquitous computing', there have been great advances in this research. Much of the enabling technologies in this area, as predicted in his visionary paper (Weiser, 1991), are now to some extent available. For instance, Global positioning systems, Personal Digital Assistants, Bluetooth networks and Radio Frequency Identification networks are all applicable in realising the concept of a ubiquitous computing environment.

It is noted that these technologies have been great successes in each of their individual consumer worlds. A smart home (or smart house), however, asks them to be integrated to a level where the technologies and appliances in the house help make life easier, safer and more enjoyable for the occupants (Rogers and Mynatt, 2003). It poses the three important issues in that a systems engineering approach is needed to make all areas of the smart house work together seamlessly (Jacko and Sears, 2003); the smart house should be transparent to the people in the home (extended from Norman, 2001); and finally multidisciplinary cooperation is required to achieve these goals. It is also note that much work has been done on individual technologies that can be of help in caring for the frail and elderly, but little has been done on their integration into a cohesive whole. The Bluetooth house project at Massey University, which was initiated in 2002, follows this systems engineering approach to develop usable smart house technologies in New Zealand, with collaboration between engineers from electronics, robotics, telecommunication technologies, and psychologists.

Apart from the academic interest, the political and social aspects of New Zealand are also considered in this project. In 2002, the authorities of New Zealand Health sector initiated their strategic approach to provide appropriate health care to the elderly population in New Zealand (Health Sector Strategic Report 2002). The report concluded that New Zealand has a high and increasing elderly population ratio without any support from the other family members and suggested that rest homes with monitoring facilities would be very effective in taking care of this population segment.

The goal of the Massey Bluetooth home project is thus to create a complete and integrated solution, which can be customised, based on individual needs, to give elderly people independence, quality of life, and the safety they require.

2. Challenges in Massey Bluetooth Home Design

The Massey Bluetooth home project, presented here, offers a potentially high opportunity to demonstrate a basic level of smart house technology, focusing on two challenging issues: Integrity and Usability.

2.1. Integration

It can be seen that the key success factor of smart house technology is how well the individual technologies can be integrated to provide a comfortable life in the home. Integration is thus what smart house researchers are ultimately aiming at. The current technologies available still need to be reconfigured for this objective, as there are many interdependency issues that arise as the individual technologies are integrated into a cohesive working application. For example, the Aware Home project at Georgia Tech (see more details in http://www.cc.gatech.edu/fce/ahri) and the project Aura at CMU (see more details in http://www-2cs.cmu.edu/%7Eaura) also focus on the integration of the individual technologies.

The Massey smart house team's approach is thus to identify the opportunities and limitations of current technologies in the home and to introduce a plausible solution for their integration through the use of a Bluetooth network. Some technologies are also being

developed to ensure the integrity of low cost smart house technology in order to meet the market's demands.

2.2. Usability

In the context of work, the key components of usability are recognized as task fit and ease of learning. Current smart houses are often designed from a mechanical view so that poor ease of use and task fit are major barriers to the uptake of such ubiquitous computing technology.

Whilst conventional concepts of usability are equally important, they miss something about the nature of smart home environments, specifically, activities in the home. Smart house researchers presume that many activities in home do not have a clear aim or task objective and may be done simply for the enjoyment they provide. Thus, in the smart house the criteria for usability have mainly to do with the user's experience rather than the user's ability to complete some task (from personal communication with Monk, 2003).

The approach of the Massey Bluetooth home project is thus to identify design principles for ease-of-use and ease-of-learning, previously developed and applied in the workplace, to the problem of configuring and re-configuring networks of devices in the smart house. At the same time new concepts of usability will be identified, building on the work on this topic currently going on in the other ubiquitous computing projects at Massey.

The following sections discuss how these aims are being investigated in this project.

3. The Massey Bluetooth Home Design

Considering current technical feasibility and the advances in other research, e.g., MIT Oxygen project, Home Automated Living and DELTA project, it is presumed that for a house to be considered as truly 'smart', it requires four things:

- Smart sensors: it needs to know who is in the house, where in the house they are, and what special needs or preferences they may have.
- Smart management: it needs a central management system, which is based on the occupants identities and locations, can coordinate all the smart appliances and devices in the house to best fit those occupants' needs.
- Smart control: a speech recognition system that allows the users to communicate with the house in a natural manner, without having to wear headsets or consciously have to activate a microphone, etc.
- Smart appliances: to be a smart home environment, it needs some smart appliances that have enhanced capabilities of the conventional home appliances. For instance, Internet-connected appliances are now within the financial reach of the ordinary consumer, resulting in a range of new services to enhance our lives. At the application level, what sort of smart appliances would be useful in the smart home environment is also an important concern.

