Multi-microphone Speech Enhancement Technique using a Novel Neural Network Beamformer

A thesis presented in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
Engineering

at Massey University, Albany,
New Zealand.

Vaitheki Yoganathan

2014

Abstract

This thesis presents a novel speech enhancement algorithm to reduce the background noise from the acquired speech signal. It introduces an innovative idea for the speech beamformer using an input delay neural network based adaptive filter for noise reduction.

Speech communication is considered as the most popular and natural way for humans to communicate with computers. In the past few decades, there has been an increased demand for speech-based applications; examples include personal dictation devices, hands-free telephony, voice recognition for robotics, speech-controlled equipment, automated phone systems, etc. However, these applications require a high signal-to-noise ratio to function effectively. The background noise sources such as factory machine noises, television, radio, computer or another competing speaker, often degrade the performance of the acquired signals. The problem of removing these unwanted signals from the acquired speech signal has been investigated by various authors. However, there is still room for improvement to the existing methods.

A multi-microphone neural network based switched Griffiths-Jim beamformer structure was implemented using the Labview software. The conventional noise reduction section of the Griffiths and Jim beamformer structure was improved with a non-linear neural network approach. A partially connected three-layer neural network structure was implemented for rapid real-time processing. The error back-propagation algorithm was used here to train the neural network structure. Although it is a slow gradient learning algorithm, it can be easily replaced with other algorithms such as the fast back-propagation algorithm.

The proposed algorithms show promising noise reduction improvement over the previous adaptive algorithms like the normalised least mean squares adaptive filter. However, the performance of the neural network depends on its chosen parameters such as learning rate, amount of training given, and the size of the neural network structure. Tests with a speech-controlled system demonstrate that the neural network based beamformer significantly improves the recognition rate of the system.

Acknowledgements

First of all, I am truly grateful to my former supervisor Associate Professor Tom Moir for his guidance throughout my studies. I have learned a lot from him, and this research wouldn't have been possible without his knowledge and advice. I would like to thank my current supervisors, Dr. Fakhrul Alam and Dr. M. A. Rashid, for their guidance in writing my thesis and their support during my time at Massey University.

This research was supported by the Technology for Industry Fellowships New Zealand scholarship for the first three years of my studies. This scholarship was awarded to work with the Vibration Consultants Company to develop a noise canceller for the speech-controlled instrument. Also, I would like to thank the company owners, Dennis Page and Elisabeth Page, for allowing me to work with them as part of this scholarship programme. I specially want to thank the company mentor, Darryl Ovens, for his guidance and support during this time.

I would also like to thank my family and friends for being there for me and encouraging me throughout my studies. I am also extremely grateful to my fellow colleagues at the university for their on-going support.

Table of Contents

List of Publications

Published conference papers:

1. Yoganathan, V., & Moir, T. J. (2010a, 24-28 Oct). Multi-microphone adaptive neural switched Griffiths-Jim beamformer for noise reduction. Paper presented at the 10th IEEE International Conference on Signal Processing (ICSP), Beijing, China.
2. Yoganathan, V., & Moir, T. J. (2010b, 21-23 Oct). Speech enhancement using microphone array neural switched Griffiths-Jim beamformer. Paper presented at the International Conference on Wireless Communications and Signal Processing (WCSP), Suzhou, China.
3. Yoganathan, V., & Moir, T. J. (2010c, 10-13 May). Speech enhancement using a nonlinear neural switched Griffiths-Jim beamformer. Paper presented at the 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA), Kuala Lumpur, Malaysia.
4. Yoganathan, V., & Moir, T. J. (2008, 2-4 Dec). Switched Griffiths-Jim beamformer using the affine projection algorithm. Paper presented at the 15th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand.

List of Figures

List of Tables

List of Abbreviations

ADALINE	Adaptive Linear Element
ANC	Adaptive Noise Canceller
ANN	Artificial Neural Network
APA	Affine Projection Algorithm
DLL	Direct Link Library
FB	Feed-Backward
FC	Fully Connected
FF	Feed-Forward
FIR	Finite Impulse Response
GCC	Generalised Cross Correlation
GJBF	Griffiths-Jim Beamformer
GSC	Generalised Sidelobe Canceller
IDNN	Input Delay Neural Network
LMS	Least Mean Squares
MLP	Multi-Layer Perceptron
MSC	Magnitude Squared Coherence
MSE	Mean Square Error
NFC	Non-Fully Connected
NLMS	Normalised Least Mean Squares
NN	Neural Network
RLS	Recursive Least Square
SDK	Software Development Kit
SGJBF	Switched Griffiths-Jim Beamformer
SNR	Signal-to-Noise Ratio
SRHR	Speech Recognition Hit Rate
TDNN	Time Delay Neural Network
VAD	Voice Activity Detector
VMS	Vibration Monitoring System

List of Symbols

$d(n)$	Primary input sample value at time *n*
$e(n)$	Output error sample value at time *n*
G_1, G_2	FIR Transfer functions of the speech signal
h1, h2	Learning rates for the Neural Network
H_1, H_2	FIR Transfer functions of the noise signal
$k(n)$	Gain vector in the RLS algorithm
n, k	Discrete-time or number of iterations
N	Number of filter weights
M	Projection order for APA
$x(n)$	Reference input sample value at time *n*
$\vec{x}(n)$	Tap-input regression vector consisting of x(n), x(n-1),…., as elements
$w(n)$	Weight value at time *n*
$\vec{w}(n)$	Weight vector consisting of w(n), w(n-1),…., as elements
$y(n)$	Adaptive filter's output value at time *n*
μ	Step-size parameter in LMS/NLMS algorithm
γ	Small positive constant in LMS algorithm
δ	Regularisation parameter in APA
λ	Forgetting factor used in RLS algorithm

CHAPTER 1

INTRODUCTION

1.1 Introduction

Speech or spoken language interface is one of the new evolutions in user interface technology, which will eventually replace some of the functions in traditional interfaces. Examples of other popular natural user interfaces include touchscreen, gesture recognition (Kavanagh, 2012), brain-machine interface, and motion based interface, to name a few. Among these methods, using speech is the most basic and the easiest way to interact with computers, rather than using other means like a keyboard or mouse. Besides, it is the preferred method for humans to communicate between themselves and they would prefer to communicate in the same way with computers. As a result, speech is currently becoming the most popular method for us (humans) to interact with electronic devices (Kamm, Walker, & Rabiner, 1997; Weiyuan, 2010).

Automatic speech recognition is the process of automatically converting the acoustic speech signal received by the microphone into a sequence of words for the computer or machine to use (Anusuya & Katti, 2009). Speech recognition technology has been around for decades. Early published work on this area was carried out at the Bell Laboratories in the 1950's. They developed the "Audrey system", but it was only capable of recognising isolated digits spoken by a single user (Davis, Biddulph, & Balashek, 1952). The improvements in the area of speech recognition can be likened to a baby learning; as time went on it became better and better. Even up until the 1990s, it was almost a fantasy to think of interacting with machines using speech. Advanced research done over the past 60-odd years on speech recognition technology, and vast improvements in computer hardware, have made it feasible to make this fantasy into a reality.

The early development stages of speech recognition technology were aimed mainly at applications used by disabled people (Noyes, Haigh, & Starr, 1989). However, this technology is now available for everyone to use in their day-to-day applications (Srinwasan & Brown, 2002). Currently, using speech to control products is one of the main trends in new technologies. Examples of these speech-based applications include voice controlled home automation systems (Portet, Vacher, Golanski, Roux,

& Meillon, 2013), voice control in cars (Hansen, Kim, & Angkititrakul, 2008; Thumchirdchupong & Tangsangiumvisai, 2013), hands-free voice control for televisions (Papp, Saric, & Teslic, 2011), voice controls for Smartphone applications (Dobler, 2000), voice recognition for robotics (Martinez, Ubeda, Ianez, Azorin, & Perez-Vidal, 2013), and voice control in motion based video games, to name a few. In particular, the leading mobile phone company Apple, introduced a voice-control application called Siri™ for their iPhone (Daw, 2011). Siri™ assists the user with speech queries and looks up information like a virtual personal assistant. Sensory went a step further, and designed their TrulyHandsfree™ Voice Control system to handle commands in hostile ambient noise (Jr, 2011).

A new wearable computer with head-mounted display has recently been released by Google, called Google Glass™ (Topolsky, 2013). This uses voice command to control the system, and is the main feature among others. In addition to these, a hands-free voice search option in Google Chrome™ web browser has been recently introduced by Google (Berkens, 2013). This is a conversational search engine that takes a speech signal and provides the searched results by speech (Ye-Yi, Dong, Yun-Cheng, & Acero, 2008). Spoken language interface in these products has come a long way since the original introduction of speech recognition technology. However, their use in the real world's noisy situations is still somewhat limited (Edwards, 2013).

Most speech-controlled systems are designed to be used indoors in a relatively quiet environment with a close talking microphone. However, using a close talking microphone will restrict the user's movements and ties the user down to the device. However, most users do not like the idea of being constrained to a device; they often prefer to move freely. A hands-free solution can be achieved by using a microphone array system in the front-end of the application. However, as the distance between the microphone and the speech signal increases, the acquired signal becomes degraded by background noise and reverberation signal. As a result, the recognition rate of the speech recognition system is affected by the degraded speech signal.

The performance of speech recognition systems often depends on many conditions such as environmental noise levels (Gong, 1995), size of the vocabulary, and variability in speech, such as speakers' pronunciation or their speaking style, and pace of speech (Benzeghiba et al., 2007), to name a few. These problems are caused by the inconsistency between the background environments in the training period and the testing period. It is somewhat possible to train the system for a person or a specific noisy environment, to improve the accuracy. However, when the trained situation changes from the test condition it will reduce the performance. As a result, recently there has been a lot of interest in improving the performance of speech recognition systems in these adverse environments. This problem has been identified as robust speech recognition (Ramírez & Górriz, 2011). The current solutions to this problem vary from improving the speech recognition algorithms to handle a noisy speech signal, to using a pre-processing system such as speech enhancement algorithms to improve the speech signal (Urmila & Vilas, 2013).

Research on the speech enhancement area has been around for decades (Benesty, Makino, & Chen, 2005). Depending on the application's requirements, the main objective of improving the acquired signal can vary, from improving the quality of speech to improving the intelligibility of speech. It is quite difficult to choose one algorithm that can improve the quality of speech and its intelligibility. There is often a trade-off between these factors. Some speech enhancement algorithms can achieve a substantial noise reduction but distort the speech in the process. These algorithms are best suited for applications that can tolerate a little distortion. Also some algorithms work especially well for stationary noise, others for non-stationary noise, or competing talking. Therefore, it is not possible to propose a versatile solution that can be used with different practical applications. A hybrid noise reduction system needs to be designed depending on the requirements of the target applications and the acoustic environment.

1.2 Statement of the Problem

As discussed in the introduction, there have been outstanding recent developments in the field of speech recognition technologies. Even though speech-based systems are used in many everyday activities, they are hardly used in industrial applications. The user could benefit a great deal by using speech to control systems in the industrial environment, as it will make the user's hands free to perform other important tasks. A potential application of the speech-controlled system is vibration monitoring systems that are used by industry to monitor the condition of rotating machinery. The current trend in speech-based applications shows that they are capable of handling day-to-day activities in relatively quiet environments. However, the uptake of this technology in industrial environments is still quite rare, as speech recognition systems often struggle to function effectively in noisy environments. Since the speech-based interfaces can add great value to many applications, it is essential to develop a noise cancelling algorithm that can improve the performance of the current speech recognition systems and allow them to perform satisfactorily in a noisy industrial setting. The problem of improving the quality of the acquired signal in a noisy environment has been addressed by many researchers. Speech enhancement methods such as speech beamformers are often used to improve the performance of the acquired speech signal in an adverse acoustic environment. However, there is still more research that needs to be done to improve their current performance in real-world environments.

This research will endeavour to develop and implement novel algorithms to improve the performance of the current speech recognition systems in noisy industrial environments, for control applications such as hands-free control of a real-life industrial vibration monitoring system.

1.3 Research Objectives

The following steps are taken to find a solution to the problem discussed in section 1.2.

- Conduct a thorough review of the relevant publications in the speech enhancement area.
- Develop a new hybrid method using a certain speech enhancement algorithm that will allow hands-free distant talking speech recognition.
- Implement the proposed algorithm in a software program and evaluate the performance of this algorithm with other commonly used methods.
- Expand this system to operate with more microphones and examine its performance.
- Implement a speech-controlled application. The target application used here is a vibration monitoring system. However, it can be modified for any command and control application.
- Test both systems together and observe the effect of the noise reduction algorithm in the speech-controlled system.

1.4 Contribution of the Thesis to Knowledge

A major contribution of this research is the development and implementation of a non-linear adaptive filter based on an input delay neural network (IDNN). This novel neural network (NN) adaptive filter was investigated with the switched Griffiths-Jim beamformer (SGJBF) structure. This switching adaptive filter approach has not been investigated before using the NN adaptive filter. The current research on NN based noise reduction concentrates mainly on the adaptive noise cancelling concept. However, this adaptive noise canceller (ANC) can only be used in applications where the noise-alone reference signal can be acquired, and this is a difficult task to achieve in the real-world environment. The SGJBF structure was introduced as an improvement to the conventional adaptive noise cancelling concept. This structure uses two adaptive filters. The proposed non-linear adaptive filter is introduced as a possible solution to the noise cancelling problem. Real-time implementation of this algorithm has been carried out using Labview software.

In addition to the above contribution, the NN noise canceller is extended to a four-channel microphone beamformer system, whereas most of the work in this area only concentrates on dual-microphone systems.

An improved version of the automatic variance estimator is introduced here as a voice activity detector (VAD) to control the update of the adaptive filters. This algorithm calculates the time varying variance of the input speech signal, using a recursive sample-by-sample approach. An improvement to the existing method is made by calculating a moving average of the variance output. This solves the problem of having the start and end point of the speech cut off by a sudden change in the signal. The improved version gives a much smoother variance output.

Another contribution made in this thesis is the idea of developing a computer-based speech-controlled vibration monitoring system. Traditionally, these systems are hand-held devices, which require the user's hands to control the system. In situations where the user's hands are required to do other tasks like holding a ladder, this system becomes difficult to use. The idea of using speech to control the system has not been tried before in this area. The basic idea of using the speech-controlled system in a noisy situation has been explored in this thesis with the use of the proposed speech enhancement system.

1.5 Outline of the Thesis

The remaining chapters of this thesis are organised as follows:

Chapter 2 gives a brief introduction to the area of speech enhancement. It gives an overview of the previous approaches done in this area and discusses the problems with using these algorithms. In particular, the adaptive noise cancelling concept and microphone array beamformer algorithms are explored. Different adaptive filter algorithms are explained. Previous research on NN-based noise reduction is discussed here. Finally, various speech detection algorithms that can be used with the speech enhancement algorithm are discussed.

Chapter 3 introduces the proposed NN based beamformer for speech enhancement. In this chapter, three main sections of the SGJBF are explained in detail. The beam-steering section makes use of the traditional normalised least mean square (NLMS) algorithm. The ANC uses the proposed non-linear IDNN noise canceller. A VAD based on recursive variance calculation is implemented, to control the update of adaptive filters. Software implementation of the proposed algorithm in the Labview software is also given, together with the methodology.

Chapter 4 evaluates the performance of the developed speech enhancement algorithms. First, the VAD algorithm is tested with real-world recorded data. Then, the NN system is tested using the partially connected and fully connected structures. This adaptive filter algorithm is tested with a simulated transfer function and real-world recording, and then the beamformer algorithm is tested with several scenarios of the real-world environment. The proposed algorithm is also compared with other common adaptive filter methods (NLMS, Least Mean Square (LMS), Affine Projection Algorithm (APA), and the Volterra filter). The dual-microphone method is extended to a four-microphone method, and comparison of its performance under different background noise is also presented here.

Chapter 5 introduces a speech-controlled instrument and evaluates its performance with the NN and NLMS based beamformer algorithm. A vibration monitoring system is used here as a target application. This system is evaluated together with the beamforming system in the real-world setting, under different types of noisy background conditions.

Chapter 6 gives the conclusion to the current research and makes suggestions for future research to improve this work.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a brief overview in the area of speech enhancement, adaptive beamforming for noise reduction, adaptive filters and their adaptive algorithms, neural network (NN) based noise reduction, and voice activity detectors (VADs).

2.1 Survey of Speech Enhancement Area

Speech enhancements in a noisy environment have received a considerable amount of attention in order to improve speech-based applications (Davis, 2002). Applications that require speech enhancement vary from hands-free telephony (Ryan & Goubran, 2003), speech-controlled equipment, automated phone systems, applications for hearing impaired individuals (Fink, Furst, & Muchnik, 2008; Qidwai & Shakir, 2012; Vanden Berghe & Wouters, 1998; Widrow & Luo, 2003), personal dictation devices (Wormek, Ingenerf, & Orthner, 1997), to name a few. Performance of these applications strongly depends on acquiring a clean speech signal, and this becomes complicated when there are background noise and reverberations that interfere with the speech signal. This noise could consist of several components propagated from different sources such as fan, motor, engine noise, air conditioner, audio equipment, or a competing talker. These interferences reduce the accuracy and performance of the speech-based applications.

A close talking microphone is usually used with the speech-controlled applications to acquire good quality speech. Currently, there are various headsets that are available in the market with an ambient noise cancelling facilities for applications such as gaming. These devices use the active noise cancelling approach to reduce the noise. However, it is well-known that in an adverse acoustic environment, using a close talking microphone alone is not enough to obtain a clean speech signal. Also, using a headset will restrict user movement. Therefore, a sophisticated speech enhancement technique has to be used with the microphones to improve the quality of the acquired speech signal. Nowadays, manufacturers are forced to take account of noisy environments in their applications' functionality. Otherwise, they face the problem of their systems not being able to work properly in these situations or they restrict the use of their system to relatively quiet environments.

Various noise reduction methods have been developed in the past for enhancing the noisy speech signals (Juang & Tsuhan, 1998). These methods can be classified by the number of microphones used (single or multi-microphone methods), or processing of data in time or frequency domain, or by the acoustic characteristics of the noise being reduced. For example, different types of algorithms have to be used to reduce stationary noise, non-stationary noise or another competing speaker (cocktail party problem), so there is no single algorithm that can solve all the problems. Therefore, choosing a suitable speech enhancement algorithm depends on the environment where it will be used.

Noise reduction is similar to filtering, in the sense that both aim to recover the original signal from the noise. Classical filters can be used in situations where the frequency of the noise is separable from the frequency of the speech signal. For example, it might be possible to use classical filters like low-pass filters, high-pass filters, or band-pass filters to reduce the noise. However, they cannot be used in situations where the frequency of speech and noise are overlapping, and also these filters are rarely optimum in producing the best estimate of the signal. In early days, several optimum filtering techniques, such as Wiener filters and Kalman filters, were used to reduce the noise, and they are based on fixed filter weights. These methods often distort the speech signal to some degree, and inevitably allow a bit of noise signal.

Norbert Wiener was one of first researchers to work on filtering in the 1940s (Wiener, 1949). He introduced the Wiener filter algorithm to minimise the mean square error between the estimated signal and the desired signal. This algorithm requires prior statistical information about the signal. A Wiener filter cannot be used in cases where the statistical information cannot be obtained. Apart from this, this filter is also limited to stationary noise. It has also been found that the more the noise is being reduced, the more the speech signal is being distorted (Jingdong, Benesty, Yiteng, & Doclo, 2006). However, a modified version of this filter is still being used these days in several applications (Han et al., 2012; Qi, 2008).

Another popular method used for speech enhancement is signal subtraction. Early work on this area can be found in Boll (1979). It is a well-known algorithm still used nowadays, due to its efficiency and simplicity. The basic idea is to estimate the power spectrum of the noise signal and subtract it from the spectrum of the noisy speech signal. This algorithm requires a high quality estimation of the noise spectrum to function effectively. This method requires the input signal to noise ratio (SNR) to be positive, when measured in decibels for most of the frequency range (Hu & Loizou, 2007). However, signal subtraction methods are still being successfully used to reduce stationary noises (Fukane & Sahare, 2011).

On the other hand, microphone array systems are used to reduce non-stationary and very strong interference signals. The main advantage of using more than one microphone is the availability of spatial information about the source (i.e. the ability to find the location of the source). Multi-microphone methods include source separation, coherence based algorithms and beamforming. Blind source separation methods are being used for separating two speech signals. Beamforming methods usually depend on statistical model based algorithms and they work on the basis of reducing the mean-square error between the input and output. Coherence based methods do not depend on noise statistics; they are often used in hearing aids and cochlear implant devices.

Performance of most of these speech enhancement methods relies heavily on the output of a speech detection algorithm. Choosing an appropriate VAD algorithm for a particular application depends on the noise characteristics and whether it is possible to distinguish between the wanted speech and unwanted noise signal. It is a difficult task as the noise characteristics may vary from application to application. They could be any noise like stationary or non-stationary noise, white noise, pink noise, coloured noise or a competing talker like in a cocktail party situation. Different environments make it impossible to choose one algorithm that can apply to all applications. There are other speech enhancement methods that exist, like the subspace algorithms that do not require a VAD algorithm.

The rest of the chapter will summarise the possible speech beamformers used for noise reduction that are beneficial to speech enhancement.

2.1.1 Beamforming Techniques

Microphone array systems use beamforming techniques to reduce the level of background noise signals while minimising distortion to the desired 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 (Van Veen & Buckley, 1988). The basic idea of beamforming came from different backgrounds such as sonar and radar. However, acoustic beamforming requires considerably different solutions. An array of microphones is distributed in a certain area, to capture the desired signal. The outputs of these microphones are processed to suppress the unwanted signals coming from the undesired directions.

Depending on how the filter weights are chosen, beamformer algorithms can be classified as either fixed beamforming or adaptive beamforming. A fixed beamformer is designed to focus on a targeted direction, independent of the interfering signals. These types of systems are very robust and require minimal processing power. However, they need many microphones to achieve high directivity and good noise reduction. Conversely, an adaptive beamformer is designed to discard the interfering signals by introducing nulls in the direction of their arrival. These types of systems are able to attain a high noise reduction with a small number of microphones but at a cost of higher processing power.

The simplest method of all microphone array beamforming is the delay-and-sum beamformer (Flanagan, Johnston, Zahn, & Elko, 1985). Figure 2.1 illustrates a conceptual diagram of the delay-and-sum beamformer for N number of microphones. Each microphone's signal is delayed by an amount of time proportional to the distance between a known target and that microphone, which is done to align the microphone signals. Then all of these delayed signals are added together to obtain a single large signal array output. The idea behind the summing process is that the desired speech signals will have the same phase so they will add together; while the interfering signals, with a different phase, will not add together. This is assuming that noise signals will not be coming from the same position as the desired signal; thus the noise signals are not coherent.