The Massey Smart House incorporates these four core technologies to allow all these individual technologies to be integrated into a seamless whole. The following outlines the details of each sub-project.

3.1. Bluetooth Network with Bluetooth Watch

There are many workable networks in the smart house design, e.g., Wi-Fi, GPS, RFID and Bluetooth. One requirement in connecting the house is that the network has the ability to detect where users are within the network. That is to say, the house needs a dynamic network throughout the house that allows devices to communicate with the house management system

that coordinates the information within the house, and the occupants of the house, based on the location of the occupants. This can be achieved through a variety of means including the technologies mentioned above. The indoor tracking system developed by AT&T Laboratories in Cambridge, for example, uses a network of ultrasonic modules to keep track of the users (Harter et al., 1999). The disadvantage with such a system is that the house will only react to occupants wearing the appropriate ultrasonic transmitters.

Bluetooth has the advantage of being an almost ubiquitous technology used in many common appliances such as cell-phones, PDAs, and more, giving it the advantage of having an already available range of transmitters, allowing the smart house to react to a wider range of people. Bluetooth has a range of around 10 meters, which is adequate for certain forms of communication. In contrast, RFID is capable of covering only a relatively short range (around 1 meter,) which is entirely dependent on the radio frequency and the power, but its speed of communication is faster than that of Bluetooth. The Massey smart house adopted Bluetooth as the communication technology in our project, because it allows devices to automatically talk to each other when they come within a certain range, at a relatively reasonable cost, and it is relatively easily extendable to allow communication throughout the entire house, e.g., up to 1000sqm.

Figure 1. Massey Bluetooth watch.

In parallel with the Bluetooth network, a small Bluetooth device embedded in a watch satisfies the requirement of locating the user's position within the network. The project includes the development of a Bluetooth enabled watch/blood pressure monitor as depicted in Figure 1, as well as the design and construction of the Bluetooth ubiquitous network, and the network software.

The Bluetooth network consists of the Bluetooth watch and several small reduced-range Bluetooth transceiver modules that are attached to the ceiling of the room or house to form a grid of linked modules (see Figure 2). The modules are spaced 2 meter apart and each module is set to have a range of 2.4 meters. The grid is connected to a computer running software to deal with the information received from the network. This software includes the ability to track the user over a map of the house, and display the personal data contained in the users watch. As smart appliances, e.g., interactive TV, then begin to be developed and integrated into the system, this information can then be used to intelligently control the appliances.

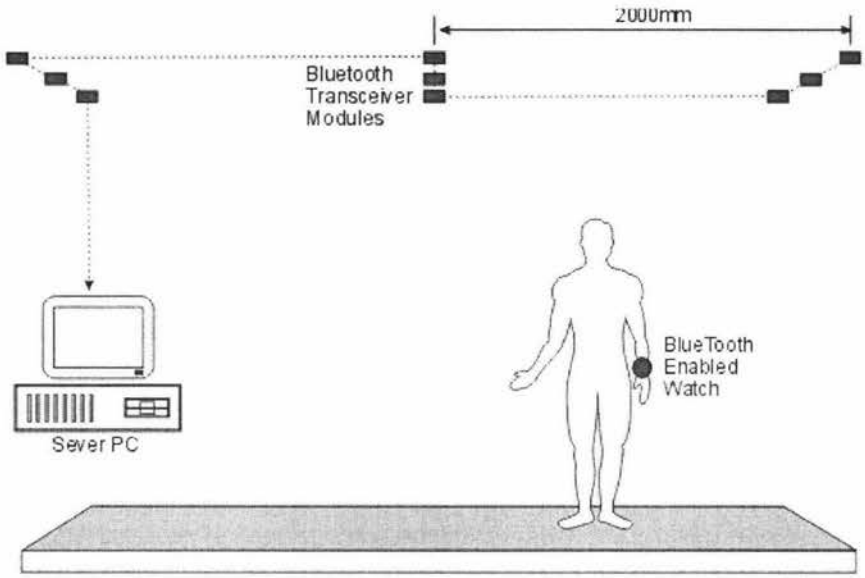


Figure 2. The Massey Bluetooth network.