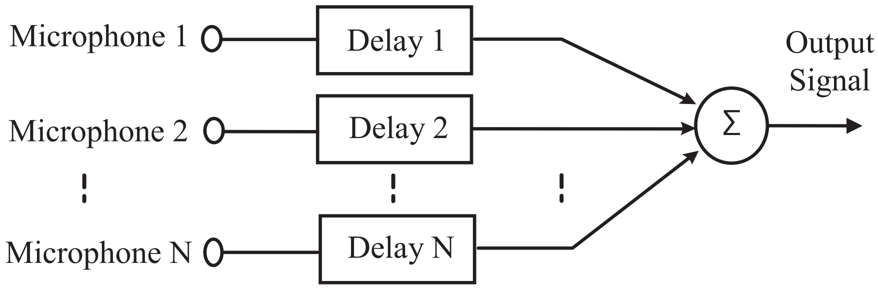


Figure 2.1. Delay-and-Sum Beamformer

The total speech power in the output signal will be multiplied by the number of microphones in the array, while the total noise power in the output signal will remain about the same as the noise power of one microphone. As a result, the SNR will be increased. A major disadvantage of using this beamformer structure is that it requires many microphones to obtain a considerable amount of improvement in the SNR. Also, it only enhances the signal in the direction to which the array is currently steered, and it does not reduce the interference itself.

An alternate method to achieve improved performance is to multiply the received signals with different gain factors (or weights) before summing the signal. An example of this kind of method is the Frost beamformer (Frost, 1972). Figure 2.2 illustrates a conceptual diagram of this beamformer for N number of microphones. The Frost beamformer uses an adaptive filter algorithm to adjust the weight vectors (W) to minimise the noise while maintaining a chosen frequency response in the direction of the desired speech signal. This algorithm is also known as linearly constrained minimum variance algorithm.

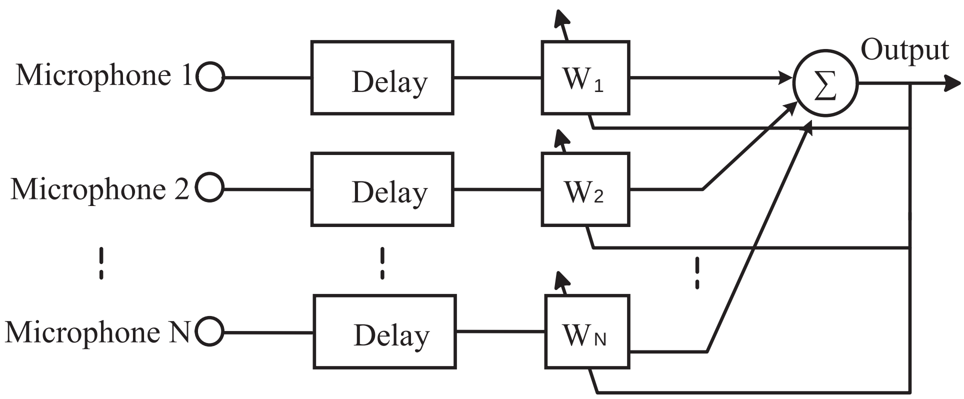


Figure 2.2. Frost Beamformer

A study done by Raykar, shows that using a beamformer improves the SNR of the output signal when compared to just using the directive microphone (Raykar, 2001). It also confirms that the Frost beamformer performs better than the simple delay-and-sum beamformer. This beamformer algorithm is found to be effective in the presence of strong interferers, as long as the interferers are uncorrelated with the desired source.

The concept of the Adaptive Noise Canceller (ANC) was first proposed by Bernard Widrow in 1975 (Widrow et al., 1975). The fundamental technique behind this algorithm seeks to improve the distorted speech by using an adaptive filter to suppress the noise components while leaving the desired speech signal unchanged. In order for this idea to work, at least two-microphone signals are required. A conceptual diagram of this adaptive noise cancellation technique is shown in Figure 2.3.

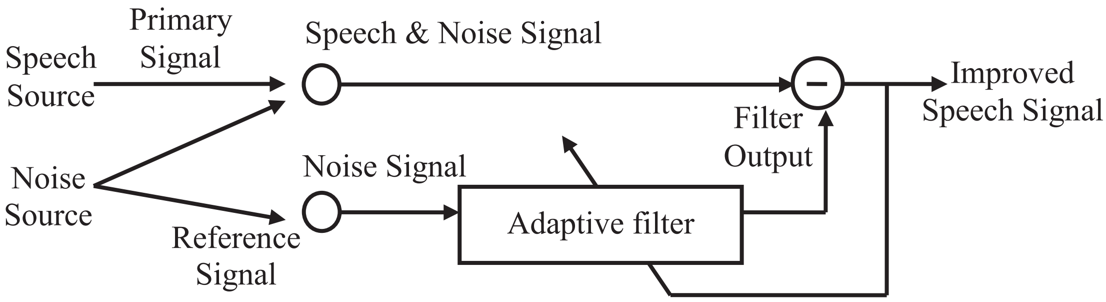


Figure 2.3. Adaptive Noise Canceller Structure

In this figure, the first microphone is positioned near the noise source and the second microphone is positioned to acquire 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 minimise the noise components while leaving the desired speech unchanged. In addition, an adaptive filter algorithm is used to align the signals in order to achieve optimum noise reduction, by minimising the error between a desired signal and the received array output. This is a noise reduction technique which attempts to use multiple signal sources to remove the noise components from the acquired signal.

In order for the noise to cancel out in the subtraction process, this technique requires that the noise entering the two microphones must be coherent. In order for the noise signal in both channels to be coherent, the microphones have to be close together. However, when the microphones are kept close together, it is impossible for the second microphone not to acquire the speech signal. If both microphones acquires the desired speech signal, in the subtraction process, this speech is cancelled with the noise; this is the main drawback of this technique. For this technique to work, the two microphones have to be kept far apart so only one of them can acquire the desired speech signal. At the same time, they have to be close enough, so the noise signals in both microphones are coherent.

An improved version of this structure was proposed by Griffiths and Jim in the 1980s (Griffiths & Jim, 1982). Their technique is known as the Generalised Sidelobe Canceller (GSC) or Griffiths-Jim beamformer algorithm. Figure 2.4 shows the basic structure of the Griffiths-Jim beamformer using two microphone inputs. Before the addition and subtraction process, a beam-steering algorithm can be used to virtually steer the microphone to the speaker's position (i.e. not physically steered).

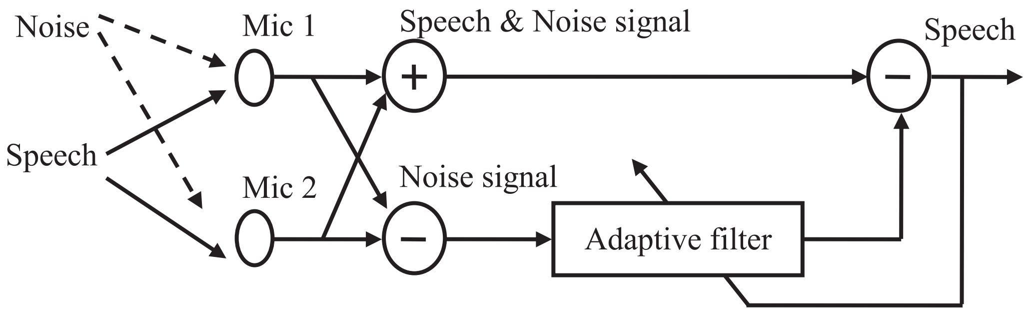


Figure 2.4. Two-channel Griffiths-Jim Beamformer

The Griffiths-Jim beamformer operates as an ANC with a pre-processor that performs the addition and subtraction of the microphone signals. This algorithm works well as long as the target signal arriving at the two microphones is aligned. The speech signals received at the microphones needs to be in phase and the noise signals to be out of phase. When this occurs, the addition process will produce twice the speech and noise signal, and in the subtraction process the speech will cancel out to produce noise signal alone. The resultant signals from this addition and subtraction process are used as the primary and reference signals for the ANC.

One of the problems is that the noise signals in the two input channels are not coherent enough for them to cancel out in the adaptive processes. The Griffiths-Jim beamformer structure is designed to solve this problem. It is sensitive to target signal leakage and cancellation, in the presence of steering vector errors and reverberations. These steering vector errors are caused by errors in the microphones' positions, the microphones' 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, Zeskind, & Owen, 1987; Er & Ng, 1994; Hoshuyama, Sugiyama, & Hirano, 1999; Van Compernolle, 1992; Yongjian, Genmiao, & Shouhong, 2001).

An improved version of the Griffiths-Jim beamformer is proposed by Hoshuyama et al. (1999). This robust adaptive beamformer uses an adaptive blocking matrix consisting of coefficient constrained adaptive filters in the GSC structure, and a multiple input canceller using norm-constrained adaptive filters. This method has been shown to cancel interference by over 30 dB. However, its performance deteriorates in the presence of coloured low-pass interference signals.

Van Compernolle proposed a technique called switching adaptive filter, which attempts to eliminate some problems of the Griffiths-Jim beamformer (Van Compernolle, 1990; 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). This method is called the Switched Griffiths-Jim Beamformer (SGJBF) method. A great deal of success has been obtained by this method compared to others. Figure 2.5 illustrates a conceptual diagram of this technique for two microphones. This design can be easily expanded to use more microphones as required. This beamformer algorithm makes use of two adaptive filters. The first filter works as a beam-steering filter, and the second one works as an ANC.

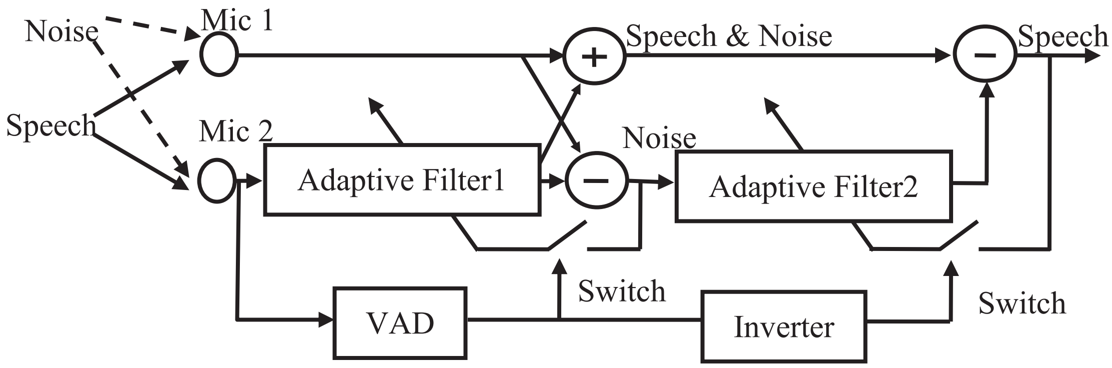


Figure 2.5. Switched Griffiths-Jim Beamformer

For real-time implementation of this algorithm, there is a trade-off between the number of channels and the number of filter taps per channel; due to limited computational power. If the microphones happened to be in the wrong position with respect to each other, then a non-causal solution can be caused as the result. To prevent this problem from happening the delays are introduced in one of the input channels to provide physical reliability. This beamformer algorithm has been proven to improve the overall performance of the system by 5 dB for a competing speaker, by 8 dB for a wideband semi-stationary noise, and by 20 dB for narrowband interference (Van Compernolle, Van Gerven, Broos, & Weynants, 1991). The next section will briefly discuss some of the adaptive filter algorithms that can be used for this adaptive beamformer.

2.2 Adaptive Filters

Adaptive filters are used when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters (Diniz, 2008; Haykin, 2002). 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. Adaptive filters can be classified as either supervised or un-supervised methods (Haykin, 2000). Supervised adaptive filters generally require a noise signal as a reference signal, to approximate the error signal, and unsupervised adaptive filters do not require any information about the source. As a result, the unsupervised variety is often called "Blind Source Separation" and more often than not is based on Independent Component Analysis (Makino, Lee, &

Sawada, 2010). Whilst the supervised adaptive filters require information on the additive noise and usually employ some form of voice activity detectors. The latter approach is considered in this thesis, due to the fact that it is less computationally intensive.

Adaptive filters have been investigated for several decades, and they have been used in practical applications where the signal varies over time. Characteristics of the adaptive filters make it attractive for signal processing and control applications. Some applications that use adaptive filters include adaptive channel equalisation (Hadei & Azmi, 2010; Malik & Sappal, 2011), interference canceller (Gan, Zahedi, Alauddin, & Ali, 2011; Lee & Lee, 2005), 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 (Diniz, 2013; 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 is discussed here, since the others are not relevant to this application. Figure 2.6 shows a basic structure of an adaptive noise cancelling configuration. For noise reduction applications, the primary input consists of "speech and noise", and the reference input consists of "noise alone". The noise present in both of these inputs must be correlated and the noise present in the primary input must be uncorrelated with the speech signal. This structure uses the reference input to reduce the effect of noise in the primary input.

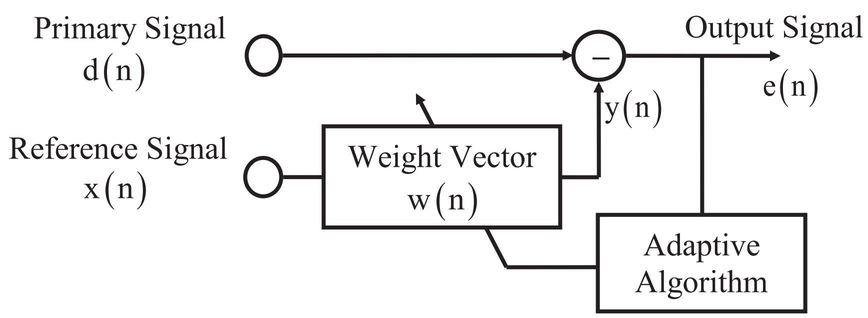


Figure 2.6. Adaptive Filter in Adaptive Noise Cancelling Configuration

As shown in Figure 2.6, this structure has two inputs (primary signal $d(n)$ and reference signal $x(n)$) and an output signal $e(n)$.

The output error signal $e(n)$ is given by:

$$e(n) = d(n) - y(n) \tag{2-1}$$

The output error signal measures the difference between the primary input signal and the output of the adaptive algorithm. The filter output $y(n)$ is given by:

$$y(n) = \vec{x}^{T}(n)\vec{w}(n) \tag{2-2}$$

where, $\vec{w}(n)$ is the time varying vector of filter coefficients (tap weights), $\vec{x}(n)$ is the column vector of the input reference signal, and superscript "T" denotes a vector transpose and → denotes a vector.

The input regressor vector is given as $\vec{x}(n)=\left[x(n)x(n\text{-}1)...x(n\text{-}N\text{+}1)\right]^{T}$ and the tap weight vector is given as $\vec{w}(n)=\left[w(0)w(1)...w(N\text{-}1)\right]^{T}$. Initially, they are both assigned to zero ($\vec{w}(0)=\vec{x}(0)=\left[0...0\right]^{T}$). However, if rough values of the tap weights are known, then these values can be used instead.

An adaptive algorithm is required in order to adjust the values of tap weights $\vec{w}(n)$ 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. This adaptive algorithm can be either linear or non-linear filters. This is explained in details in the following sections.

2.2.1 Adaptive Algorithms

Many adaptive algorithms have been proposed in the past for adaptive filtering. They vary a lot from time domain analysis to frequency domain analysis. Time domain analysis is relatively simple to compute when compared with the frequency domain analysis. A few important algorithms on the time domain adaptive filtering area will be discussed in this section.

Widrow and Hoff were among the first to investigate the adaptive filters in the late 1950s (Widrow, 2005). Their work resulted in the famous Least Mean Square (LMS) algorithm (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 drives its tap weight parameters to values corresponding to minimising the Mean Squared Error (MSE) between the filter output and primary input signal. The complexity of this algorithm is very low and its results are satisfying in many cases.

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

The most common solution to this problem is the Normalised LMS (NLMS) algorithm (Yassa, 1987), which utilises a variable convergence factor that minimises the instantaneous error. This NLMS algorithm usually converges much more quickly than the conventional LMS algorithm, for both uncorrelated and correlated input signal (Slock, 1993). Due to the simplicity of this algorithm, it has been one of the most popular algorithms to be used for updating the adaptive filter.

Other adaptive algorithms like Recursive Least Square (RLS) algorithm and Affine Projection Algorithm (APA) (Ozeki & Umeda, 1984) are available with much faster convergence rates than the NLMS, but they are more computationally complex, and require more processing time. These algorithms use the steepest-descent based technique to recursively compute and update the values of the weight vectors.

Table 2.1 shows the recursive equations used to update the tap weight vector $\vec{w}(n)$ of these adaptive filters (Zaknich, 2005), computed at iteration (time-step) $n+1$.

Table 2.1. Comparison of Adaptive Algorithms

Adaptive Filter	Adaptive Algorithm	Computational Complexity
LMS	$\vec{w}(n+1)=\vec{w}(n)+(\mu e(n)\vec{x}(n))$ where μ denotes step-size	$O(2N)$
NLMS	$\vec{w}(n+1)=\vec{w}(n)+\left(\dfrac{\mu_n}{\gamma+\vec{x}^T(n)\vec{x}(n)}\right)e(n)\vec{x}(n)$ where γ is a small positive constant used to prevent division by zero situations	$O(3N)$
APA	$\vec{x}(n)=\left[x(n)x(n-1)\ldots x(n-N+1)\right]^T$ $\mathbf{X}_P(n)=\left[\vec{x}(n)\vec{x}(n-1)\ldots\vec{x}(n-M+1)\right]$ $\vec{w}(n+1)=\vec{w}(n)+\mu\mathbf{X}_P(n)\left[\mathbf{X}_P(n)^T\mathbf{X}_P(n)+\delta\mathbf{I}\right]^{-1}\vec{e}(n)$ where N is the number of tap weights, M is the projection order, δ denotes regularisation parameter	$O(MN)$
RLS	$\vec{w}(n+1)=\vec{w}(n)+\vec{k}(n)e(n)$ $\mathbf{P}(n+1)=\lambda^{-1}\mathbf{P}(n)-\lambda^{-1}\vec{k}(n)\vec{x}(n)^T\mathbf{P}(n)$ Gain vector $\vec{k}(n)=\dfrac{\lambda^{-1}\mathbf{P}(n)\vec{x}(n)}{\left[1+\lambda^{-1}\vec{x}(n)^T\vec{x}(n)\mathbf{P}(n)\right]}$ where λ is forgetting factor, which is set to less than but close to 1. Initially $\mathbf{P}(0)=\delta^{-1}\mathbf{I}$, where δ denotes small positive value.	$O(N^2)$

(In Table 2.1, $\vec{w}(n)$ denotes the weight vector, $\vec{x}(n)$ denotes the input regressor vector, $\vec{e}(n)$ denotes the error signal (defined in equation (2-1)), T denotes transpose, **I** denotes an identity matrix, → denotes a vector, and **Bold** denotes a matrix.)

The APA's computational complexity and convergence rate is considered to be in-between the NLMS and the RLS algorithms. Due to the high computational complexity of the RLS it is not widely used for real-time applications, though low-order APA is more appropriate for real-time implementation. Comparisons of NLMS and APA have been carried out by many researchers and APA is found to perform better under certain conditions (Yoganathan & Moir, 2008). A faster version of the APA has been also proposed to reduce the computational complexity of this method (Gay & Tavathia, 1995). Several methods like the fast RLS and block RLS methods (Montazeri & Duhamel, 1995) have also been proposed to improve the RLS.

In the least square algorithms, step-size (μ) controls the amount of gradient information used to update each coefficient. Thus, the value of μ directly affects how quickly the adaptive filter will converge toward the unknown system. This value of μ is chosen between 0 and 1. If μ is small, the value of the filter coefficients changes only by a small amount at each update, 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 μ is too large, the coefficients may change too quickly and the filter will diverge, hence the system will become unstable.

Choosing the correct μ value has been a problem in many of these adaptive filter algorithms, since choosing a large μ will result in faster convergence, but might cause instability in some situations, or choosing a small μ will result in slower convergence. This problem has been identified by many researchers and different solutions have been proposed in the past two decades. An attractive solution is to use a Variable Step-Size based algorithm (Kwong & Johnston, 1992). Many methods have been proposed to calculate an appropriate step-size (Aboulnasr & Mayyas, 1997; Evans, Ping, & Liu, 1993).

Noise cancelling methods discussed above are mostly focusing on the assumption that the corrupted noise is added to the speech signal. Generally, linear filters work best to remove this type of noise signal. However, the problem only arises when the noise signals are not added. So, by design, these filters are unable to completely approximate the non-linear part of the given system. As a result, there has been a lot of interest in the non-linear signal processing area. Volterra series expansion was used in earlier days for this problem (Mathews, 1991). However, a shortcoming of this method is that the highest degree of expansion has to be defined before hand, which is not always possible for real-time processing. In this thesis, using NN based adaptive filters for noise reduction is explored. The next section will give a review of the NN techniques used for noise reduction.

2.3 Neural Networks

The inspiration for neural network (NN) study came from the biological field, when looking at how the brain works. Many researchers have tried to understand the power of neurons and how they are used by humans for thinking. Researchers have been trying to recreate the capability of the neural nets for many years and study in this field is called Artificial Neural Network (ANN). Early work on artificial neurons can be referred to McCulloch and Pitts research (McCulloch & Pitts, 1943). They introduced the first computational model of the neuron as a threshold logic unit where the output of the system is 0 or 1 and a threshold level is used to activate the output (shown in Figure 2.7).

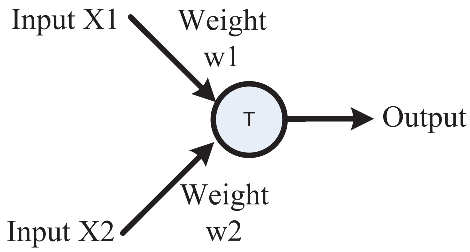


Figure 2.7. McCulloch-Pitts Model

Initially, perceptron was studied and was discarded; this created a gap in the field and not much research has been done on this area. After the introduction of single-layer perceptron, there was a period of nearly two decades when not much improved in the NN area. This continued until the introduction of the multi-layer perceptron (MLP) idea, which sparked a strong interest in this area (Rumelhart & McClelland, 1986).

The first neuron structure, called an adaptive linear element (ADALINE) was known to be built by Widrow and Hoff in 1960 (Liew Ban, Hussain, & Samad, 2000; Stella, Begusic, & Russo, 2006). This research is based on the McCulloch and Pitt's neuron model. It is a single neuron linear adaptive filtering structure based on the LMS algorithm.

In recent times, ANN has been used for many different applications in a variety of disciplines (Badri, 2010; Gaonkar & Savan, 2012; Hoyt & Wechsler, 1990). Examples of applications in the signal processing area include: system identification, pattern recognition, image processing, face recognition (Nagi, Ahmed, & Nagi, 2008; Revathy & Guhan, 2012), speech recognition, pattern and statistical classifiers in biomedical engineering (Murugesan & Sukanesh, 2009) and adaptive beamformer for antenna array applications (Du, Lai, Cheng, & Swamy, 2002), to name a few. Previous work in the noise reduction area using the NN is detailed below to give the reader a basic understanding of this subject.

2.3.1 Speech Enhancement Using Neural Networks

Early work on NN based noise reduction method was proposed by Tamura and Waibel (Tamura, 1989; Tamura & Waibel, 1988). They introduced a four-layered feed-forward NN to map noisy speech to a noise-free speech signal. This was tested on artificially generated noisy Japanese speech data using stationary and non-stationary noise. This system was also compared with a conventional spectral subtraction method, and was found to be comparable or better. Although the output signal from the network was clear, they conclude that processing does not improve the intelligibility of speech signals. The main shortcoming of this method is that it requires a clean speech signal for training the network and this is not possible to obtain in all cases. In addition to this, the NN structure used is not suitable for real-time processing.