The Bluetooth-enabled watch is loaded with the users' personal data, and it has a range of 2.4 meters. As the watch is generally between 500mm to 2000mm from the floor (depending on the user's arm position), it is always within range of at least one receiver module but never in range of more than 4 receiver modules. It then becomes simple geometry to determine the location of the user relative to either one, two, three or four modules. This is well within the capabilities of Bluetooth, which is capable of communicating simultaneously to up to seven devices. An additional advantage of using Bluetooth is that standard Bluetooth enabled devices, such as cell-phones, can be used in tracking visitors in the house.

3.2. House Management System

In order to function efficiently and to avoid the duplication of much of the technologies being developed, the house requires a central computer and software package capable of using the information from the speech recognition system for commands, feedback from smart appliances, and user location information. Based on this information, the house management system makes intelligent decisions on what to do, or relaying necessary information to individual intelligent devices throughout the house.

The house management system involves PC based software that, based on the information received from the Bluetooth network, user commands via speech-recognition, and other sensors throughout the house, is capable of making intelligent decisions. Currently, the system is composed of two modules: an expert system to process the received commands and a conversation module that would operate as a 'Chatbot' to converse with the occupants in the home. Further modules such as a visual element to receive some forms of visual communication from the occupants are also envisaged.

Table name	Information held	Description
Commands	Command ID Command Degree of certainty	The commands table contains all the possible commands that the house management system can implement within the house. Each command has a unique command ID and a degree of certainty which indicates that the command received may be unsafe or needs to be confirmed.
Devices	Device ID Device	The device tables contains all the devices that may be accessed through the commands given by the user. Each device also has a unique device ID
Room	Room ID Room	The room table represent all the rooms (or physically isolated location) in the house such as kitchen or the garage. Each location also has a unique room ID
Commands-device	Command ID Device ID	The command device table links the commands and the device, i.e., 'light on' command in the universal remote control is linked to the device 'light'
Room-Device	Room ID Device ID	The room device table holds information about the devices that a particular room has within it.

Table 1. Massey smart house database.

For the working prototype, the house management system is made up of the house database which contains the commands and the rules to operate the devices. The current house database is represented in Table 1. As any process occurs it is checked against the database and the house management system passes the commands to Switching system so as to access appropriate devices.

3.3. Interaction via Speech Recognition

In order to make it as easy as possible for the occupants of the smart house to interact with and the various smart appliances, the medium of communication must be as natural as possible. Other forms of interaction styles, e.g., eye-tracking, may also be applicable for specific appliances or functions, but a good speech recognition system would resolve most of the traditional communication difficulties.

The Massey smart house therefore adopts a speech recognition system, which is capable of analysing spoken language and extracting necessary instructions from it. It involves a speech recognition system that allows the users to communicate with the house management system and a universal remote control for smart appliances, without the need for a headset microphone. The microphone system makes use of beam follower technology (see for details Griffiths and Jim, 1982), and a commercial speech-recognition software is employed, i.e., Dragon™ Naturally Speaking, for a universal remote control that is activated by voice.

3.3.1. Universal Remote Control

Too many remote controls are very problematic in a home, as the user generally intends to control only a particular appliance. That is where a universal remote control comes in. This allows the user to control all their appliances without using separate remote controls.