Later research done by Cox also showed the possibility of applying the back-propagation network for filtering noise signals (Cox, 1988). His results prove that this network can be trained to work as a digital filter. By doing so, he also supports the research findings of Tamura and Waibel. It was also found that the network was able to work in extremely low signal-to-noise ratios. In addition, the NN can be used for other applications, apart from pattern recognition. Following this studies there were many other researchers who also worked on adaptive filtering using NN, but they have applied it to other areas (Anderson & Montgomery, 1990; Rao & Pisharam, 1990; Weber, Crilly, & Blass, 1991).

Later research done by Kencht introduced the idea of using non-linear noise filtering and the perceptron based beamformer (Knecht, 1994). In that paper, they consider a fixed beamformer for noise reduction, where the weights are estimated in the training period and are used for testing. This paper also studies the relationship between the perceptron and the Volterra filter (Reed & Hawksford, 2000). Their research led to work on a non-linear two-microphone based method for speech enhancement in a hearing-aid applications (Knecht, Schenkel, & Moschytz, 1995). This paper uses the Griffiths-Jim beamformer structure and compares the perceptron filter and the Volterra filter for noise reduction.

There have been several publications on adaptive noise cancellation structures based on NNs (Al-Anbaky, 2004; C. K. Chen & Tzi-Dar, 1996; Kaur & Kaur, 2011; Mistry & Kulkarni, 2013; Stella et al., 2006; Tian & Wang, 2012). They seem to show promising results using this structure. However, they do not cover the limitations caused by this structure. This thesis improves the current research work by combining this idea and the area of beamforming to produce a novel noise reduction algorithm.

2.4 Voice Activity Detectors

Speech detection algorithms or voice activity detectors (VADs) are used to differentiate the speech signal from the non-speech (noise alone) signal in the acquired signal. The general function of the VAD algorithm is to extract some measured quantities from the received signal and compare these values with a threshold value, to make decision about the received signal. For example, if the calculated value is greater than the threshold then the received signal is considered as a speech and noise signal; otherwise, the received signal is considered as a noise alone signal. The output of this calculation is either zero or one, where a one denotes "speech and noise" and a zero denotes "noise alone". This output is generally used with the speech enhancement algorithms to control the noise cancellation.

VAD algorithms are generally used as a pre-processing unit for the speech applications to gain knowledge of the acquired signal. Research also shows that using a VAD with a noise reduction technique significantly improves the performance of the array processing algorithms (Krasny & Oraintara, 2002; Ramirez, Górriz, & Segura, 2007). VAD algorithms have also been used for other speech communication applications such as speech recognition in noisy environments (Babu, Vanathi, Ramachandran, Rajaa, & Vengatesh, 2010; Karray & Martin, 2003; Maganti, Motlicek, & Gatica-Perez, 2007), variable rate speech coding (Gersho & Paksoy, 1992; Hoffman, Li, & Khataniar, 2001), hands-free telephony (Bouquin-Jeannès, Faucon, & Ayad, 1996; Krasny & Oraintara, 2002) and echo cancellation.

VADs were first investigated to be used on Time Assigned Speech Interpolation systems (TASI systems) in the early 1960s for electronic switching of voice circuits (Bullington & Fraser, 1959). VAD achieves silence compression in universal mobile telecommunication systems, and it reduces the average bit rate by using the Discontinuous Transmission (DTX) mode (Freeman, Cosier, Southcott, & Boyd, 1989). The global system for mobile communication (GSM) telephony uses silence detection and comfort noise injection for higher coding efficiency (Srinivasan & Gersho, 1993). In a cellular radio system, it reduces co-channel interference and power consumption in portable equipment (Ezzaidi, Bourmeyster, & Rouat, 1997).

For the past few decades, many researchers have studied different strategies for detecting speech in the presence of background noise, and their influence on speech related applications (Bouquin-Jeannes & Faucon, 1995; Potamitis & Fishler, 2003; Sangwan et al., 2002; Sohn & Sung, 1998; Tanyer & Ozer, 1998; Van Gerven & Xie, 1997; Wei, Du, Yan, & Zeng, 2003; Woo, Yang, Park, & Lee, 2000). The performance of these applications depends heavily on the performance of the VAD algorithm. This is an important problem in the speech processing area, as it is very hard to detect the presence of speech in different types of background noise. Background noise can be anything from stationary to non-stationary noise sources, such as a moving speaker. Therefore, it is very important to choose an appropriate

VAD algorithm for the target noise source that is being differentiated in that particular application. 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).

Voice activity detection algorithms mainly fall into two 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 VAD algorithms are Itakura LPC distance measure (Rabiner & Sambur, 1977), energy distribution, timing, pitch (Arcienega & Drygajlo, 2002), zero crossing rates (Junqua, Mak, & Reaves, 1994), cepstral features (Haigh & Mason, 1993), long-term modulation based methods (Maganti et al., 2007), automatic variance control (Moir, 2001a), statistical model based methods (Jongseo, Nam Soo, & Wonyong, 1999; Kim & Chang, 2012), adaptive noise modelling of voice signals, wavelet based methods (Jeub, Kolossa, Astrudillo, & Orglmeister, 2009), and periodicity measure methods (Tucker, 1992). These algorithms have some trade-offs like computational cost, sensitivity, and accuracy.

The energy-based (Rabiner & Sambur, 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 presence of a speech signal. This algorithm is simple to implement and does not require a lot of assumptions about the characteristics of the noise. The only assumption made here is that the speech is sufficiently louder than the noise. This assumption is accurate at high SNR values. To improve the accuracy of the energy based VAD, they can be used with 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 on 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 (Ganapathiraju, Webster, Trimble, Bush, & Kornman, 1996).

Another well-known VAD is the entropy-based method. 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 the spectral nature of the noise. Shannon's entropy is used to measure the organisation of the signal. It measures the average length of bit code per symbol under optimal coding. 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 performs 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 detailed implementation 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 proven 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.

The drawback of the above statistical based VAD algorithms is that they mainly require the user to have a high SNR. However, this might not be feasible in all cases. An alternative approach is to use the direction of the received signal to differentiate between the presence of wanted speech and unwanted noise signals (Agaiby & 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 originating 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. These types of methods restrict the user to stay within this zone of activity, where the speech is expected to be. However, this algorithm has been proven to work under a variety of noise conditions including competing speakers, because it uses direction as the main factor to distinguish the wanted speech, while most of the other VAD algorithms seem to be unsuccessful when there is a competing speaker in the room. The direction of the signal source can be estimated by using two-microphones and the exact position of the signal source can be identified by using three-microphones.

The direction of the signal is generally approximated by estimating the time delay between the two received signals at the microphones (Strobel & Rabenstein, 1999). Many authors have proposed various approaches for estimating the time delay between two received signals (Quazi, 1981). Some of these include 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 for estimating time delay is the well-known GCC method (Knapp & Carter, 1976). In order to avoid detecting the reverberation signal as speech, the magnitude squared coherence (MSC) calculation is also used with the GCC algorithm to detect the reverberation signals (Bouquin-Jeannes & Faucon, 1995). The MSC is calculated for the received signal and compared with a stored threshold to make a decision on whether echo is present or not. MSC computed for a non-reverberation speech signal is high (close to 1) and MSC computed for a reverberation speech is low (close to 0). Typically, a threshold value of around 0.5 is chosen. Any MSC 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 the estimated time delay and the MSC functions detect the presence of speech. More detailed implementation of this technique can be found in the literature (Agaiby & Moir, 1997; Moir, 2001b). An enhanced version of this algorithm has been implemented for three-microphones, which seems to give better results than the two-microphone algorithm (Chen & Moir, 1999). The main drawback with this algorithm is that it requires lot of computation power. For an application that requires a quick response this might not be a feasible choice.

2.5 Speech Recognition in Noisy Environments

Speech-controlled applications are becoming more commonly used in day-to-day activities. The commercially available systems often require the user to have optimal conditions such as speech characteristics that match the training, a noise-free environment and proper speaker adaptation. As a result, their performance can be as good as 98% to 99% in a relatively quiet environment with a trained system (Jiju, Singh, & Sharma, 2012). However, the main problem with using a speech recognition system is that the performance of their system degrades under noisy environments. The results from research (Chan, Yong, Nordholm, Yiu, & Lam, 2014) show that the commercial recogniser can achieve a 65% recognition accuracy when no enhancement method is used under the signal-to-noise Ratio (SNR) of 5 dB.

An attractive solution is to use a pre-processor with this system to improve its performance. Different noise reduction methods have been used with the speech recognition system to improve its performance. Research by Bitzer, Simmer, and Kammeyer (1999) presents a comparative study of multi-microphone techniques for hands-free speech recognition. Their research shows that using multi-microphone techniques is the way to improve the speech recognition rate. The next chapter will discuss a novel speech enhancement algorithm that can be used with the speech recognition system to improve its performance.

CHAPTER 3

DESIGN AND DEVELOP A SPEECH ENHANCEMENT ALGORITHM

This chapter introduces the novel multi-microphone speech enhancement technique, which is based on a hybrid non-linear neural network (NN) based adaptive beamformer. A fully connected and partially connected input delay neural network (IDNN) is introduced here for noise reduction. This is used in the switched Griffiths-Jim Beamformer (SGJBF) structure, to reduce the background noise. This section also provides the Labview software implementation of the proposed method.

3.1 Problem Formulation

An acoustical model of a dual-microphone speech enhancement system is shown in Figure 3.1.

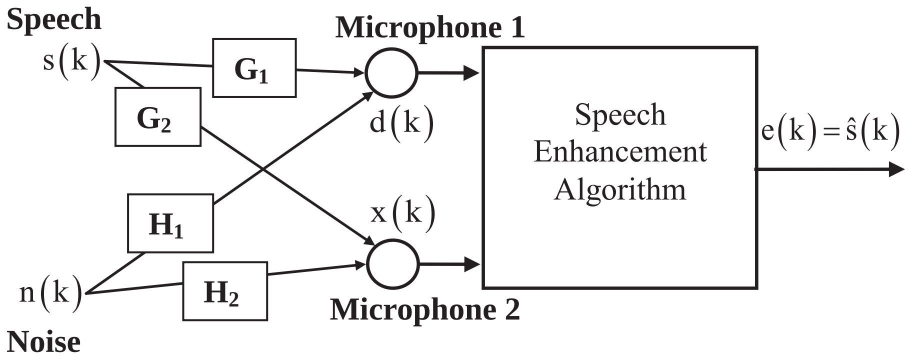


Figure 3.1. Acoustical Model of the Microphone Signals

The speech source and the noise source produce a transfer function from their location to the sensor. These Finite Impulse Response (FIR) transfer functions (i.e. G_1, G_2, H_1, and H_2) are defined in the backwards shift z-transform z^{-1} as:

$$H(z)=\sum_{n=0}^{N_b} b_n z^{-n} \qquad \textbf{(3-1)}$$

where b_n is the filter coefficient (it is given as $b_n = \left[b_0 b_1 b_2 \ldots b_{N_b}\right]$), and N_b is the order of the filter.

The speech and noise are mixed together to produce the input signals for the speech enhancement system. The input signal at the first microphone is given by:

$$d(k)=G_1 s(k)+H_1 n(k) \qquad \textbf{(3-2)}$$

where $s(k)$ is the speech signal, $n(k)$ is the noise signal. Also, G_1 is the FIR transfer function from the speech source to the first microphone, and H_1 is the FIR transfer function from the noise source to the first microphone.

The input signal at the second microphone is given by:

$$x(k) = G_2 s(k) + H_2 n(k) \tag{3-3}$$

where G_2 is the FIR transfer function from the speech source to the second microphone, and H_2 is the FIR transfer function from the noise source to the second microphone.

The aim of the speech enhancement algorithm here is to enhance the speech signal by reducing the unnecessary noise present at the microphones. The next section introduces a novel multi-microphone speech enhancement algorithm based on NN.

3.2 Proposed Speech Beamformer Algorithm

As discussed in the literature review section, the studies done by Stella et al. (2006) and Van Compernolle et al. (1991) are the inspiration for this research work. A combination of these works leads to the hybrid system proposed here. Stella et al., (2006) only use an “ADALINE neural network” structure as an adaptive filter in the traditional ANC structure. Their application of the adaptive filter is to reduce the engine noise in cars. They show that ADALINE performs well as an adaptive filter to reduce the noise in real-life scenarios. Their experiment only concentrates on reducing stationary noise.

Over the last few years, there has been an increased interest in NN based noise reduction algorithms. The general trend in recent research work shows that a variety of NN methods have been applied to the adaptive noise cancelling (ANC) structure for noise reduction. They show quite promising results in certain applications. However, results in these papers are produced from computer simulation. Also, in a few of these studies, it is assumed that there is a noise-alone signal for the ANC structure. However, it is only possible in some practical applications to acquire this noise-alone signal. In many cases, this reference signal gets corrupted by speech, because it is often impossible to avoid the speech entering the second microphone, while keeping the noise signal correlated in both microphones.

Research by Griffiths and Jim improved the current ANC algorithm by introducing a speech alignment section before the noise cancellation section (Griffiths & Jim, 1982). However, this algorithm still has a few shortcomings, like the speech being cancelled in the noise section. Van Compernolle's work (1990) improved the Griffiths-Jim structure by controlling the update of the adaptive filtering structures. It is important to control the update of these filters. Supposing that the first filter is updated all the time, then there is a possibility of the beam being directed towards the noise source, thinking that is the desired signal. The second adaptive filter's weights are only updated during the noise-alone signal; which is done to avoid the speech signal being corrupted by the noise reduction. The second adaptive filter will reduce the noise signal; therefore, it is not necessary to align the noise signal in the beam-steering section. Also, using a switch based on the VAD eliminates the requirement for speech-free noise references.

Figure 3.2 shows the outline of the proposed switching adaptive filter structure for speech enhancement. This design will be referred to as the Switched Griffith-Jim Beamformer (SGJBF) structure.

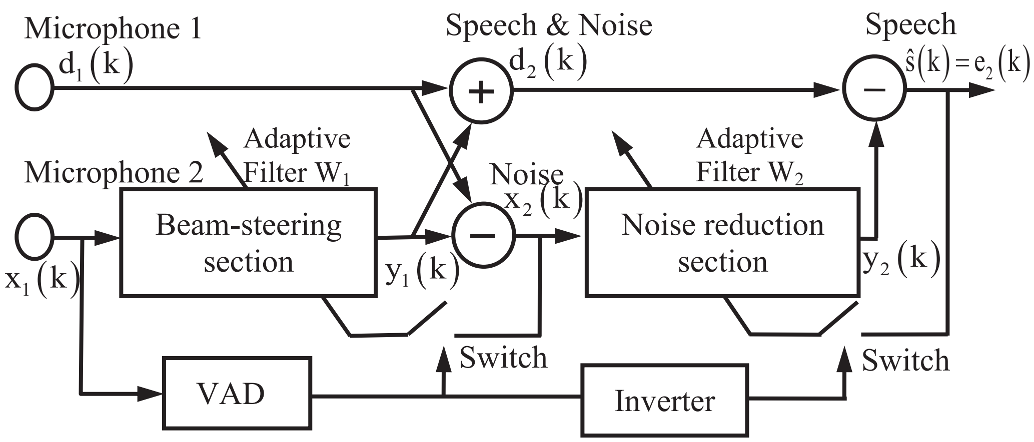


Figure 3.2. Proposed Non-linear Neural SGJBF Structure

The desired signal $d_1(k)$ and the reference signal $x_1(k)$ of the beam-steering filter are given as follows:

$$d_1(k) = G_1 s(k) + H_1 n(k) \qquad \textbf{(3-4)}$$

$$x_1(k) = G_2 s(k) + H_2 n(k) \qquad \textbf{(3-5)}$$

(Please refer to Figure 3.1 for more details about the variables and the transfer functions used here.)

Adaptive filters are used to determine the difference in the acoustics transfer function of the room. The output signal from the beam-steering filter is given by:

$$y_1(k) = \left(G_2 s(k) + H_2 n(k)\right) W_1 \qquad \textbf{(3-6)}$$

This first filters output signal and the desired input signal $d_1(k)$ are added together to produce the second filters desired signal $d_2(k)$ and subtracted together to produce the second filters reference signal $x_2(k)$. The aim of this process is to obtain the "speech and noise signal" and the "noise-alone signal" for the noise cancelling filter.

The output from the addition function is given as:

$$d_2(k) = d_1(k) + y_1(k) \qquad \textbf{(3-7)}$$

By substituting $d_1(k)$ and $y_1(k)$ in the equation (3-7), we obtain the following results:

$$d_2(k) = \left(G_1 s(k) + H_1 n(k)\right) + \left(G_2 s(k) + H_2 n(k)\right) W_1 \qquad \textbf{(3-8)}$$

The aim of the beam-steering filter is to approximate the transfer function $W_1 = {G_1}/{G_2}$ in the above equation. As a result, $d_2(k)$ becomes:

$$d_2(k) = \left(G_1 + \frac{G_1}{G_2} G_2\right) s(k) + \left(H_1 + \frac{G_1}{G_2} H_2\right) n(k) \qquad \textbf{(3-9)}$$

The two speech signals are added together to produce a single signal given that the exact transfer function is approximated by the first filter.

$$d_2(k) = 2G_1 s(k) + \left(H_1 + \frac{G_1}{G_2} H_2\right) n(k) \qquad \textbf{(3-10)}$$

More details on the adaptive algorithm used to approximate the transfer function of the speech signal are given in section 3.2.1. In order to simplify the equation (3-10), let $K=\frac{G_1}{G_2}H_2$ and by substituting K in $d_2(k)$, we obtain the following equation:

$$d_2(k)=2G_1s(k)+(H+K)n(k) \tag{3-11}$$

The output error signal $e_1(k)$ from the subtraction function is given as:

$$e_1(k)=x_2(k)=d_1(k)-y_1(k) \tag{3-12}$$

By substituting $d_1(k)$ and $y_1(k)$ in the equation (3-12), we obtain the following results:

$$x_2(k)=\left(G_1s(k)+H_1n(k)\right)-\left(G_2s(k)+H_2n(k)\right)W_1 \tag{3-13}$$

$$x_2(k)=\left(G_1s(k)-\frac{G_1}{G_2}G_2s(k)\right)+\left(H_1n(k)-\frac{G_1}{G_2}H_2n(k)\right) \tag{3-14}$$

The first term in the equation (3-14) vanishes giving:

$$x_2(k)=\left(H_1-\frac{G_1}{G_2}H_2\right)n(k) \tag{3-15}$$

Substituting K in the equation (3-15) we obtain:

$$x_2(k)=(H_1-K)n(k) \tag{3-16}$$

The second adaptive noise cancelling filter's output is given by:

$$y_2(k)=\left((H_1-K)n(k)\right)W_2 \tag{3-17}$$

The output error signal $e_2(k)$ from the second filter is given as:

$$e_2(k)=d_2(k)-y_2(k)=\hat{s}(k) \tag{3-18}$$

By substituting $d_2(k)$ and $y_2(k)$ in the equation (3-18), we obtain the following results:

$$e_2(k) = \left(2G_1 s(k) + (H_1 + K)n(k)\right) - \left((H_1 - K)n(k)\right)W_2 \qquad \textbf{(3-19)}$$

The aim of the noise cancelling filter is approximating the transfer function W_2, which should approximate to: $W_2 = \frac{(H_1 + K)}{(H_1 - K)}$

As a result, $e_2(k)$ becomes:

$$e_2(k) = \left(2G_1 s(k) + (H_1 + K)n(k)\right) - \left((H_1 - K)\frac{(H_1 + K)}{(H_1 - K)}n(k)\right) \qquad \textbf{(3-20)}$$

The noise signals are minimised to produce the speech signal when the exact transfer function is approximated. More details on the adaptive algorithm used to approximate the transfer function of the noise signal are given in section 3.2.1.

As a result, the noise term in the equation (3-18) vanishes giving:

$$e_2(k) = 2G_1 s(k) \qquad \textbf{(3-21)}$$

The remaining sections will explain in detail the adaptive algorithms used for approximate these transfer functions, and also the VAD algorithm used to control the update of this algorithm for the proposed neural SGJBF algorithm.

3.2.1 Beam-steering Filter

As explained earlier, the purpose of the beam-steering filter is to achieve an optimal phase alignment between the input channels. Therefore, this filter is only updated during the presence of the speech signal. Hence, this filter creates an invisible look direction and focuses on the desired speaker. The aligned speech signals are then processed by adding together to produce the primary signal for the ANC and subtracted together to produce the reference signal for the ANC. However, the reference signal for the ANC is hardly ever speech-free, as a result of multi-paths and strong signal conditions.

The addition and subtraction section of this algorithm is quite essential part for noise reduction. At the output of the addition section, twice the speech and noise signal is creates by adding the speech signals together. At the output of the subtraction section, a noise-alone signal is created by cancelling out the speech signal. If the speech signals are not aligned properly, they will not be subtracted completely to give a speech-free noise reference signal, and they will not be added together to create the primary speech signal.

An adaptive filter algorithm is used here to approximate this transfer function. The two microphones are placed about 30 cm apart, and the location of the user is directly in front of the system. A simple filter is sufficient to perform the necessary tasks in the beam-steering section as it is not necessary to follow all the small head movements. The well-known NLMS algorithm aligns the speech signals at the microphones, and Figure 3.3 shows the conceptual structure of this algorithm. This is an improved version of the traditional LMS algorithm. The main advantages of using this algorithm are that it is simple to implement, has more stability and a faster convergence rate than the LMS algorithm. Also, it is one of the most popular adaptive filter algorithms to be used in signal processing applications. However, it can be replaced with any algorithm that fits the target application.

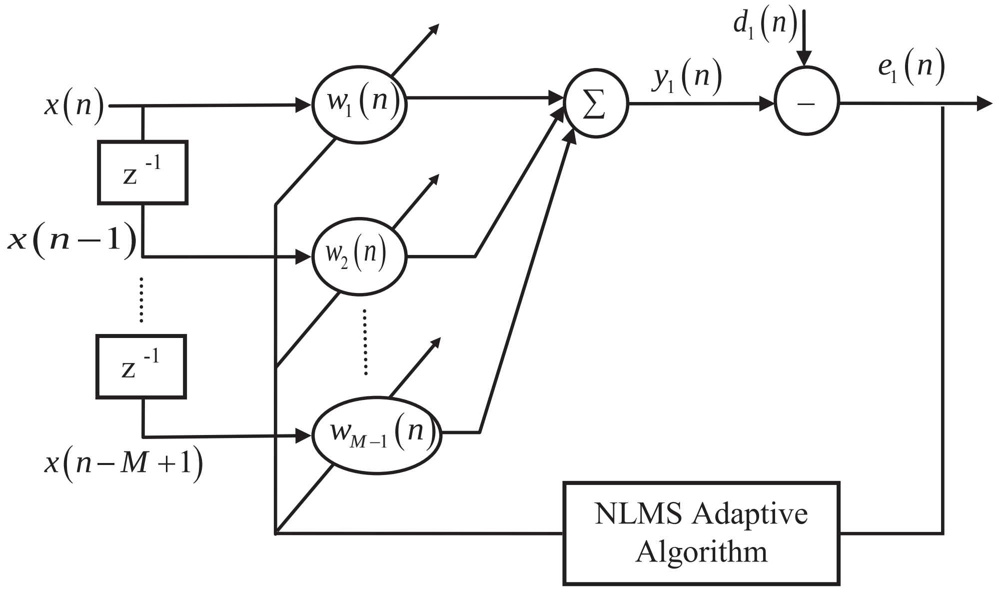


Figure 3.3 Normalised Least Mean Square Algorithm Structure

The output of the NLMS filter $y(n)$ at the nth iteration is given by:

$$y(n) = \sum_{i}^{M} x_i(n) w_i(n) = \vec{x}^T(n)\vec{w}(n) \quad \textbf{(3-22)}$$

where → denotes a vector, and M is the order of the filter. The reference input signal $\vec{x}(n)$ at the nth iteration is given by:

$$\vec{x}(n)=\left[x(n),x(n-1),\cdots,x(n-M+2),x(n-M+1)\right]^{T} \quad \textbf{(3-23)}$$

The weight vector $\vec{w}(n)$ at the n^{th} iteration is given by:

$$\vec{w}(n)=\left[w_0(n)w_1(n),\cdots,w_{M-2}(n)w_{M-1}(n)\right]^{T} \quad \textbf{(3-24)}$$

The following NLMS equation calculates updating weight coefficient $\vec{w}(n)$ at n+1 iteration (Tsuda & Shimamura, 2002):

$$\vec{w}(n+1)=\vec{w}(n)+\left(\frac{\mu_n}{\gamma+\vec{x}^{T}(n)\vec{x}(n)}\right)e(n)\vec{x}(n) \quad \textbf{(3-25)}$$

where μ_n is the step-size, and γ is a small positive constant introduced in the denominator to prevent division by zero situations.

In order to reduce the impact of the input signal's amplitude on the gradient noise, the step-size μ_n is divided by the variance of the input signal $\left[\vec{x}^{T}(n)\vec{x}(n)\right]$. This process is called the normalisation of the step-size. This value is chosen to be in the range of 0 to 1. This normalisation process makes the convergence rate insensitive to the power level of the input signal.

It is necessary to update this weight vector only during the presence of the speech signal to prevent the beam from steering to the noise source. The output switch from the VAD function is used to control this update. Figure 3.4 shows the software implementation of this algorithm in the Labview software.

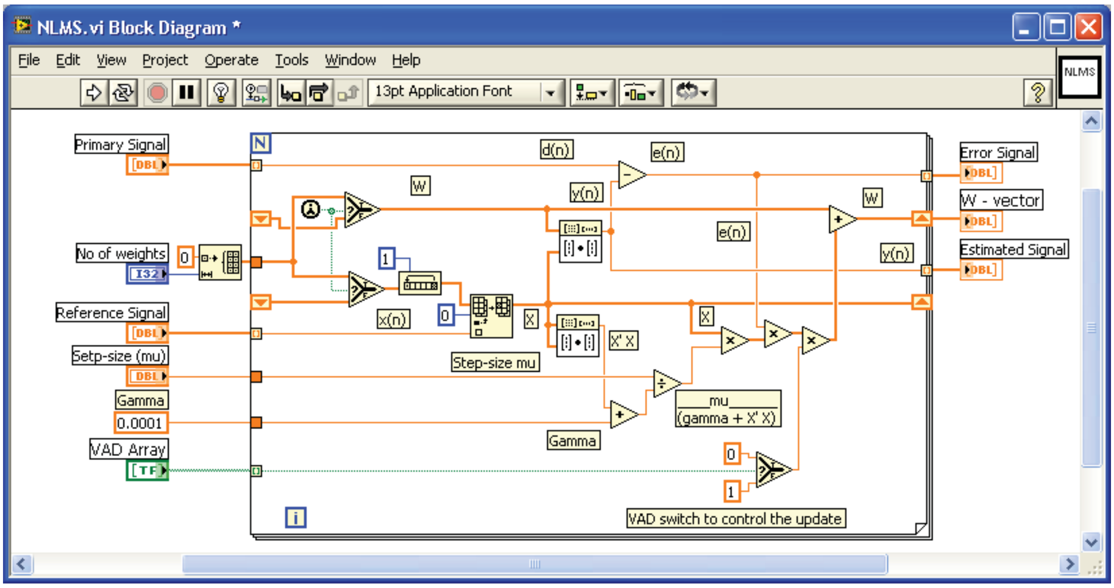


Figure 3.4. Implementation of the NLMS Adaptive Filter in the Labview Software

This algorithm can be replaced by any other algorithms like the affine projection algorithm (APA) and recursive least mean square (RLS) algorithm that can outperform this algorithm, but they require more processing power. (In Appendix A, the paper published by the author on the APA based beamformer is given for further reading). There is a trade-off between the performance and the complexity of the algorithms. As a result, they will not be suitable for real-time processing with the combination of the NN based noise reduction system as well as the speech-controlled application.

3.2.1 A New Neural Network Based Noise Reduction Filter

An adaptive noise canceller (ANC) structure is used here to reduce the unwanted background noise. However, this filter is only updated during the noise-alone period; which prevents the target signal leakage problem raised in the traditional ANC structures. Hence, by stopping the filter from updating its weights during the presence of a speech signal, it eliminates the possibility of speech getting corrupted. The output signal from the ANC is the enhanced speech signal. This filter has to respond quickly to all the sudden changes in the noise signal.

In general, traditional SGJBF uses a linear filter like the NLMS to reduce background noise. However, real-time processing requires a large amount of weights to get any amount of noise reduction. This is because the transfer function of the real-world noise is unknown, and when a linear filter is used here, it does not perform as well as it did in the simulation. A non-linear adaptive filter is required, in order to reduce this non-linear signal effectively. Therefore, a non-linear NN based adaptive filter is proposed here.

There is a wide range of non-linear filters available that can be used for this noise reduction. However, this thesis focuses on an application that requires real-time processing. As a result, an adaptive filter that does not require a lot of processing will be more appropriate for this thesis. A NN based non-linear adaptive filter is chosen here (as discussed in section 2.2).

The following options are used for the NN structure.

Feed-forward (FF) or Feed-backward (FB)

The flow of data in the network can be either feed-forward (static) or feed-backward (dynamic). In the feed-forward (FF) structure, the data passes from the input layer to the output layer. In the feed-backward (FB) structure, the output of any layer may be fed back to itself or to any other layers (Landau & Taylor, 1998). It is also known as the recurrent network. Figure 3.5 shows an example conceptual diagram of both of these structures. As shown in the diagram, the FF network only has the data flow in one direction and the FB network might have any combination of data flow. The choice of these structures depends on the requirements of its application. For this particular application, a FF NN is more appropriate as it only requires the reference input data in the network structure.

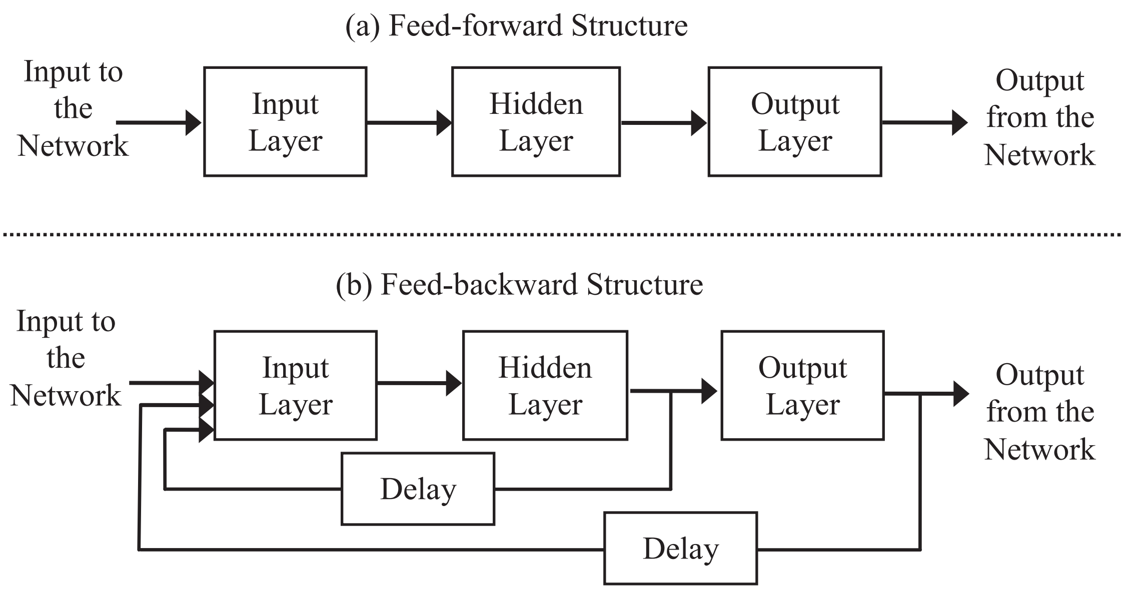


Figure 3.5. The Direction of the Data Flow in the NN Structure (a) Showing the FF Network and (b) Showing the FB Network

Single or Multi-Layer Perceptron (MLP)

The number of layers can be either a single layer (i.e. with no hidden layers) or a multi-layer perceptron (MLP). It is found by many researchers that any network structure with a single hidden layer can be used as a universal approximate. Since the structure is aimed towards real-time processing, only the basic three-layer perceptron structure is investigated here. It is also evident from many research works, that a three-layer network is good enough to approximate any unknown systems, given that a sufficient amount of hidden units is available (Hornik, Stinchcombe, & White, 1989).

Time Delay Neural Network

In addition to the above configurations, the input data to the NN system will be given as a tapped delay line structure. This is called the Input Delay Neural Network (IDNN) structure, as the input to the network is delayed. This is a special feature of the Time Delay Neural Network (TDNN) structure. The TDNN structure introduces the delay in the NN architecture's connectivity; on the other hand the IDNN structure only introduces the delay in the input signal and not in the architecture itself. In the adaptive filter area, this method is considered as the tapped delay line. This structure uses time series data for the input value. It is important to note that this structure is a FF network; however, the feedback behaviour at each synapse of the network acting as a FIR filter (Sinha, Gupta, & Rao, 2000). The IDNN adaptive filter proposed here is shown in Figure 3.6. This consists of a FF three-layer perceptron with one hidden layer.

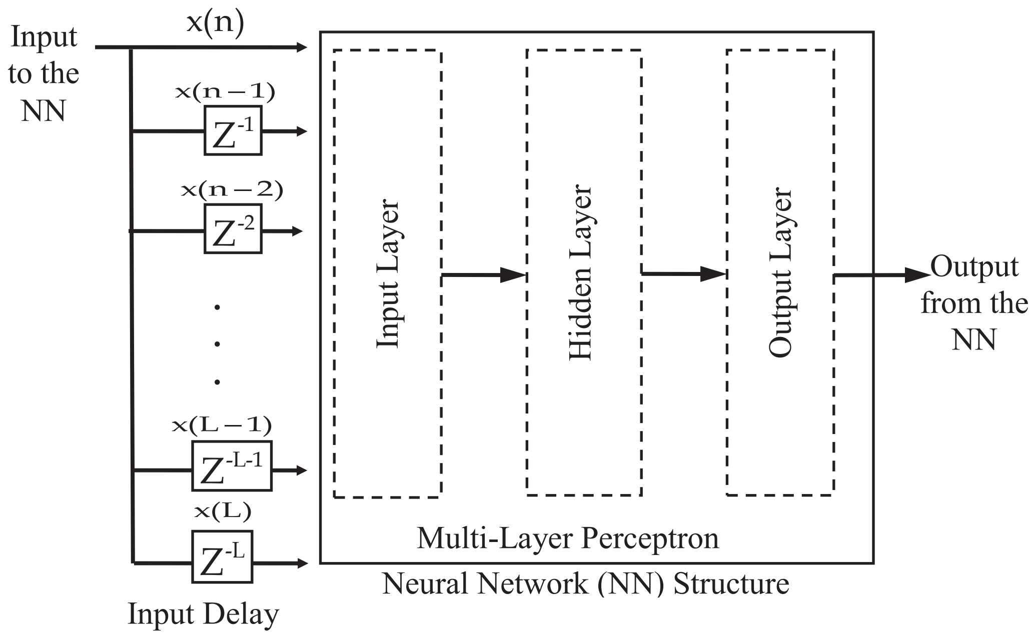


Figure 3.6. Overall Structure of the MLP with IDNN

Fully or Partially Connected Network

The connections between the input and output layers can either be fully connected (FC) or partially connected (non-fully connected or NFC). Figure 3.7 shows the connectivity for the fully and partially connected network structure. The FC network means that all the input nodes are connected to each of the neurons and the NFC network means that only some of the input nodes are connected to the neurons. All the hidden layers are connected to the output layer for both structures.

In the NN structure shown in Figure 3.7, there are eight input nodes, five hidden nodes and an output node (8-5-1). In the FC network, all the input nodes are connected to the hidden layers, and in the NFC network only four inputs are connected to the hidden layer. This greatly reduces the amount of weight vectors in the first layer. The choice of this connectivity, and the amount of nodes required in each layer, is decided experimentally in the next chapter.

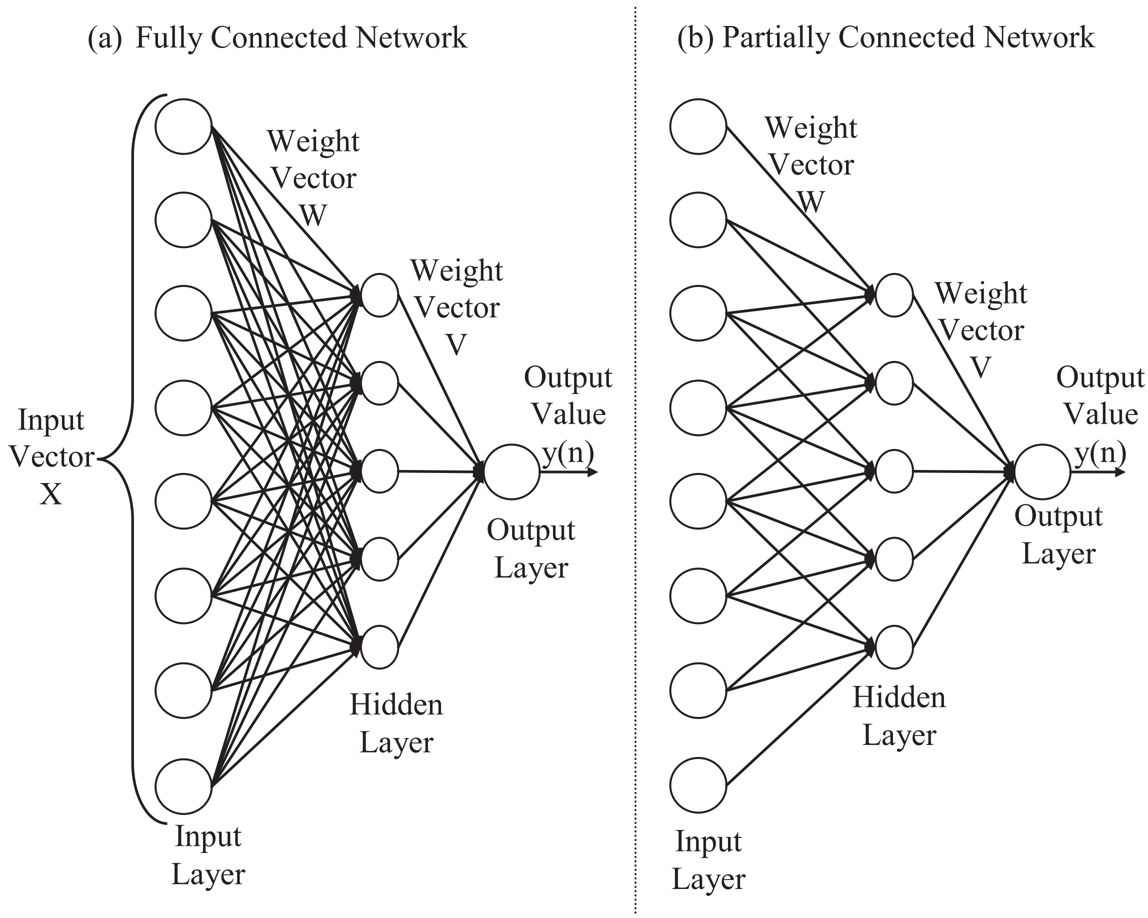


Figure 3.7. Fully and Partially Connected Network Structures

Computational Complexity

In a fully connected structure, the computational complexity of a single neuron structure of the NN is almost the same as the traditional LMS algorithm. As the number of neurons in the hidden layer increases, the complexity of the algorithm is multiplied by the number of neurons used. When using the partially connected structure, the computational complexity can be reduced slightly so that not all the inputs are required.

Activation Function

Sigmoid transfer function is most commonly used activation function by many researchers. Examples of sigmoid functions include arctangent and hyperbolic tangent. However, later researchers have found that MLP has the universal approximation capability, and intermediate layer activation functions are not as crucial as was thought earlier (Stinchcombe & White, 1989).

The activation function chosen here is the hyperbolic tangent non-linear sigmoid function; it is used for all neurons on the network. This function and its derivative are given as follows:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \textbf{(3-26)}$$

$$f'(x) = 1 - f(x)^2 \quad \textbf{(3-27)}$$

An example of a hyperbolic tangent graph and its derivative is shown in Figure 3.8.

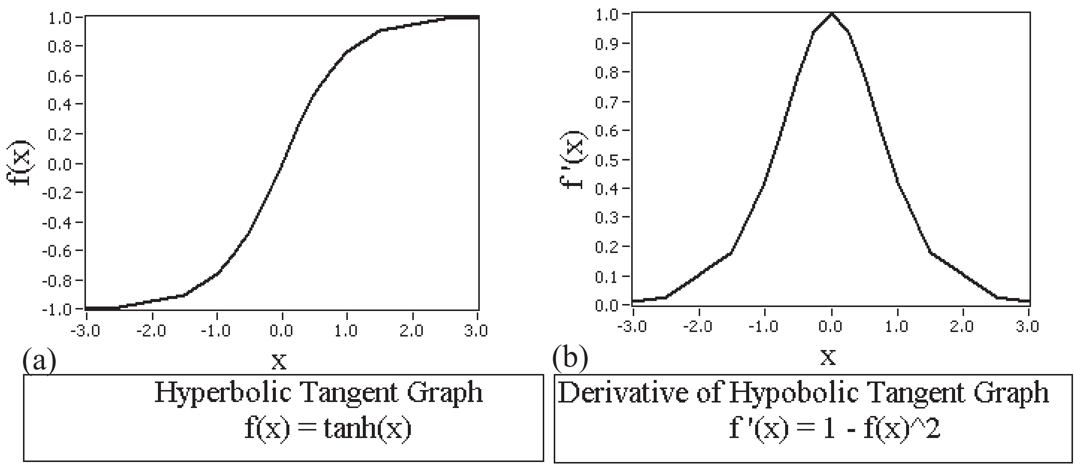


Figure 3.8. Activation Function (a) Hyperbolic Tangent Graph, (b) Its Derivative

Figure 3.9 shows the proposed NN structure in detail. In the proposed system, a partially connected three-layer IDNN is used as an adaptive noise canceller for the SGJBF structure. It is partially connected to achieve real-time processing. This is the particularly important original contribution to this area. The structure has not been used in this way for speech enhancement before (to the author's knowledge).

This structure will be tested with different background noise sources to analyse the performance of this system.

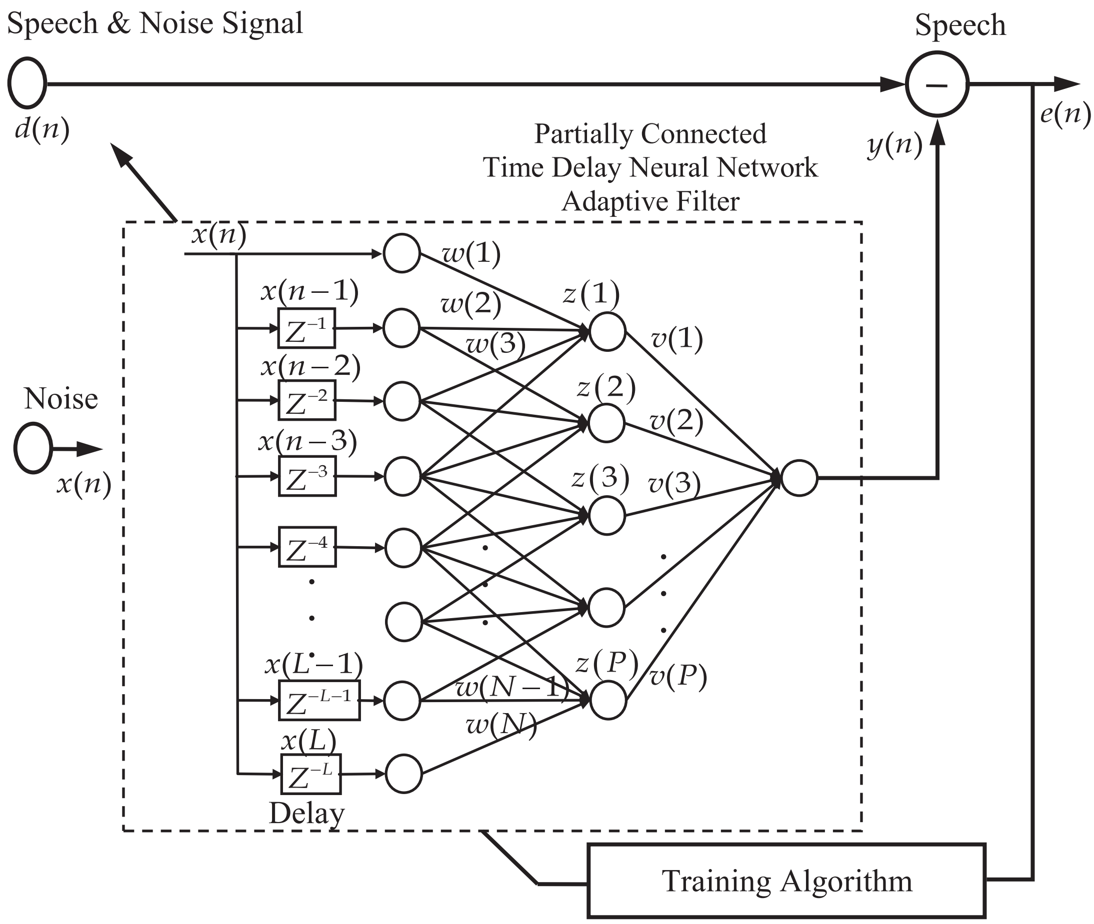


Figure 3.9. Non-linear Partially Connected Three-layer FF NN ANC Structure

In the noise reduction filter shown in Figure 3.9, the output error signal $e(n)$ is the "enhanced speech" signal, desired signal $d(n)$ contains the "speech and noise" signal, and "estimated" signal $y(n)$ is the output signal from the NN adaptive filter. The output of this filter is given by:

$$y(n) = f\left(\left(\sum_{j=1}^{P} v_j(n) z_j(n)\right) + a\right) \quad \textbf{(3-28)}$$

where $\vec{v}(n)$ is the second weight vector at the hidden layer, $z(n)$ is the input signal from the first layer, P is the number of neurons, a is the bias value at the output layer, and $f(\ )$ is the activation function.