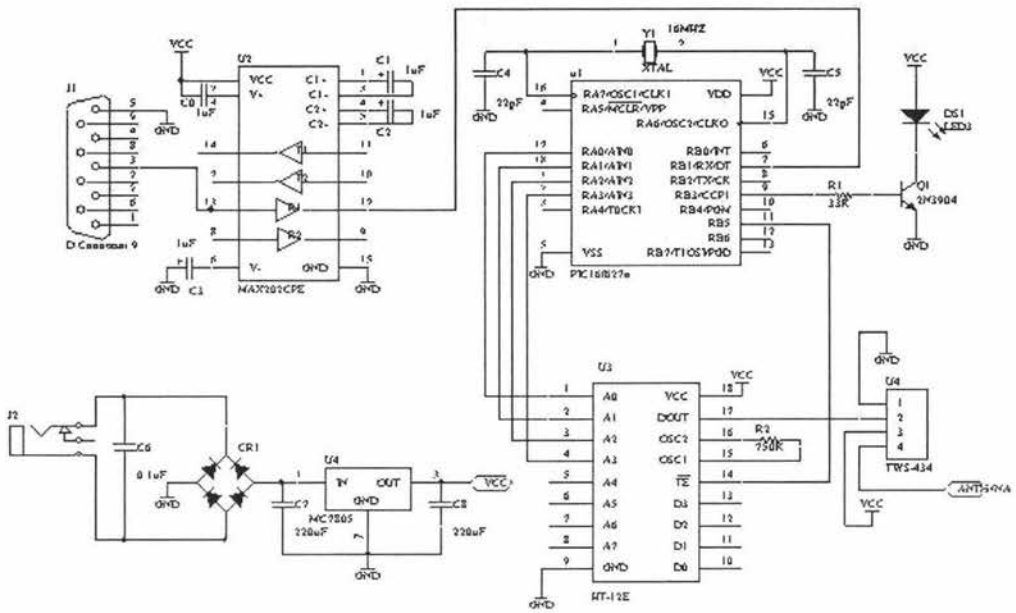


Figure 3. Circuit design of the universal remote control.

Yet, from a 'smart' perspective, the current universal remote control still needs to be advanced from its current interaction style based on keypad-input, to a more intuitive style using speech-recognition technology. A universal remote control is being designed, on which all commands can be activated with human speech rather than through the current user's keypad input on the physical remote.

Based on this account, a voice-activating universal remote control is being developed. The control flow in our design of the universal remote control is described as follows: First, Dragon™ Naturally Speaking recognises what the user said. Then, a program written in LABVIEW™ generates the related binary string which is the same as what the normal remote controller creates when it is clicked. Finally through the RS232 port, the binary strings are transmitted to a microcontroller and the corresponding pulses modulated at 38 kHz are propagated through the infrared LED. The schematic of this circuit design is shown in Figure 3.

In order to dim a light, TWS-434, RWS-434 and HT-12E, HT-12D are used as RF transmitter, receiver encoder and decoder, respectively. LS7631 is a dimmer chip which can adjust the fire angle of a Triac. A signal on the Din pin of HT-12D activates the oscillator which is in turn continuously checked by a decoder. The transmission is not valid until the received address is matched with the decoder's local address. The 555 circuit acts as a pulse generator which can generate different pulse widths which are fed to the LS7631. This voice-activated universal remote control is currently being tested in a laboratory environment.

3.3.2. Microphone Array Beamformer

The same interaction style used with the universal remote control is also used to analyse the occupants spoken language and extract necessary commands for the house management system. A simplistic solution to this problem is to use a headset or body worn microphone to acquire the speech from the user. Yet, as the smart house technology is being aimed toward the frail and elderly population, this is not so practical. Furthermore, the majority of the occupants would prefer not to wear a microphone in their house at all times. Therefore, a microphone system is being developed that allows the users to move around freely and interact easily with the house management system by speaking naturally.

Using a directional microphone is a simple solution to this problem, but they seem to fall short of their objectives in adverse environments. The received signal from the microphone might include the desired speech signal as well as the background noise, which can consist of

several components propagating from different sources such as computer fans, radio, TV, and other talkers. For the speech recognition system to work efficiently a signal to noise ratio of greater than 20dB is typically required.

Another potential solution to this problem is the use a microphone array, which will give the smart house users the advantage of really being able to move freely around the house. The microphone array makes use of beamforming technology to fight against the effects of the acoustic environment.

The selected adaptive algorithm to be used in the smart house is based on a modified version of Griffiths-Jim beamformer (Griffiths and Jim, 1982), which was originated by Van Compernolle (Van Compernolle, 1990). This algorithm has been demonstrated to perform well under noisy and reverberant conditions. The algorithm makes use of two adaptive filters based on Least Mean Square (LMS) (Widrow and Hoff, 1960). Since the LMS algorithm has some drawbacks with stability and selection of the step-size the system will instead be using an adaptive filter based on Normalised LMS (NLMS) (Haykin, 2002).

This speech beamformer makes use of two NLMS algorithms. The first NLMS is updated during a speech segment and the second NLMS is updated during the noise segment. The first one acts as an adaptive beam-steering filter and the second one acts as a filter for the noise. Only one of these NLMS algorithms is updated at a given time. The technique also uses a simple voice activity detector to analyse the received speech signal and determine if it is speech or noise. The corresponding NLMS algorithm is updated depending on the result obtained from this voice activity detector. The above algorithm is in the process of being implemented in real-time on a Texas Instruments digital signal processor.