Each neuron value at the hidden layer is calculated by:

$$z(n)=f\left(\left(\sum_{i=1}^{N} w_i(n)x_i(n)\right)+b_i\right) \quad \textbf{(3-29)}$$

where $\vec{w}(n)$ is the first weight vector at the hidden layer, $x(n)$ is the reference signal, b is the bias value at the hidden layer, and N is the number of inputs to the neuron.

The model of a single neuron at the hidden layer is given in Figure 3.10.

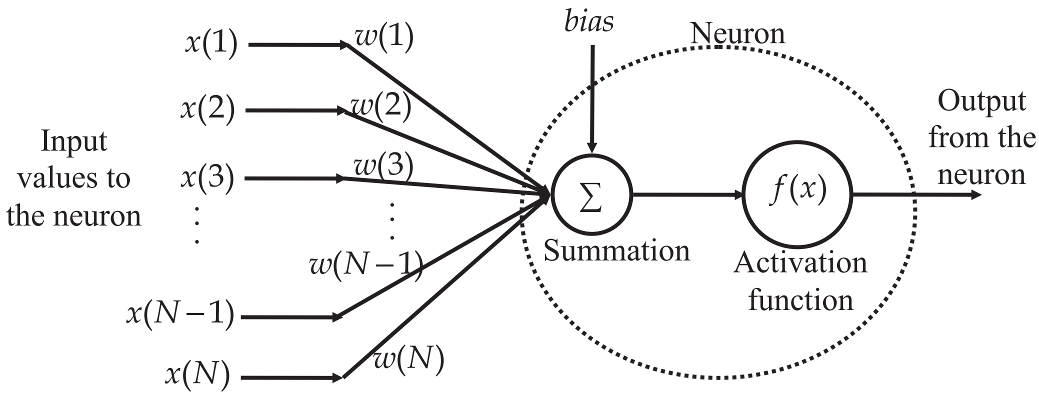


Figure 3.10. Single Neuron Model Structure

At the start, all the weights are initialised to random values between 1 and -1. These weights are updated at each iteration using a training (learning) algorithm.

Learning Algorithm

Learning algorithms are used for the NN structure, to train the network's weight to approximate the behaviour of the system. This idea of training or learning is the same as adapting the weight vectors in the adaptive filtering area. Figure 3.11 shows how the learning algorithm is used in the NN based adaptive filter structure. Figure 3.11 (a) shows the structure during the training period (i.e. when the VAD output is **False -** the weight vectors are updated). Figure 3.11 (b) shows the structure during the non-training period (i.e. when the VAD output is **True -** there is no updating done).

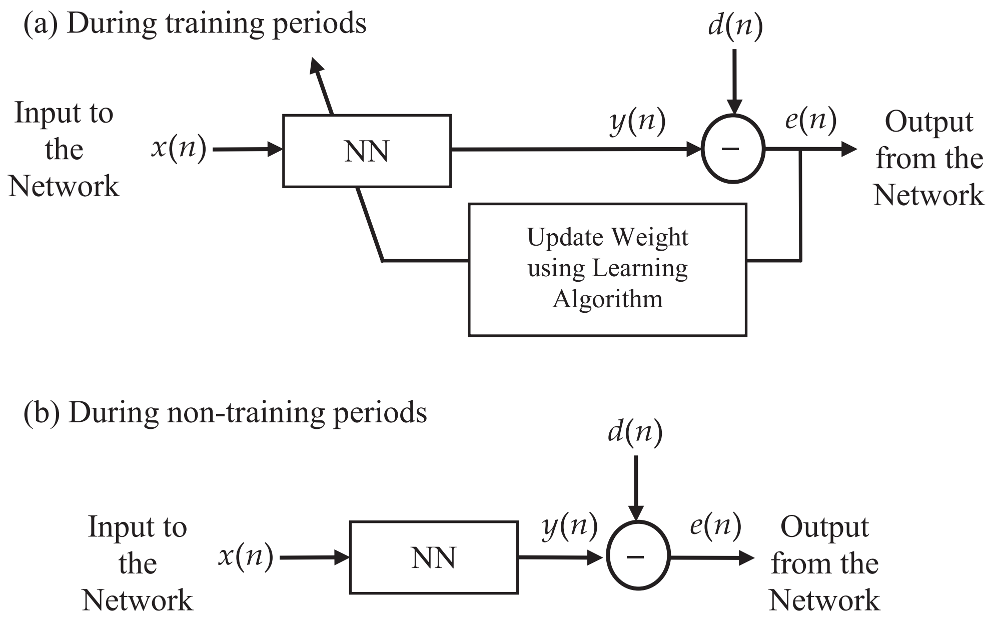


Figure 3.11 Applying the Learning Algorithm in the NN Adaptive Filter Structure

There is a wide range of learning algorithms that are available in the literature. They vary from supervised, and unsupervised to reinforcement learning rules. Examples of the learning rules vary from perceptron (Threshold Learning Unit), and Delta-rule to the back-propagation algorithm. The error back-propagation is one of the most commonly used learning algorithms to train the FF NN systems (Widrow & Lehr, 1993). It is efficiently used for updating the weights in the network and minimising the error value.

Using this algorithm the new weight values are given by:

$$w(n+1)=w(n)+\big(h1e(n)v(n)z'(n)x(n)\big) \tag{3-30}$$

$$v(n+1)=v(n)+\big(h2e(n)y'(n)z(n)\big) \tag{3-31}$$

where $h1$ and $h2$ are the learning rate, $z'(k)$ is the derivative of the hidden layer neuron, and $y'(k)$ is the derivative of the output layer neuron.

This algorithm is sometimes considered as being too slow if it does not converge. This generally happens in cases where the network size is too small when compared to the problem. However, there are other options available to improve the performance of the learning process; a summary of these can be found in Sarkar (1995).

Momentum Term

A momentum term can be used to improve the performance of the back-propagation algorithm and also the local minima problem can be reduced. The advantages of using the momentum term have been studied by many researchers (Abbas, Ahmad, & Bangyal, 2010).

The new weight vector including the momentum term is given by:

$$w(n+1) = w(n) + \left(h1e(n)v(n)z'(n)x(n)\right) + \alpha\left(w(n) - w(n-1)\right) \quad \textbf{(3-32)}$$

$$v(n+1) = v(n) + \left(h2e(n)y'(n)z(n)\right) + \beta\left(v(n) - v(n-1)\right) \quad \textbf{(3-33)}$$

where α and β are forgetting factors, which are normally chosen to be less than 1. These values control how much the new weight value changes from one-iteration to the next-iteration.

The Labview software implementation of this single hidden layer FF NN with input delay is given in Figure 3.12. This neural beamformer structure proposed here can be easily extended to a three-microphone or a four-microphone neural SGJBF system (as shown in Chapter 4).

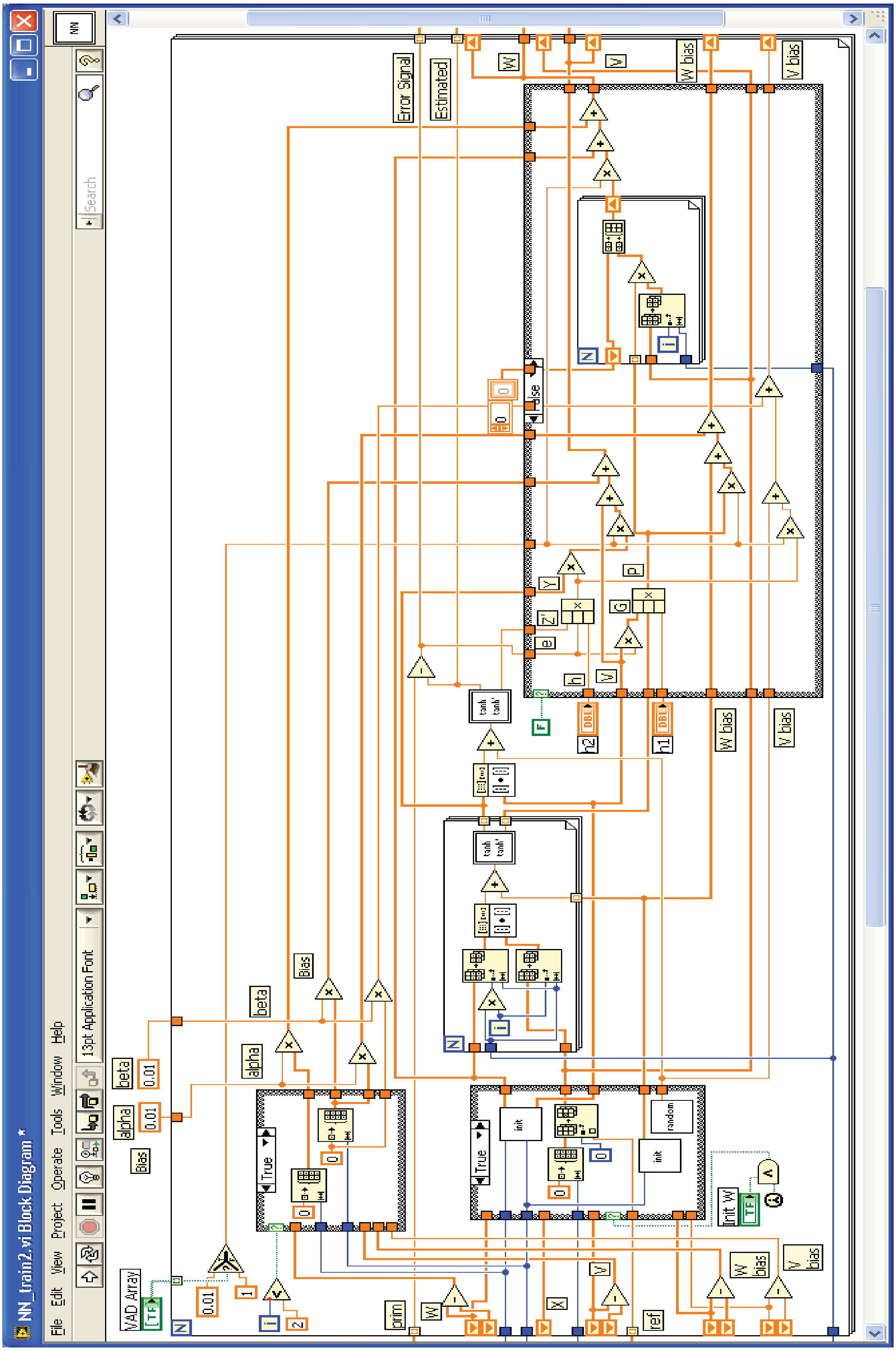


Figure 3.12. IDNN Implementation in the Labview Software

3.2.2 Voice Activity Detector

A Voice Activity Detector (VAD) is used to analyse the acquired signal to determine if it contains a speech signal or not. This is also referred to as a speech detection algorithm. A reliable VAD is essential in order to prevent the speech signal from becoming distorted or cancelled. A VAD based on the variance of the input signal is used here. This is a much simpler approach to determine the presence of speech, but there are other more complicated algorithms that exist, which could be used instead (as discussed in section 2.4). Also, since the proposed beamformer algorithm requires a lot of processing, this VAD algorithm is chosen to maintain the overall real-time processing of the application. A brief explanation of this VAD algorithm is given below.

Traditionally, the variance of the signal is calculated by squaring the standard deviation value (σ), and it is given by:

$$\text{Variance}(X) = \sigma^2 = \int (x-\mu)^2 f(x)dx = \int x^2 f(x)dx - \mu^2 \qquad \textbf{(3-34)}$$

where μ is the expected value of x, and it is given by:

$$\mu = \int x f(x)dx \qquad \textbf{(3-35)}$$

This equation requires the calculation of the mean for a set of samples, and it is processed as batches. For a large number of samples, this algorithm will not be able to track any sudden changes in the noise source. It is important here to recognise these noise variances as it might lead to an incorrect result. An algorithm that can track a non-stationary noise source is required. Therefore, another method called "automatic variance estimator" is used here to calculate the time varying variance sample by sample (Moir, 2001a). This method is a form of automatic gain controller, but it is calibrated to calculate the variance rather than the amplitude. Generally, automatic gain controllers are used in control applications to amplify the signals. Figure 3.13 shows a block diagram of the automatic variance estimator used here (Moir, 2012).

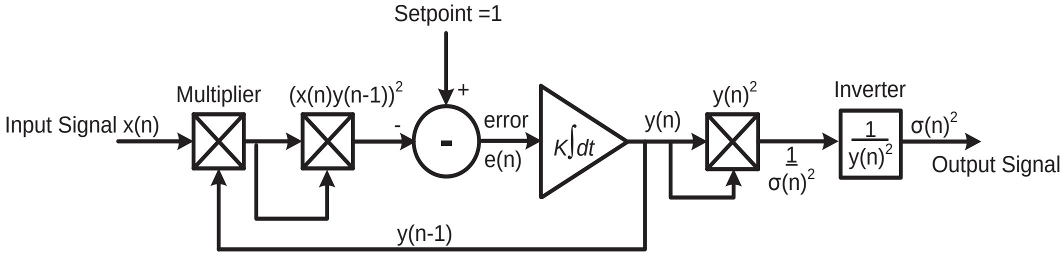


Figure 3.13. Automatic Variance Estimator

An iterative algorithm is used to calculate the variance in real-time at each sample by sample basis and for the n^{th} iteration it is given by:

$$\text{Variance} = \sigma(n)^2 = \frac{1}{y(n)^2} \qquad \textbf{(3-36)}$$

where $y(n)$ is the output from the integrator, and it is given by:

$$y(n) = K\int e(n)dt = Ke(n) + y(n-1) \qquad \textbf{(3-37)}$$

where K is a really small value called threshold gain, $e(n)$ is the output from the summing junction and $y(n-1)$ is the integrator output from the previous iteration. Threshold gain limits the amount of time varying information from the input to the output signal. Hence, choosing a really small value will smooth the output graph, or choosing a larger value will track all the small movements in the input signal.

The output from the summing junction is given by:

$$e(n)=\text{setpoint} - \left(x(n)y(n-1)\right)^2 \qquad \textbf{(3-38)}$$

where $x(n)$ is the input signal which contains a noisy speech signal and the set point is generally chosen as 1 for simplicity.

A threshold value is used to distinguish between the noise-alone signal and the speech signal. This threshold value is chosen at run-time, depending on the level and type of the background noise in the target application. A comparison of the calculated variance value is carried out with the chosen threshold value. If the calculated value is greater than the threshold value, then the VAD output is equal to "True", or else it is equal to "False". The output from this VAD function is used to control the update of the adaptive filter weights. The result from the VAD algorithm only allows one filter to be updated at any given time. Figure 3.14 shows the overall process of the VAD algorithm in this chosen beamformer algorithm.

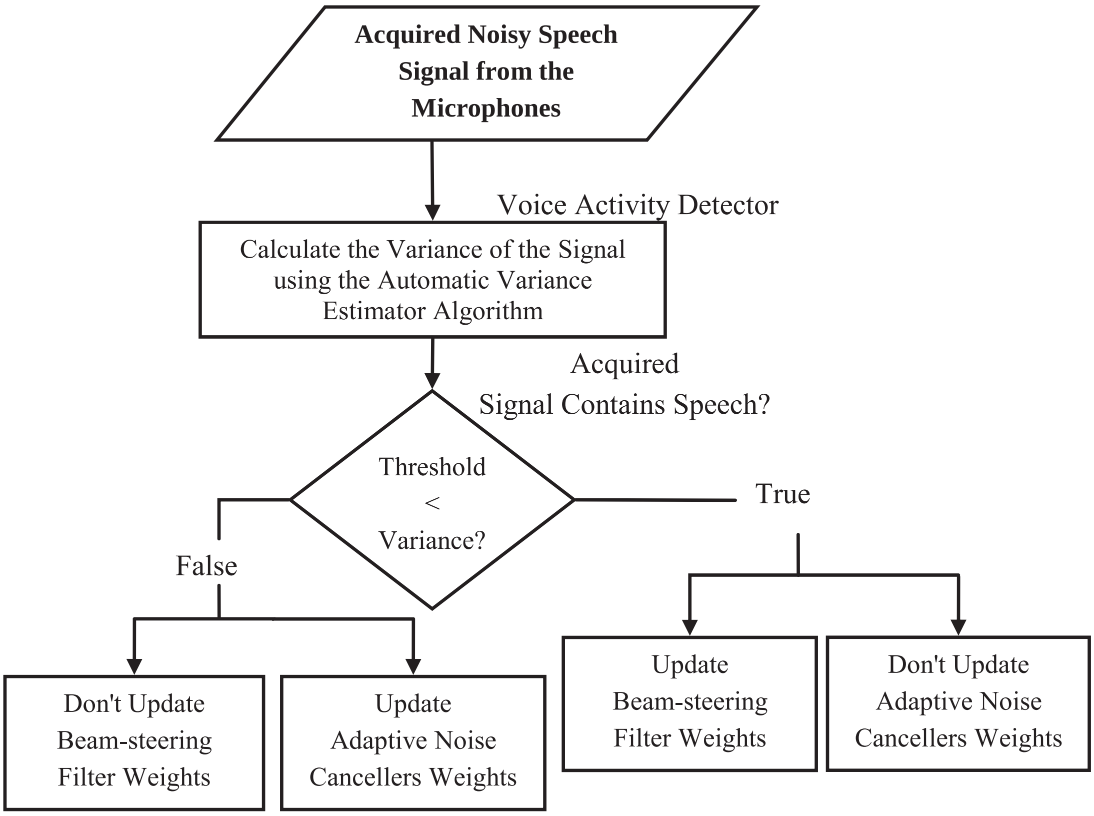


Figure 3.14. Overall Process of the Voice Activity Detector Algorithm

Implementation of the above algorithm is carried out in the Labview software. Figure 3.15 shows the screen shot of this program. At the start of the program, the initial $y(n-1)$ and the "set point" value are set to 1.

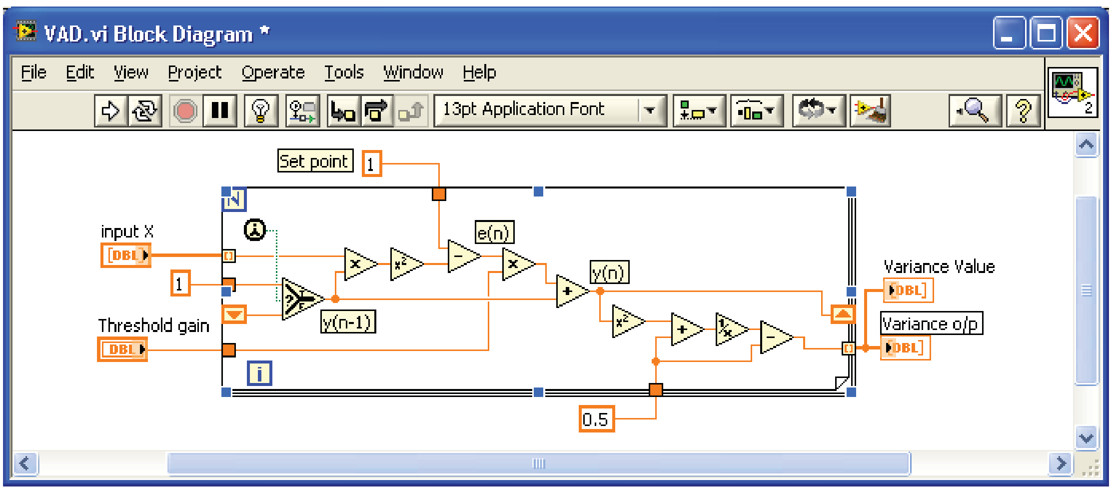


Figure 3.15. VAD Based on Automatic Variance Estimation

The drawback with this method is that it requires the power of speech to be higher than the noise power. However, the NN noise reduction algorithm is the main focus of this research and this VAD function can be easily replaced to improve the performance of the system at low SNRs.

3.3 Summary

In conclusion, a novel NN based switching adaptive filter structure is proposed here for real-time speech enhancement. A single hidden layer IDNN structure is used for noise reduction. This proposed system will be referred to as the Neural Switched Griffiths-Jim beamformer (Neural SGJBF) algorithm. This is a non-linear version of the well-known Van Compernolle's (1990) switching adaptive filter structure. This NN based adaptive noise canceller structure is a very powerful novel contribution to this area; as this structure has not been used in this way before with NN. This neural adaptive filter algorithm is a generic algorithm and can be used for any application that requires non-linear adaptive filtering.

In the following chapter, the VAD algorithm, the NN based adaptive filter and the proposed NN based beamformer algorithms are tested in different situations, to show the capability of this algorithm.

CHAPTER 4

EXPERIMENTAL RESULTS FROM THE SPEECH ENHANCEMENT ALGORITHM

This chapter evaluates the performance of the developed speech enhancement algorithm. The performance of the voice activity detector (VAD) and the adaptive filters are tested individually. The algorithms are first tested with computer simulations and then they are tested with real-world recordings. In addition to this, the use of more microphones is also tested with real-world recordings.

4.1 Experimental Results from the Variance Based VAD Algorithm

This section analyses the capability of the VAD algorithm under different types of noisy backgrounds. Several experiments were performed using the VAD algorithm based on the automatic variance estimator (explained earlier in Chapter 3, section 3.2.2).

The following pseudo code is used to discriminate between the presence of the "speech and noise" segment and "noise-only" segments.

If (Variance > Threshold value) then

VAD=True

Else

VAD=False

End

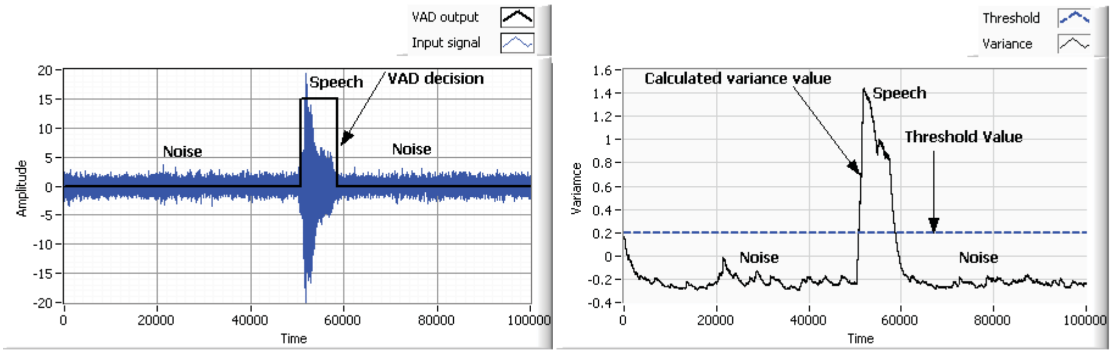
The VAD is turned on when the calculated variance value is above the threshold value and it is turned off when the calculated variance value is below the threshold value. This VAD output is used to control the update of the adaptive filter weights.

Initial threshold value is determined after calculating the variance of the noise-only segment. For all the experiments shown here, a value above the maximum noise variance is chosen as the threshold value. Since the background noise is non-stationary and the type of noise will be different for each application, the user is required to set a different threshold value for each experiment.

4.1.1 Experiment 1

This experiment is carried out to test the VAD algorithm in a simple situation, with a low level of stationary background noise and an isolated speech signal. A simple real-world recording was carried out to record an isolated speech signal at a low level of background noise. The speech signal contains the command "one". The background babble noise was taken from the Noise-X database and played through a speaker ("NOISEX-92," 1996). An initial system configuration was done to determine the threshold value. The user is able to observe the calculated variance graph, and decide the initial threshold value. The threshold value is chosen slightly above the maximum variance value of the "noise-alone" period. For this experiment, the threshold value was set to 0.2. After several trials, a value of 0.0003 was chosen as an appropriate threshold gain. It is found that a lower threshold gain gives a much smoother variance graph.

Figure 4.1 (a) shows the input signal to the VAD function and the output VAD decision from the VAD algorithm. This VAD decision is used later on, to control the update of the adaptive filters. Figure 4.1 (b) shows the calculated variance value from the VAD function and the chosen threshold value.



(a)Input signal and VAD decision (b) Variance graph

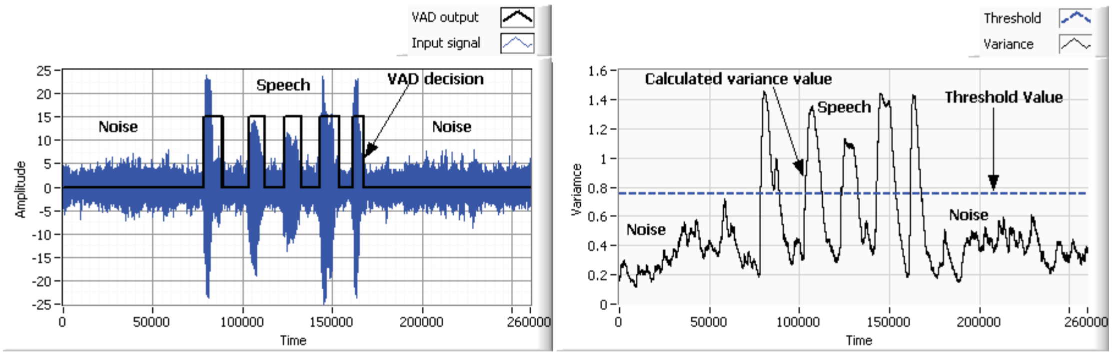
Figure 4.1. Experiment 1 Output Results from the VAD Function

As is noticeable from the noise variance results, the noise is quite constant and can be easily distinguishable from the variance of the speech signal. This is a simple example that does not create any problems for the VAD algorithm.

4.1.2 Experiment 2

This experiment is performed to test a situation, with several speech commands at a slightly higher level of non-stationary background noise. Another recording similar to experiment 1 was carried out, with the input signal containing the speech commands "one, two, three, four, five" and a slightly higher level of background noise. The background contains the factory noise which was obtained from the Noise-X database. After an initial test run, the threshold value of 0.75 and threshold gain of 9e-5 is chosen here for this experiment.

Figure 4.2 (a) shows the input signal to the VAD function and the output VAD decision from the VAD algorithm. Figure 4.2 (b) shows the calculated variance value from the VAD function and the chosen threshold value.



(a)Input signal and VAD decision (b) Variance graph

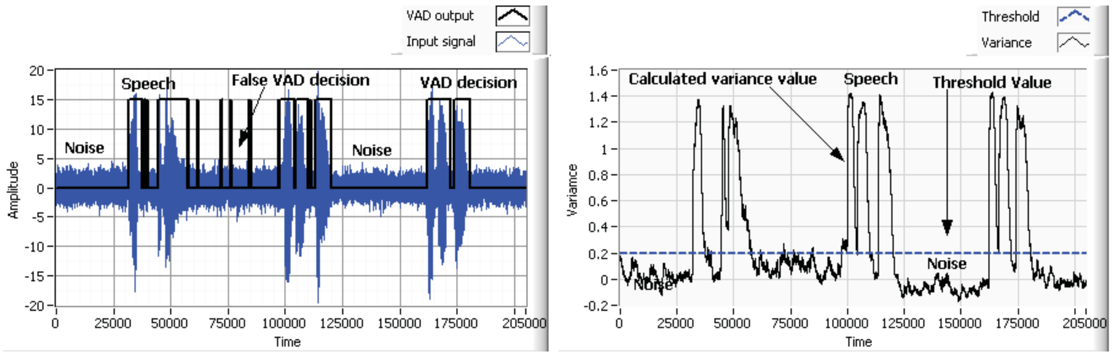
Figure 4.2. Experiment 2 Output Results from the VAD Function

It is clear from the results that the variance of the speech is still much higher than the background noise and can be easily identified. Therefore, it can be seen from these experiments that this type of energy based VAD algorithm works well under low to medium levels of background noise.

The rest of the experiments will demonstrate the capability of our VAD algorithm in several extreme cases. Experiment 3 shows the importance of choosing an appropriate threshold value and experiment 4 shows a really extreme situation, with a high level of non-stationary background noise. A solution to these problems will also be discussed here.

4.1.3 Experiment 3

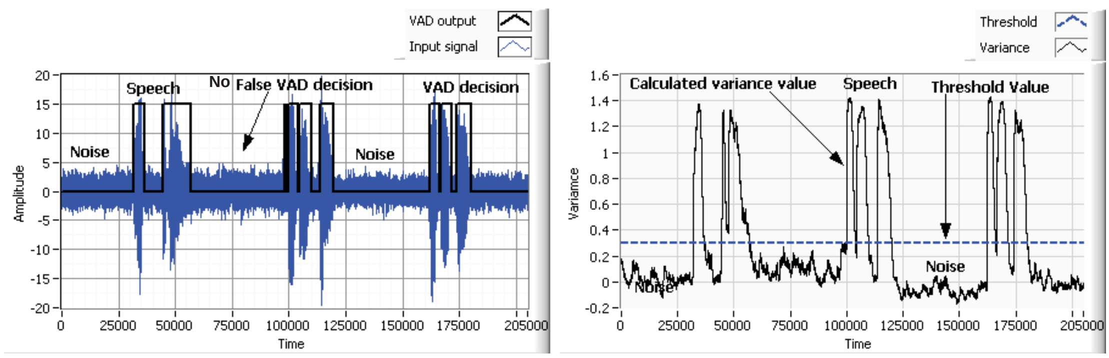
This experiment is performed to test the importance of choosing an appropriate threshold value. For this experiment, the speech signal contains several speech commands and a high level of non-stationary background noise. The speech command contains “open file”, “start acquire” and “start analyse”. The background noise is the factory noise obtained from the Noise-X database. Figure 4.3 shows an example output from the low SNR situation. After an initial test run, the threshold value of 0.2 and threshold gain of 0.0003 is chosen here for this experiment. As can be seen from the graph, there are several false VAD decisions.



(a)Input signal and VAD decision (b) Variance graph

Figure 4.3. Experiment 3 Results from the VAD Function (Threshold Value=0.2)

Choosing the correct threshold value is somewhat important in our algorithm as this could lead to false VAD detections, and the start and end of the speech signal being cut off. Figure 4.4 shows another instance of this recording with a higher threshold value of 0.3. As can be seen from the graph, the VAD decision is correctly identified, but there is a possibility of the start and end of the signal being cut off.



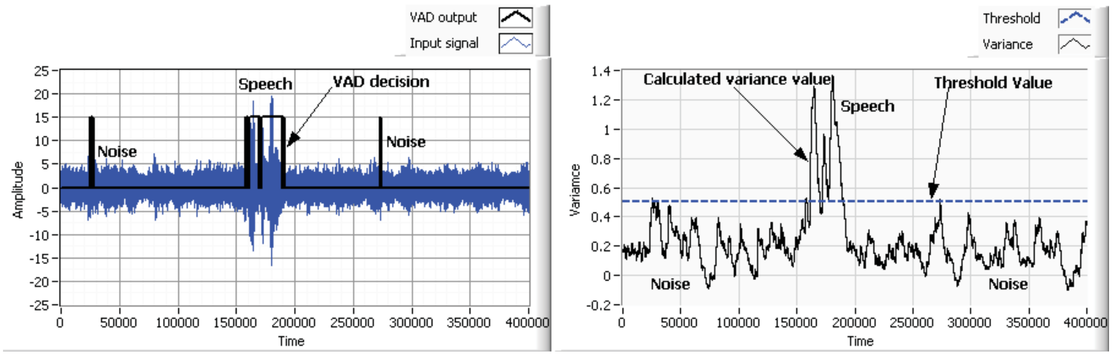
(a)Input signal and VAD decision (b) Variance graph

Figure 4.4. Experiment 3 Results from the VAD Function (Threshold Value=0.3)

The false VAD decision can be avoided by choosing a higher threshold value, but this could also create another problem of the start and end of the speech signal being cut off. This is the main disadvantage with this type of VAD algorithm. For our application, it is fairly important to avoid the false detection during the start and end of the speech signal. As a result, it could lead to a distorted speech signal or to false speech recognition. In the next experiment, an extreme case where the VAD will struggle to function is tested and a solution to this problem is proposed.

4.1.4 Experiment 4

This experiment is done to test an extreme case where the background noise is non-stationary and is really high. This experiment has the speech command "start acquire" and factory noise as the background noise. (This file is also from the Noise-X database). This recording has a high level of background noise (also known as the low SNR situation). Figure 4.5 shows an example output of this situation. After an initial test run, the threshold value of 0.5 and threshold gain of 9e-5 is chosen here for this experiment.

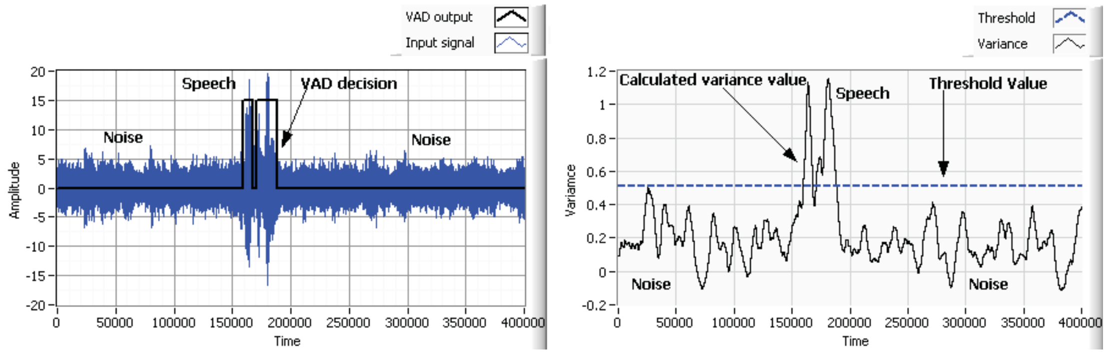


(a)Input signal and VAD decision (b) Variance graph

Figure 4.5. Experiment 4 Output Results from the VAD Function

As can be seen from the graph, the variance of speech and variance of the noise-alone period are quite close to each other when compared to the other experiments. A higher threshold value can be chosen to avoid most of the false VAD decisions, but there are still a few anomalies in the VAD results. However, doing this led to the start and end of the speech signal being cut off. Since this algorithm is mostly dependent on the energy of the signal, this sort of behaviour is to be expected. For our application, it is important to make sure we get the start and end of the speech signal as this can lead to a false recognition.

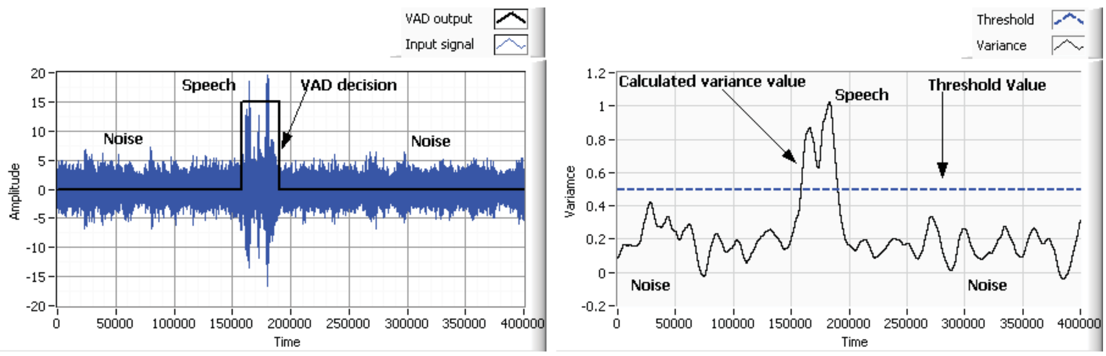
In order to avoid the problems discussed earlier a moving average of the variance is calculated. A set of variance values are averaged, and a much smoother variance output is calculated. A different buffer size is tested, and the number of samples is chosen to be 2500 for this experiment. After an initial test run, the threshold value of 0.51 and threshold gain of 0.0003, are chosen here for this experiment. An example of the result is shown in Figure 4.6. As can be seen from the graph this combination gives much smoother variance values in the noisy signal period.



(a)Input signal and VAD decision (b) Moving average variance graph

Figure 4.6. Experiment 4 Improved Results (Threshold Gain of 0.0003)

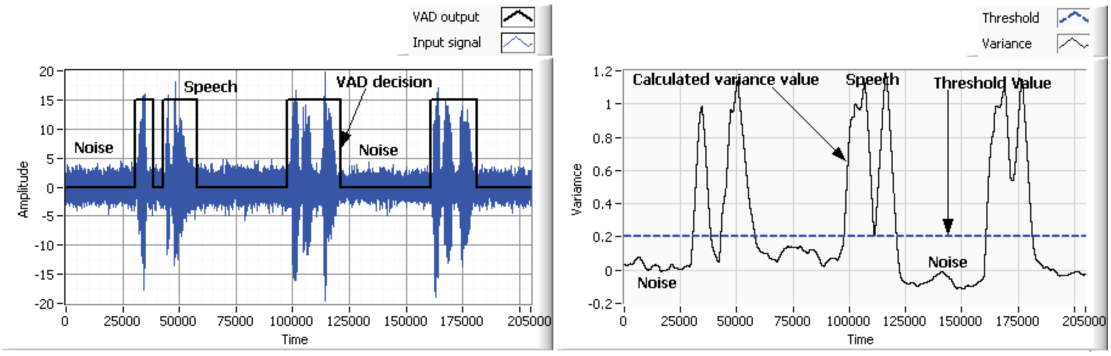
Another experiment is done with the threshold value of 0.5, threshold gain of 9e-5 and with 5000 samples chosen for the averaging. An example of the result is shown in Figure 4.7. The results confirm that there is no single correct threshold value. Initial testing has to be done to approximate this value for different background noise.



(a)Input signal and VAD decision (b) Moving average variance graph

Figure 4.7. Experiment 4 Improved Results (Threshold Gain of 9e-5)

Figure 4.8 shows the results from experiment 3 using the new improved averaging method. After an initial test run, the threshold value of 0.2 and threshold gain of 0.003 is chosen here for this experiment.



(a)Input signal and VAD decision (b) Moving average variance graph

Figure 4.8. Using the Moving Average Algorithm to Detect the Speech for the Recording Done in Experiment 3

As can be seen from the results, this solves the problem of the start and end point being cut off and also there are no VAD decision errors in the results. In Figure 4.8 it should be noted that the VAD decision of the second and third commands are considered as one speech; and the VAD decision is true for the entire length of the command. It is likely that the noise is constant during this period; therefore, it is not necessary to turn on and off the ANC in between the words.

This section tests the automatic variance estimator based VAD algorithm. After finding a few limitations in the current algorithm, this method was improved with a moving average of the output variance value. This modified automatic variance estimator algorithm is proposed to be used with the beamformer algorithm. This method was found to be effective under stationary and non-stationary noise sources.

4.2 Testing the Neural Network based Adaptive Filter

The single hidden layer IDNN based adaptive filter is implemented using Labview in a fully connected (FC) and partially connected (non-fully connected or NFC) structure. Both NN systems use the back-propagation algorithm to train the weights. It is found that the convergence of this algorithm depends on the initial weights, learning rate, the momentum term and the network architecture (Hamid, Nawi, Ghazali, & Salleh, 2011). In the following experiments, the convergence behaviour of the back-propagation is analysed.

4.2.1 Network Size

There is no set theory behind choosing the size of the network. Choosing an appropriate network size depends on the problem at hand and it is often chosen experimentally. The general concept is that the number of neuron size chosen is to be larger than the problem at hand (Zhongguang & Zongyuan, 2000). As a result, there may be several architectures that can give promising results.

Note that short-hand notation for describing the architecture/topology is used where by 10-5-1, denotes ten in the input layer, five in the hidden layer, and one in the output layer.

4.2.2 Effect of Learning Rate

The first experiment is carried out to show the effect of learning rate (LR) in a FC network structure and NFC structure. For this experiment the input signals are simulated according to Figure 4.9.

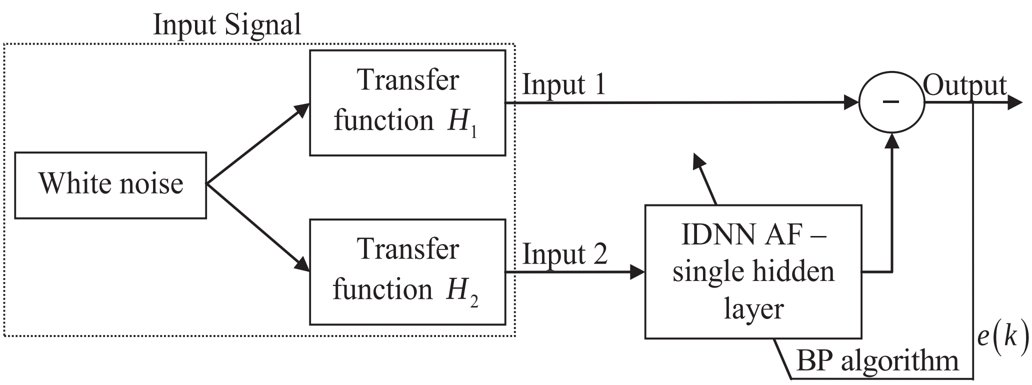


Figure 4.9. Simulation Setup to Test the NN Based Adaptive Filter

Gaussian white noise is simulated in the Labview software with the standard deviation of 0.01 and 70,000 samples. The two input signals to the adaptive filters are achieved by simulating the noise signal with the transfer functions, H_1 – input 1 and H_2 – input 2. Then input 2 is fed into the NN structure and the error between the signals is minimised, by approximating the network weights using the back-propagation algorithm. Each sample of the input signal is fed into the IDNN structure, and an output error value is calculated, this is then used to update the new weights values for the next sample.

The FC IDNN architecture used here has ten input values, five neurons, and one output value (10-5-1). The NFC IDNN architecture also has ten input values and five hidden neurons, but only five inputs values are passed to hidden layer. The hyperbolic tangent function is used here as the activation function for this structure. The weight vectors are initialised to random values between -0.5 to 0.5. The transfer functions are set to $H_1 = 1$ and $H_2 = 1 - 0.5z^{-1}$.

These two network structures are tested with different values of LR. Figure 4.10 shows the Mean Square Error (MSE) results from the FC and NFC IDNN. Hence, the MSE value shows the rate of convergence of the output error signal. As shown in the figure, a large learning rate gives a much faster convergence in both network structures. However, the FC network performed much better than the NFC network. This is because the NFC requires more time to approximate the function as it only takes a few input values to calculate the neuron.

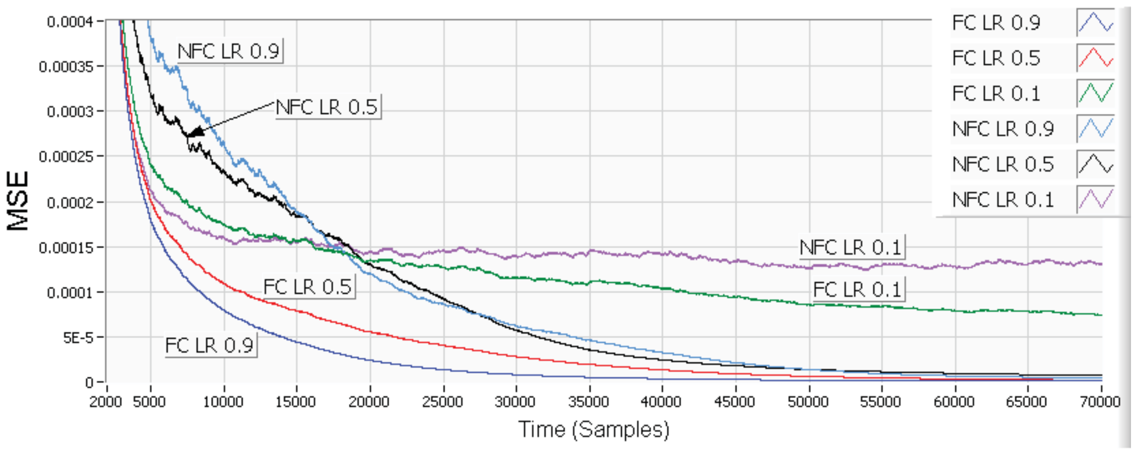


Figure 4.10. Comparison of MSE Results from the FC and NFC - IDNN Adaptive Filter with Different LR

The FC network can be used in cases where the processing time is not the main concern, and the NFC network can be used for real-time processing where a much faster response is required.

4.2.3 Selection of Initial Weights

The weight values are generally initialised to small random values. This experiment shows the effect of the initial weight selection on the output of the network. The experiment setup is kept similar to the above experiment. The network structure is kept the same as above (ten input values and five hidden neurons), and the learning rates are set to 0.5. Four different ranges of random values are tested in this experiment. They are initial weights with random values (-0.05 to 0.05), (-0.25 to 0.25), (-0.5 to 0.5), and (-1 to 1).

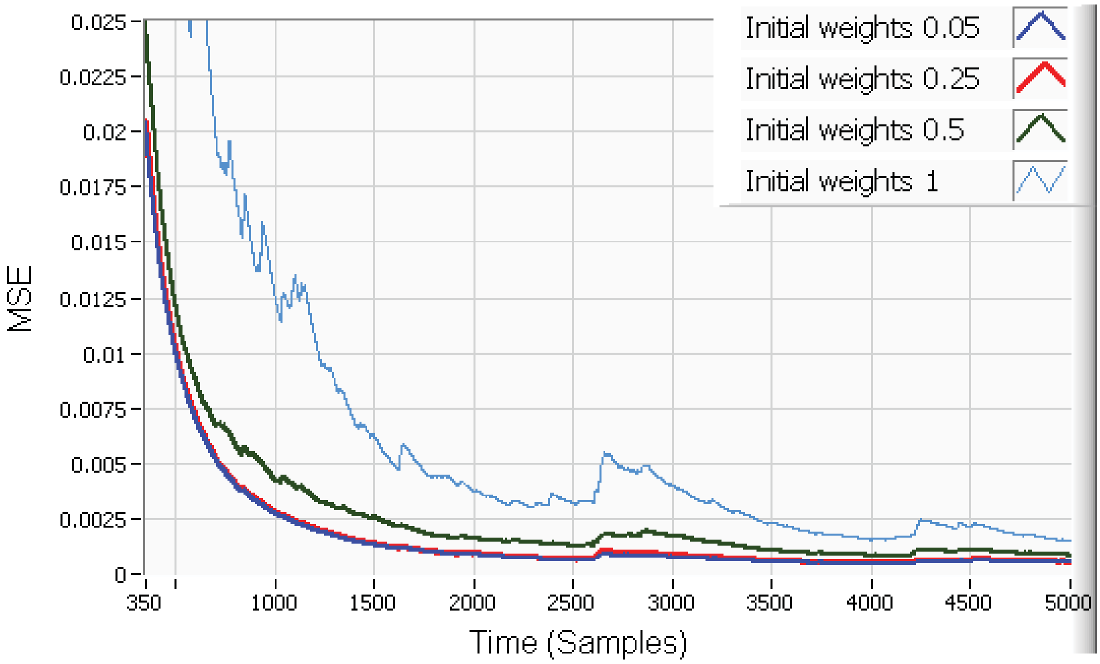


Figure 4.11. MSE Comparisons of Different Initial Weight Values (I-10, N-5, LR-0.5)

The results show that initial convergence is affected by the initial weights but eventually they all approximate the system. A smaller value below 0.25 can be used to reduce the convergence rate of the NN structure. A larger weight value will take longer to converge. Also, it is essential to note that the initial weight value cannot be equal to zero.