3.4. A Smart Appliance: Interactive TV and User Experience

If, as discussed above, many of the activities in the home are undertaken for the enjoyment they can provide, i.e., watching TV and cooking, then the smart house has to provide an enjoyable experience that can promote the uptake of smart house technology by users in the future.

For instance, new television technology such as digital television might produce more pleasant experiences for housebound, disabled or elderly people, as it would allow them to access richer and more customisable information from their home. In particular, it is noted that the elderly are the biggest current consumers of television in New Zealand, watching on average more than 5 hours a day (National Statistics of New Zealand). This implies that they have less communicative involvement with their neighbourhood, and that they are at risk of becoming socially isolated from their community.

It is believed that our smart house, together with new television technology, can help lessen this social problem, as the technology facilitates social interaction in the community. In a similar context, Hampton (2003) has set up a wired community to see how much information and communication technologies facilitates community participation and collective action. Yet, the previous research does not propose the development of applications in the interactive TV environment, thus encouraging elderly people to adopt it. Following this work, the research team aims to investigate what kinds of applications in the interactive TV environment would facilitate interaction with their neighbourhood for elderly people, thus extending the concept of smart home into the smart community.

Based on this understanding, an interactive Java™ TV environment with the voice-mediated technology is being developed. The main functionality of it is to enhance the broadcast and viewing experience by providing such features as programming information and chatting with friends while watching TV programmes. This chatting facility can increase social ties between people, even while they are watching television. A prototype is being designed and implemented.

4. Conclusions and Future Work

This paper presented how the Massey Bluetooth smart house design project has been carried out and what a smart home integrated with these technologies may look like in the near future. Currently, the Massey Bluetooth house is only working in a laboratory environment. The house management system, the universal remote control, the beamformer microphone and the interactive TV are still being implemented or tested in this same laboratory environment. The evaluation of both the individual systems and the integrated system has been planned.

This paper has not discussed other smart house issues such as accessibility, emotionality, privacy, security, and sociality, along with the technological approach. We did not intend to trivialise these issues; however, as the current project aims at building a physical house for people to use as a model house for future living, the issues were not included in the immediate project, but will be studied as part of future work.

In conclusion, the Massey Smart House is not so much aiming at a cutting-edge technology in smart house design, but at integrating technologies into a seamless, cohesive whole, and drawing a picture of the technological home in the New Zealand environment as well as developing some business ideas with industry partners such as construction companies, appliance companies, and the government. This research is also intended to communicate the concept of the smart house to the public, encouraging people to access our facility, and thus feel the added benefits of integrating smartness into the home. In the end, the main beneficiaries of this project will be our elderly population who want to retain their independence, and their families and friends who can be secure in the knowledge that they are safe, well and comfortable. The health sector will thus benefit by being able to more effectively help and monitor people in their care. There will also be flow-on benefits for the construction industry, appliance industry, and for people who wish to improve their quality of life.

Bibliography

Griffiths, L. J. and Jim, C. W. (1982) In *IEEE Trans. Antennas Propagation*, pp. 27-34.

Hampton, K. N. (2003) *Grieving for a lost network: collective action in a wired suburb, The Information Society,* **19,** 417-428.

Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P. (1999) *The anatomy of a context-aware application., ACM/IEEE International Conference on Mobile Computing and Networking*, 59-68.

Haykin, S. (2002) In *Adaptive Filter Theory*, (Ed, Kailath, T.) Prentice Hall, Inc., Upper Saddle River, New Jersey.

Jacko, J. A. and Sears, A. (Eds.) (2003) *"The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications"*, Lawrence Erlbaum Assoicates Publishers, Mahwah, NJ.

Norman, D. (2001) In *Communications of the ACM*, pp. 36-37.

Rogers, W. A. and Mynatt, E. D. (2003) In *"How can technology contribute to the quality of life of older adults?"* (Ed, Mitchell, M. E.), pp. 22-30.

Van Compernolle, D. (1990) In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90.*Albuquerque, pp. 833-836 vol.2.

Weiser, M. (1991) *The computer of the 21st century, Scientific American,* **265,** 66-75.

Widrow, B. and Hoff, M. E. (1960) In *IRE WESCON Convention Rerecord,* New York, pp. 96-104.