4.3 Evaluation of Various Adaptive Filters

This section evaluates the capabilities of several adaptive filter algorithms (LMS, NLMS, APA and NN) in different noise situations. This section is not intended to show that one adaptive filter is better than the other algorithm; it is only carried out to show how each of these adaptive filters perform under different conditions. Appendix B gives details of the Labview software implementation of other adaptive filters used in this experiment.

4.3.1 Experiment 1

This experiment is carried out to verify that the implemented adaptive filters are functioning appropriately, and it is achieved by identifying a defined system. This experiment setup resembles a system identification process. The input signal setup and the experiment simulation setups are shown in Figure 4.12. The Gaussian white noise signal is simulated with the transfer functions H_1 and H_2 to obtain the primary input signals and reference input signal, respectively. These transfer functions are simulated as a representation of the acoustical room response from the noise source to the microphone.

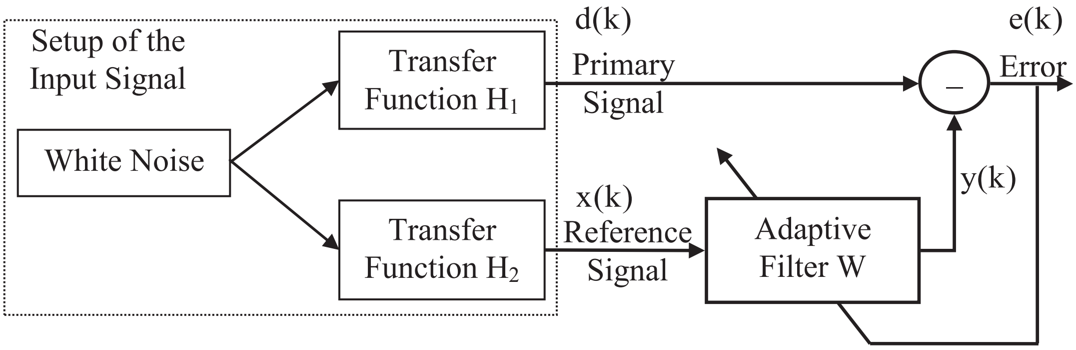


Figure 4.12. Experimental Setup Used to Approximate a Certain Linear Transfer Function

At the k^{th} iteration, the input primary signal $d(k)$ is given as $d(k) = H_1 n(k)$, reference signal $x(k)$ is given as $x(k) = H_2 n(k)$, and the output signal $e(k)$ is given as $e(k) = d(k) - y(k)$.

The adaptive filter's output is given as $y(k) = X^T W$. The aim of the adaptive filter is to approximate the transfer function so that the error between the two signals will minimise to zero. Hence, the weights vector W should approximate to $W = \frac{H_1}{H_2}$.

For this experiment, the input signal $n(k)$ is set up with a Gaussian white noise, which has a standard deviation of 0.055 and zero mean value. The primary signal $d(k)$ has a transfer function H_1 of 1.558-0.81z^{-1}, and the reference signal $x(k)$ has a transfer function H_2 of 1. The rest of the variables in the adaptive filters are set to the values in Table 4.1.

Table 4.1. Variables Used for the Adaptive Filters

APA	N = 20, M = 15, μ = 0.4, δ = 0.1
LMS	N = 20, μ = 0.4
NLMS	N = 20, μ = 0.4, γ =0.0001
Volterra	N = 20, μ = 0.4
NN Fully connected	20 - input nodes, 15 - hidden nodes and 1 - output node Learning rate h1 = 0.4 and h2 = 0.5

where N denotes the number of weights, M denotes projection order, μ denotes step-size, γ denotes gamma value, and δ denotes regularisation factor. More details on this can be found in section 2.2.

The first test is to observe whether the output weight values of the adaptive filters are approximating to the designed transfer function of the system. The NN filter was not included in this comparison as the weights do not approximate to the transfer function. Figure 4.13 shows the weight values calculated from the adaptive filters at each sample. As evident from the graph, after 50 iterations the APA algorithm has converged to the appropriate a0 =1.558 and a1=-0.81 values. The NLMS and Volterra algorithms converged after 200 iterations. However, the LMS algorithm took longer to converge. As shown in Figure 4.14, it eventually converged at about 4000 iterations.

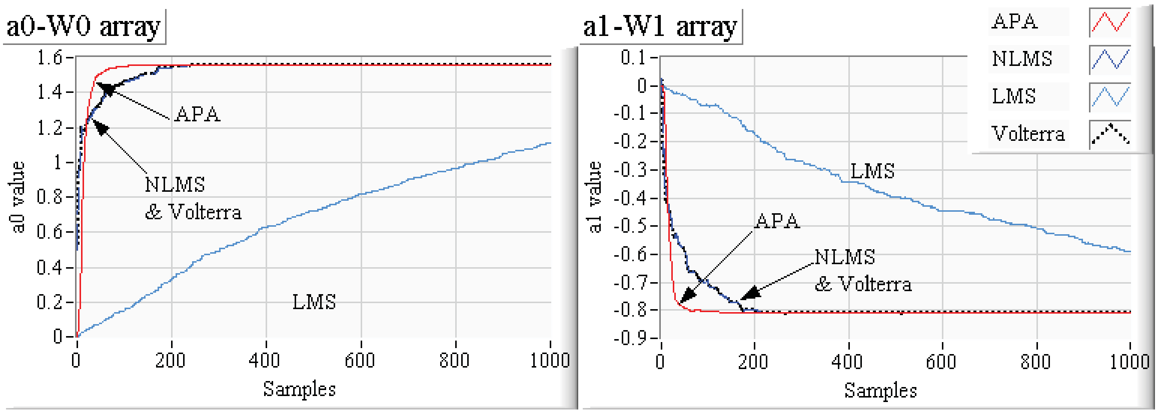


Figure 4.13. Adaptive Filters - Weight Values after 4,000 Samples

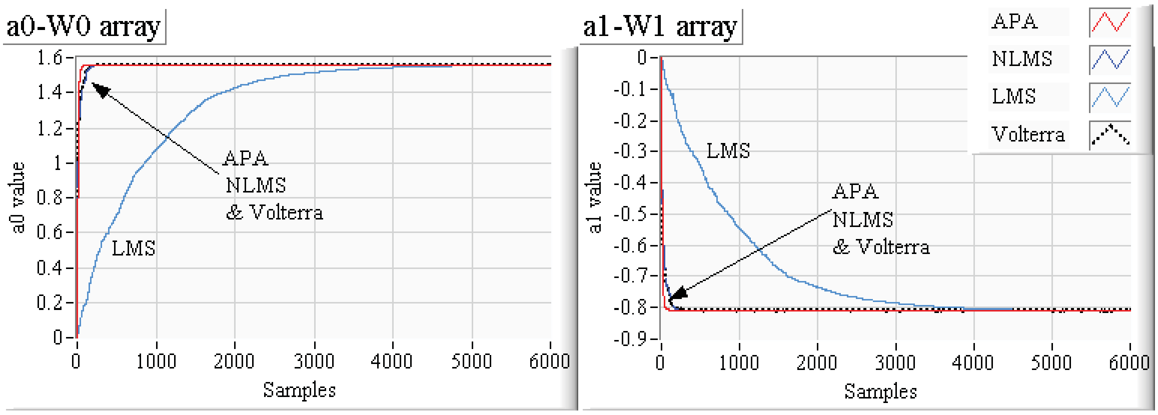


Figure 4.14. Adaptive Filters - Weight Values after 20,000 Samples

The above experiment was carried out with a lower μ value to show the convergence characteristics of the adaptive filters. In certain situations, a higher μ value can give a much faster convergence, although for the LMS algorithm a higher μ value might also cause the system to become unstable. The NLMS is not affected by this instability problem as it normalises the μ value by its input signal.

This experiment shows that even with a lower value of μ, the APA and NLMS algorithm perform well compared to LMS. The convergence rate of the adaptive filter depends on the μ value. Figure 4.15 shows the comparison of MSE for all the adaptive filters. This graph shows that all of the adaptive filters converge, but LMS is slightly slower than the other filters.

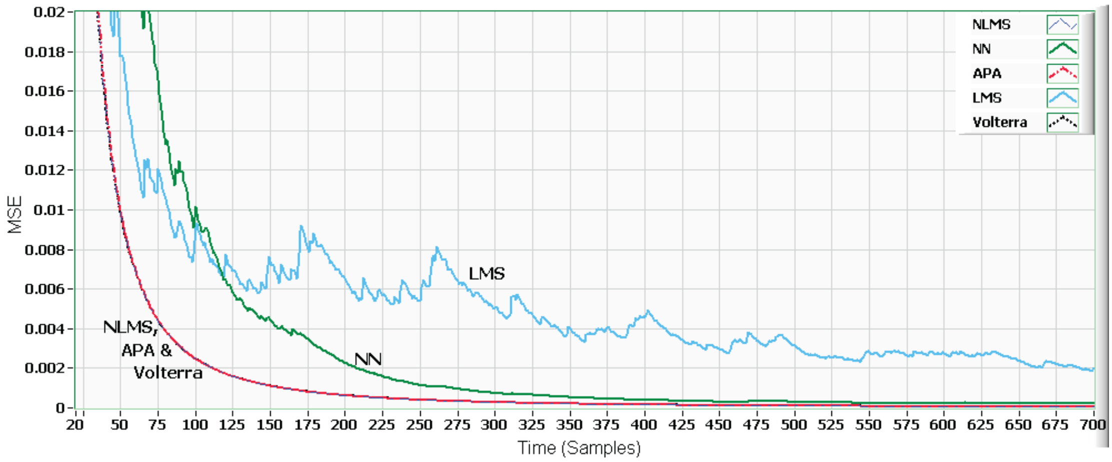


Figure 4.15. Comparison of MSE for all the Adaptive Filters (NLMS, NN, APA, LMS and Volterra Filter)

This experimental setup shows that all the filters are working correctly. This experiment is tested with a linear system where the Least Square (LS) algorithms should not have any trouble approximating. In the next experiment, much more difficult situation is tested, where the LMS algorithms will find it difficult to perform.

4.3.2 Experiment 2

This experiment is carried out to show the performance of the adaptive filters in coloured noise situations. Initially, a Gaussian white noise signal is put through a transfer function, H, to achieve a coloured noise. Then this signal is simulated with the transfer functions, H1 and H2, to obtain the primary and reference signals as per the above experiment. The input signal setup is shown in Figure 4.16.

The fact that the LMS algorithm has a slower convergence rate when the input signals are correlated has been established in the literature (Bilcu, 2004; Haykin, 2002). Therefore, this experiment will test the performance of all the adaptive filter algorithms, with one transfer function with lower eigenvalue spread, and one with higher eigenvalue spread values in coloured noise signals. These transfer functions are obtained from Haykin's 4th edition book and a more detailed explanation on this area can also be found there (Haykin, 2002).

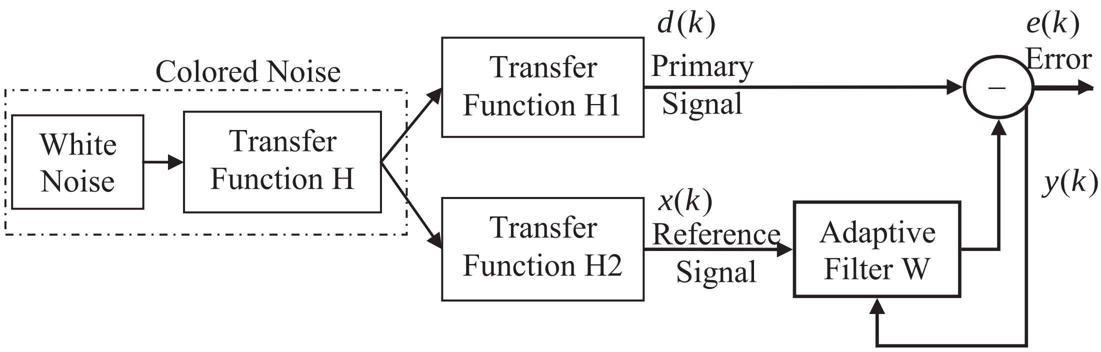


Figure 4.16. Experiment 2 Setup

Case 1.

The transfer function H is set to 1-0.1950z^{-1}+0.95z^{-2}, which will give a small eigenvalue spread of 1.22. The transfer functions, H1, H2 and the rest of the variables in the adaptive filters are kept the same as the first experiment (as shown in Table 4.1). A μ value of 0.4 was chosen for all the adaptive filters as any value above this caused problem for the LMS algorithm.

Figure 4.17 shows the results from this experiment. As can be seen from the graph, the LMS algorithm has a slower convergence rate than the other algorithms. This was to be expected from this type of experimental setup. It is important to note that all the adaptive filters eventually did converge to the weight value (a0 =1.558 and a1=-0.81).

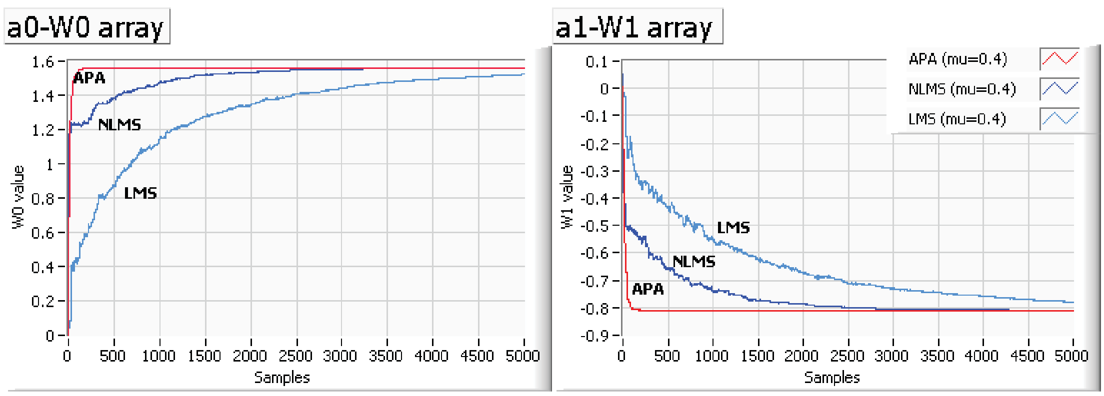


Figure 4.17. Experiment 2 Results with a Small Eigenvalue Spread of 1.22

Figure 4.18 shows a comparison of MSE between all the adaptive filters. As can be seen from the graph, all the adaptive filters have relatively similar results. This was expected from this input signal.

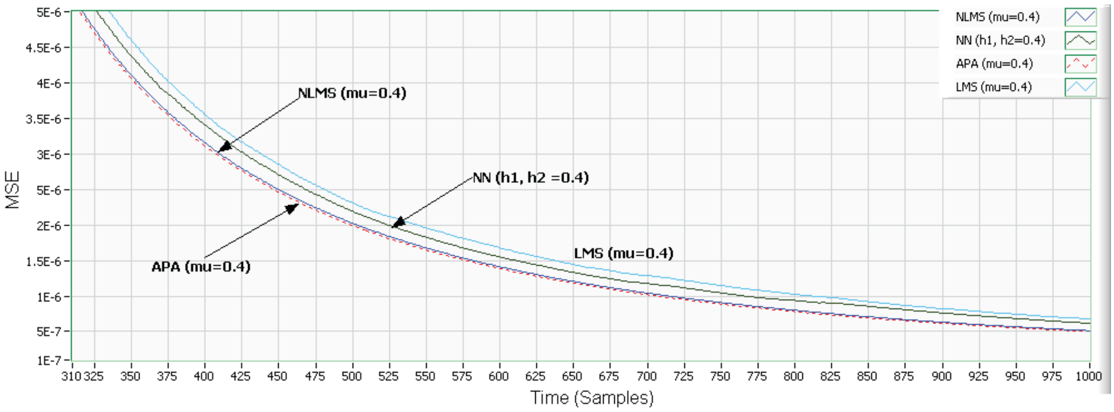


Figure 4.18. MSE Comparisons of the Adaptive Filters - LMS, NLMS, NN and APA (Eigenvalue Spread of 1.22)

Case 2.

The transfer function, H, is changed to $1\text{-}1.9114z^{-1}+0.95z^{-2}$, which will give a large eigenvalue spread of 100. The remaining variables are kept the same as above. In this experiment, different μ values are tested, and the results are analysed. It is found that, for the LMS algorithm, a μ value of 0.01 has to be used to get a valid result, otherwise the algorithm becomes unstable.

Figure 4.19 shows the results from this experiment. It also shows a comparison of the NLMS algorithm with high and low μ values. The APA algorithm and the NLMS algorithm with high μ value converge to the weight value (a0 =1.558 and a1=-0.81) much more quickly than the other algorithms.

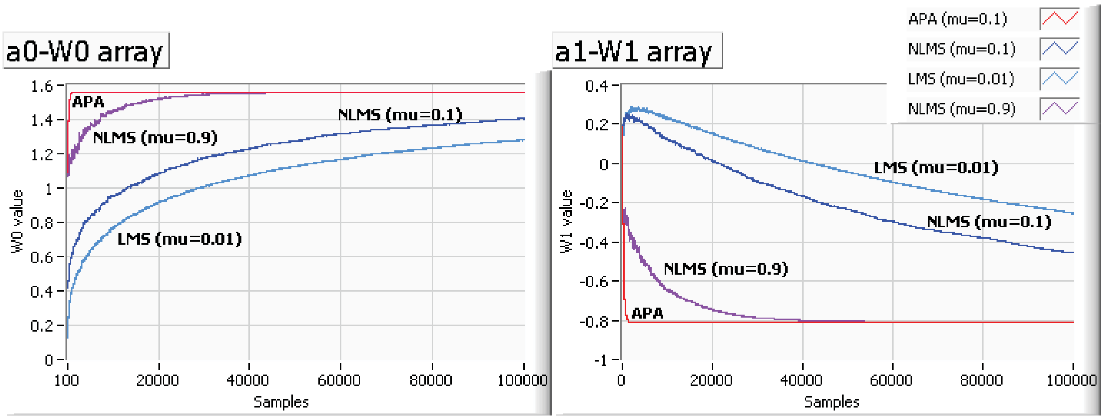


Figure 4.19. Experiment 2 Results with a Large Eigenvalue Spread of 100

Figure 4.20 shows a comparison of MSE between all the adaptive filters. As can be seen from the graph, the LMS algorithm struggles to perform under these conditions. The NLMS algorithm also slightly struggles with a low μ value (μ=0.1). However, the performance of the NLMS algorithm can be improved by using a higher μ value.

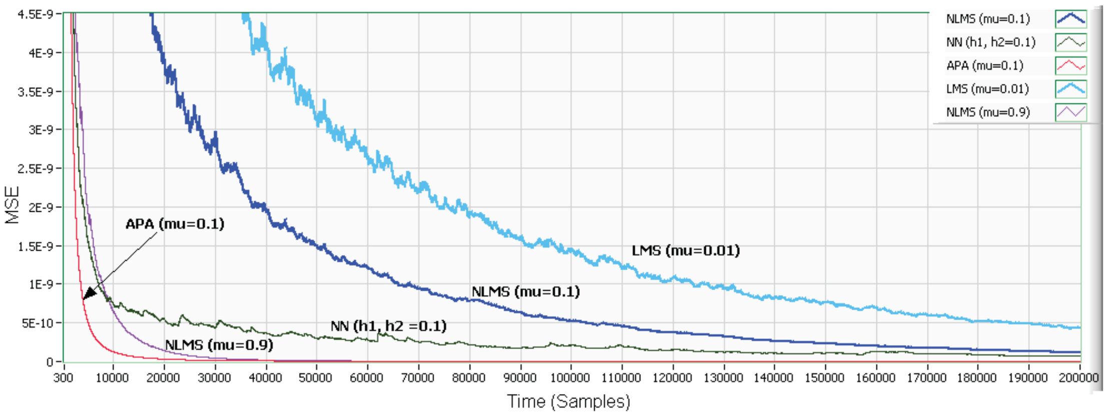


Figure 4.20. MSE Comparisons of the Adaptive Filters – LMS, NLMS, NN and APA (Eigenvalue Spread of 100)

It is shown by these experiments that the rate of convergence depends on the step-size μ and eigenvalue spread of the input signal.

4.4 Real-Time Experiments of the SGJBF

This section tests and compares the performance of the NLMS based SGJBF algorithm with the proposed Neural SGJBF algorithm. Real-time recordings were carried out to test these beamformer algorithms. Initially, dual microphone systems are implemented and tested. Later, this study is extended to show the effects of using more microphones with these algorithms.

Figure 4.21 shows a typical office environment where the recordings are carried out for the remaining experiments. The microphones are mounted on the computer monitor in a linear array about 30 cm apart from each other. The experiment is set up as follows: the target speech signal is generated in front of the two microphones and the interference signal is generated at about a 45 degree angle to the microphones.

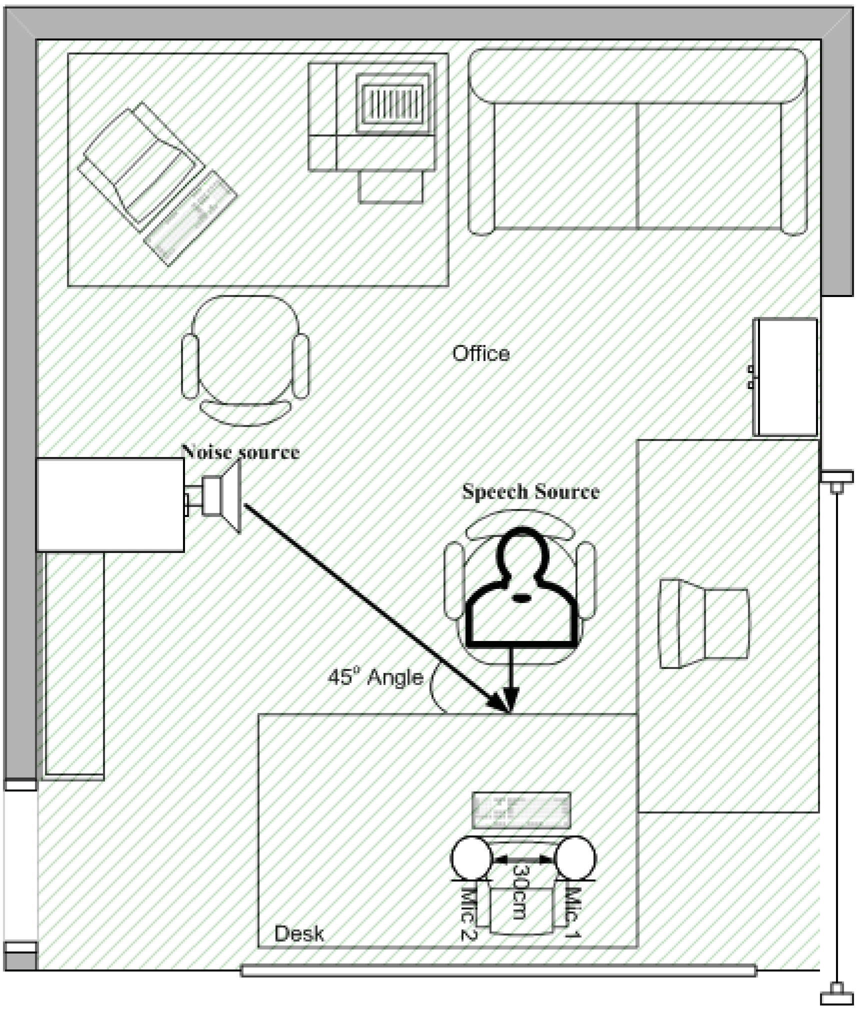


Figure 4.21. Experiment Setup

4.4.1 Experiments with Dual-Microphone SGJBF

Two different arrangements of SGJBF algorithms are developed in the Labview programming language and compared in this section:

- First the proposed method with the first filter based on NLMS and the second filter based on NN is implemented. This arrangement will be referred to as Neural SGJBF. A partially connected IDNN structure with one hidden layer is used here for real-time processing.
- The second method is based on two NLMS. (i.e. the first filter and the second filter both use the NLMS algorithm). This arrangement will be referred to as NLMS based SGJBF.

A VAD based on automatic variance control discussed in section 3.2.2 is used with both of these beamformer arrangements. The parameters used for this experiment are given in Table 4.2.

Table 4.2. Variables Used for the Adaptive Filters

Parameters used for NN adaptive filter	
Three Layer Network – Input node - 20 (1st layer), Hidden node - 17 (2nd layer), Output node - 1 (3rd layer)	
Number of connections in the first layer (from the input nodes to the hidden node)	4 - Partially connected
Number of connections in the second layer (from the hidden nodes to the output node)	17 - Fully connected
Iterations of training given for the NN	100
Learning rates h1 and h2	0.8
Parameters used for NLMS adaptive filter	
No of weights coefficients N	85
Step-size μ	0.8
Gamma value γ	0.0001

The quality of the output is generally accessed using the signal-to-noise ratio (SNR). This provides a ratio of how much noise is present in the output of the signal compared to the input signal.

The quality of the outputs signal is measured by the SNR and is calculated using:

$$\text{Signal to Noise Ratio (SNR)} = 10\log_{10}\left(\frac{P_{\text{Signal+Noise}} - P_{\text{Noise}}}{P_{\text{Noise}}}\right) \quad \textbf{(4-1)}$$

where $P_{\text{Signal+Noise}}$ indicates the power of the signal and noise and P_{Noise} indicates the power of the noise alone signal.

For the first experiment, the speech signal contains the word “one” and the white noise as interference signal from the Noise-X database. One example signal from this experiment is presented in Figure 4.22. The input signal contains the word “one” and the interference signal was played through a speaker. Figure 4.22 (a) shows the input primary signal with a SNR of 11.48 dB and the VAD results, and Figure 4.22 (b) shows the input reference signal with a SNR of 12.14 dB. Figure 4.22 (c) shows the output signal from the NLMS based beamformer with a SNR of 16.17 dB and Figure 4.22 (d) shows the NN based beamformer with a SNR of 24.89 dB.

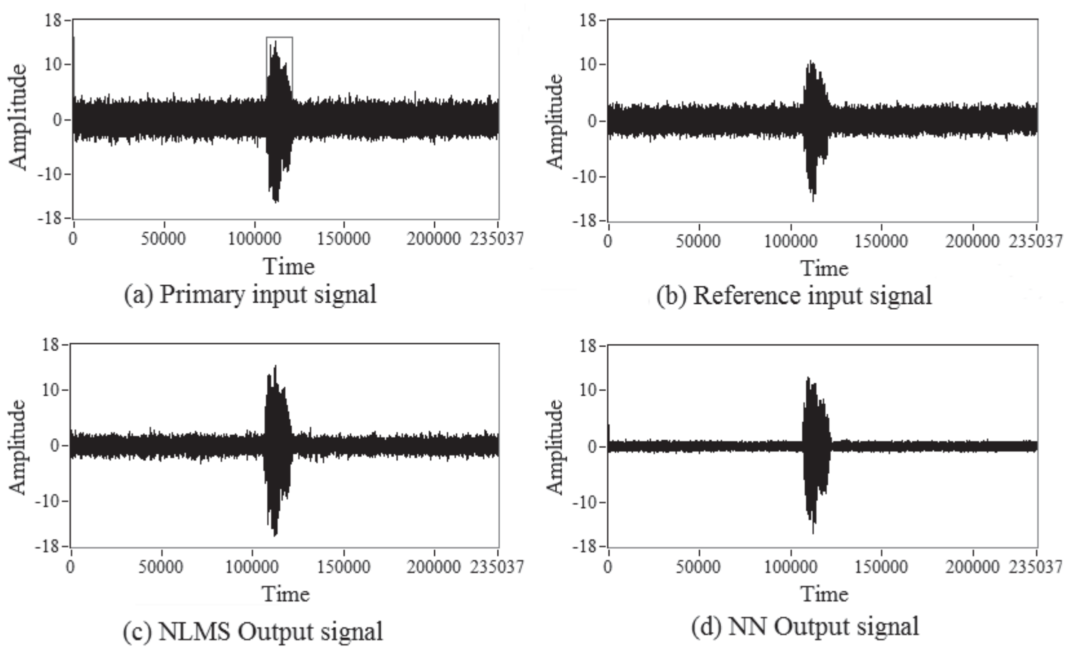


Figure 4.22. Input and Output Signal from the White Noise Interference Experiment

As can be seen from the results, the NN based beamformer algorithm performed much better than the NLMS based beamformer. This is quite possibly because NN is able to handle non-linear adaptive filtering when compared to the traditional the NLMS filter. In the earlier simulation section, NLMS filter output outperformed the NN filter. The transfer functions of the room acoustics are unknown in the real-world recording. In theory, it is well-known that non-Gaussian noise signals require non-linear adaptive filtering. The background noise in this experiment might have created a non-linear transfer function in the real-world situation or it could be highly correlated. As a result of these, NLMS has not performed as well as it did with the simulations done in the earlier section.

This result is further analysed by comparing the MSE of the output signals. Figure 4.23 shows the comparison of the MSE of the output signal for the NLMS and NN based beamformers. As can be seen from the graph, the NN converged more smoothly than the NLMS. This shows that the NN has a better convergence rate than the traditional NLMS algorithm.

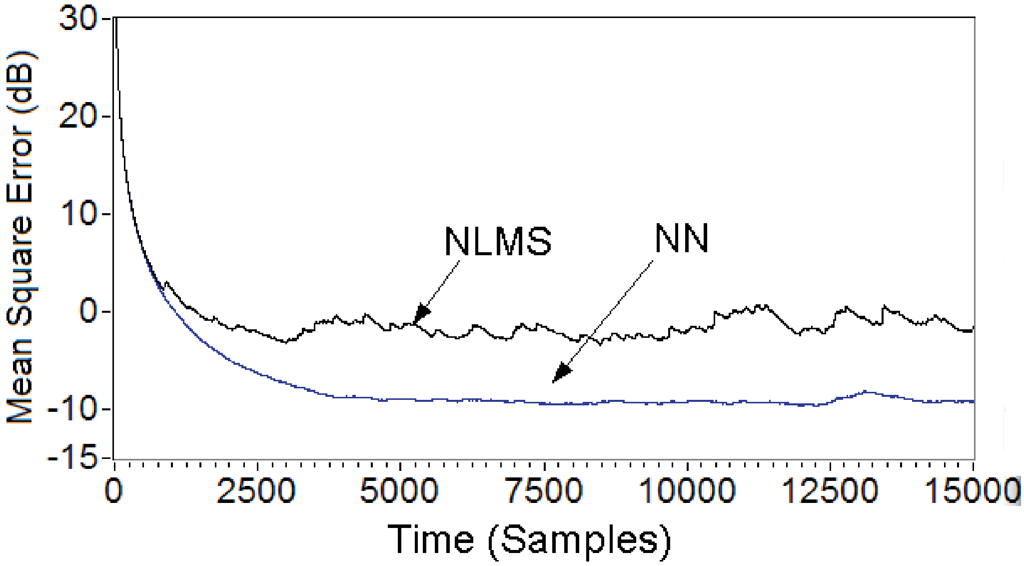


Figure 4.23. MSE (dB) Comparison of the Dual Microphone NLMS Based SGJBF and NN Based SGJBF Output Signals

Several more experiments were conducted by varying the level of the white noise, while keeping all the other variables the same as in the experiment above. Figure

4.24 shows the comparison of the input SNR versus the output SNR for both NLMS and NN based beamformers. It is evident that with the real-world recording, the NLMS filter is finding it difficult to perform as well as it did with the simulated data.

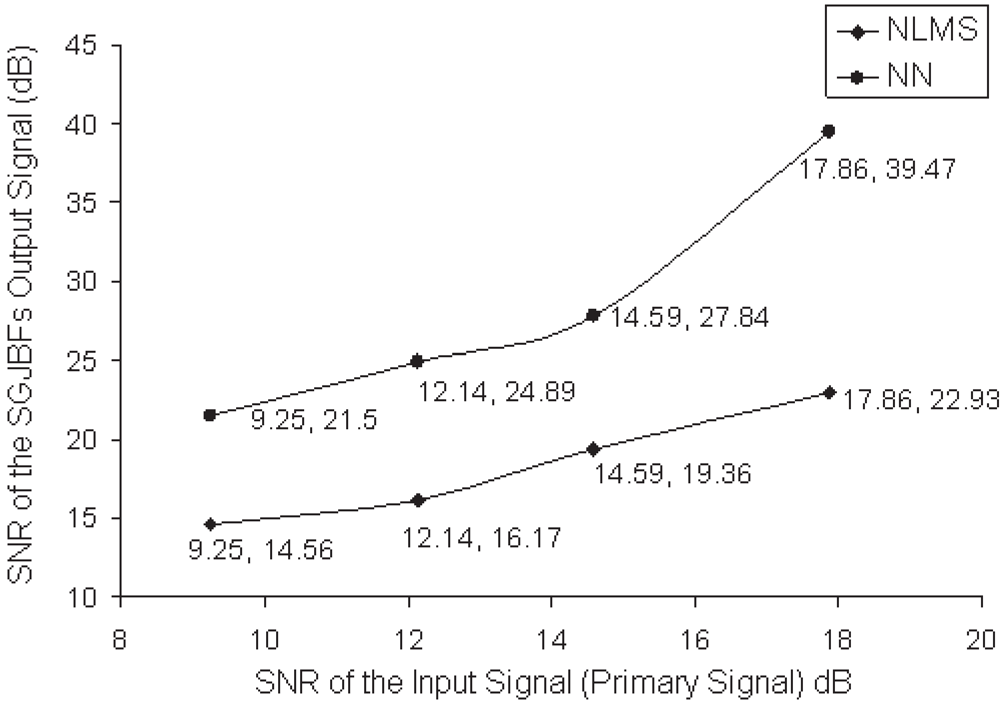


Figure 4.24. Comparison of SNR of the Input vs SNR of the Output Graph

The output signal from the NLMS based beamformer showed only a small improvement in SNR of about 5.5 dB. In most cases, the NN beamformer showed a notable SNR improvement of about 13 dB, and an impressive improvement of 22.6 dB was achieved in one particular case.

The above experiments show the performance of the beamformer with different levels of white noise as the interference signal. In the next experiment, this beamformer algorithm is tested with the babble noise, and then with the factory noise (all these files are from the Noise-X database). The experimental setup is kept the same as above. Table 4.3 shows the comparison of the SNR of the input and the output signal for both NLMS and NN based beamformers, with the babble noise and factory noise signals.

Table 4.3. Comparison of NN and NLMS GJBF with Different Interferences

Type of Interference	Input SNR (dB)		Output SNR (dB)	
	Primary	Reference	NLMS	NN
Babble Noise	19.39	16.71	29.97	34.85
	13.32	10.45	24.23	29.35
Factory Noise	14.06	13.56	23.35	26.66

As seen from the results, the NLMS was able to improve the output by about 10 dB, and NN was able to improve the output by about 15 dB. In general, the NN based beamformer out-performed the traditional NLMS based beamformer by up to 5 dB for different types of interference signals and at different SNRs.

4.4.2 Experiment Using More Than Two Microphones

This section discusses the effect of using more microphones with the proposed beamformer structure. The performance of the dual microphone system decreases as the number of noise sources increases. The theory states that N-1 noise source can be approximated using N number of microphone array systems. Therefore, this problem can be dealt with by adding additional microphones to the proposed system. The complexity and the time it takes to calculate the output will be increased, when adding additional microphones. It will be interesting to see if this additional processing increases the output performance.

Figure 4.25 shows a generic N number of microphone neural SGJBF structure. The first microphone is considered as the primary signal and the rest of the microphones are reference signals. Each microphone signal will contain different levels of speech and noise signal. The speech present in the reference signal is aligned to match the primary signal. All the NLMS based beam-steering filters (NLMS1, NLMS2 and NLMS N) use the primary microphone 1 signal as the reference and align the other microphones' speech signals to this signal. The output from the subtraction section produces (N-1) different noise signals. They are subtracted from the output of the addition section to produce the speech signal. The proposed NN based noise reduction filter is used here to reduce this noise at the microphones. A VAD algorithm is used to control the update of these adaptive filter weights (explained in section 3.2.2).

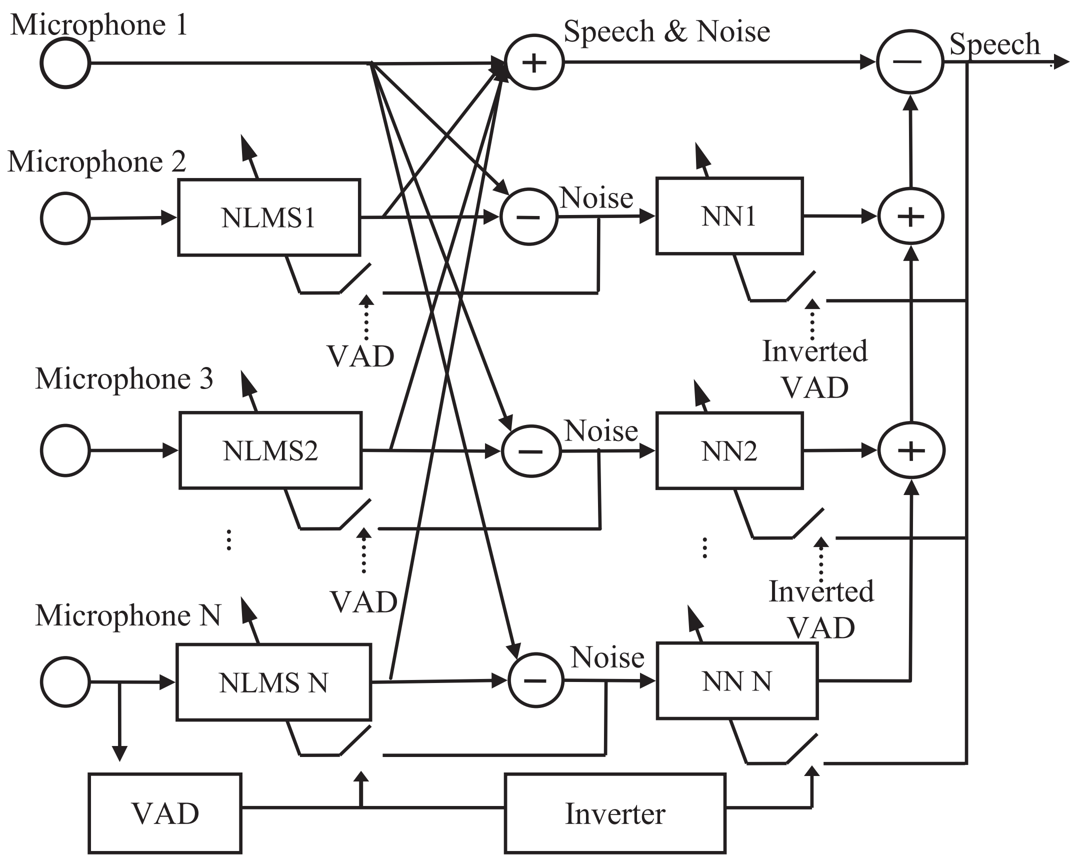


Figure 4.25. N Number of Microphone Neural Switched Griffiths-Jim Beamformer

4.4.2.1. Three-Microphone Array System Experiments

This section discusses the three-microphone array beamformer system. The three-microphone neural switched GJBF and the traditional three-microphone NLMS based switched GJBF have been developed using the Labview programming language G. The NN structure uses a partially connected structure as this system is aimed towards a real-time experiment.

A similar setup to the above experiment is used here. The experiment is set up as follows: the desired speech signal is generated in front of the microphones and the interference signals are generated at an angle to the microphones. The three-microphones are positioned in a linear array approximately 15 cm apart; Figure 4.26 illustrates this arrangement. The interference signals used here are babble noise and factory noise. They are accessed from the Noise-X database. The babble noise is positioned at about a 65 degree angle on the right, and the factory noise is positioned at about a 30 degree angle on the left.

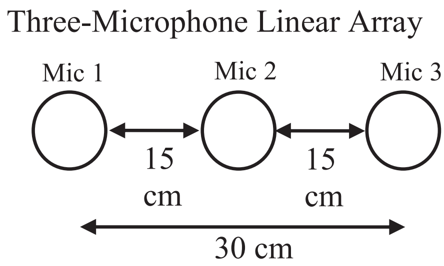


Figure 4.26. Arrangement of the Three-Microphone Array

After carrying out several experiments the parameters from Table 4.4 were set for the NLMS and NN based beamformer. Initially, The NN structure was trained with a small section of the interference signal (3000 samples).

Table 4.4. Variables Used for Three-Microphone Beamformer Experiment

NLMS - SGJBF	Number of weights = 85, $\mu = 0.8$, $\gamma = 0.0001$
NN - SGJBF	20 - Input nodes, 17 - Hidden nodes and 1 - Output node Number of inputs to a neuron was set to 4 Learning rate h1 = 0.5 and h2 = 0.5
VAD	Threshold value =-0.25, Threshold gain = 0.01, Buffer size = 2500

Figure 4.27 (a) shows the input signal from the first microphone; Figure 4.27 (b) and Figure 4.27 (c) show the output signal from the NLMS and the NN based beamformer. The input signal contains the spoken words “one, two, three, four, five” and the interference signals are babble noise and factory noise.

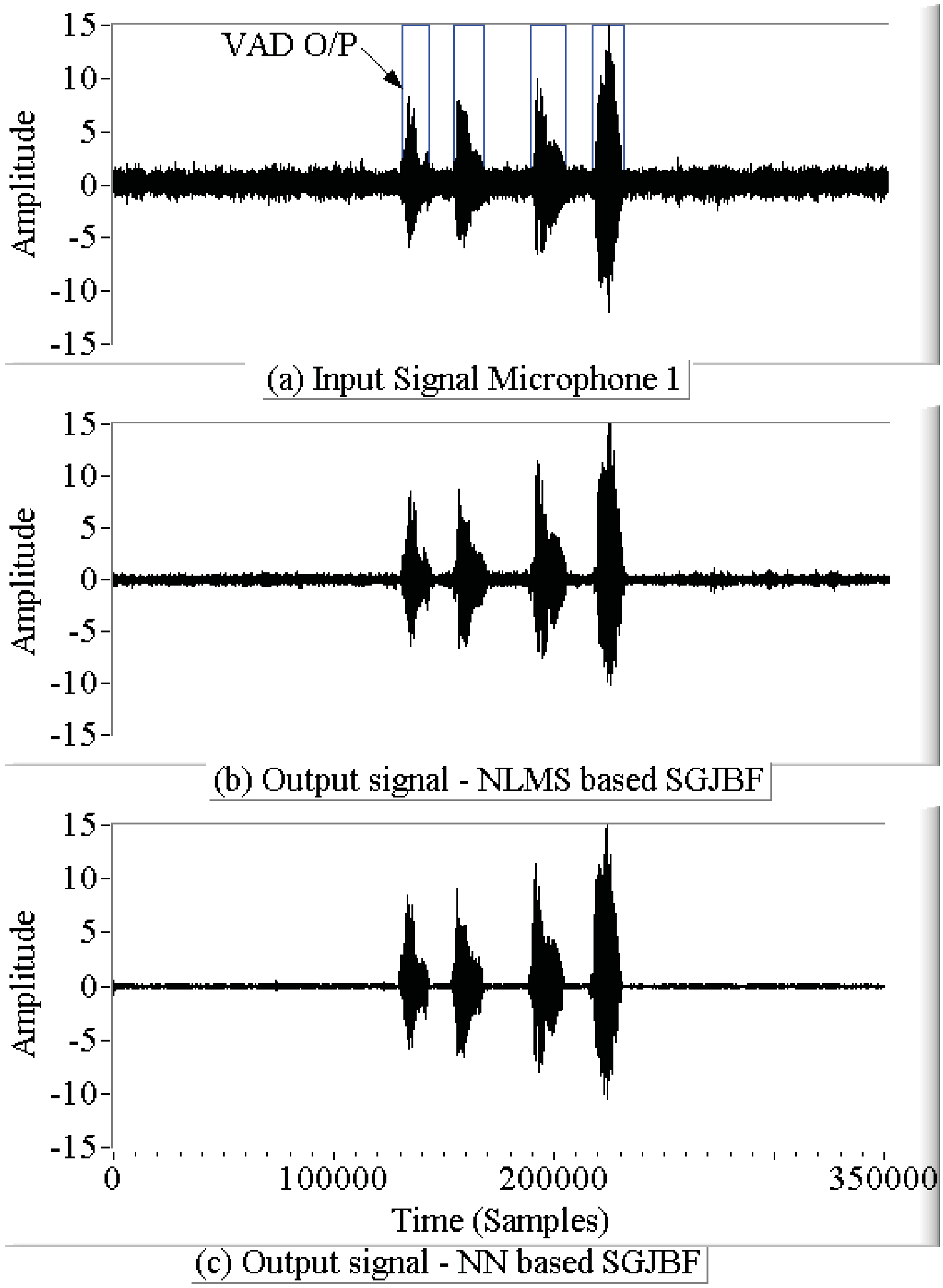


Figure 4.27. (a) Acquired Input Signal (b) Output Signal from the NLMS Based SGJBF, and (c) Output Signal from the Neural SGJBF

Table 4.5 shows the SNR of the input and the output signals shown in Figure 4.27. It shows that the NN based beamformer reduced the noise by about 4.48 dB more than the traditional NLMS based beamformer.

Table 4.5. SNR of the Input Signals and the Output Signal from the Three-Microphone NLMS and NN Based SGJBF

Primary Input Signal	3.51 *dB*
NLMS-SGJBF Output Signal	7.73 *dB*
NN-SGJBF Output Signal	12.21 *dB*

Figure 4.28 shows the comparison of MSE of the output signal of the NLMS and NN based SGJBF.

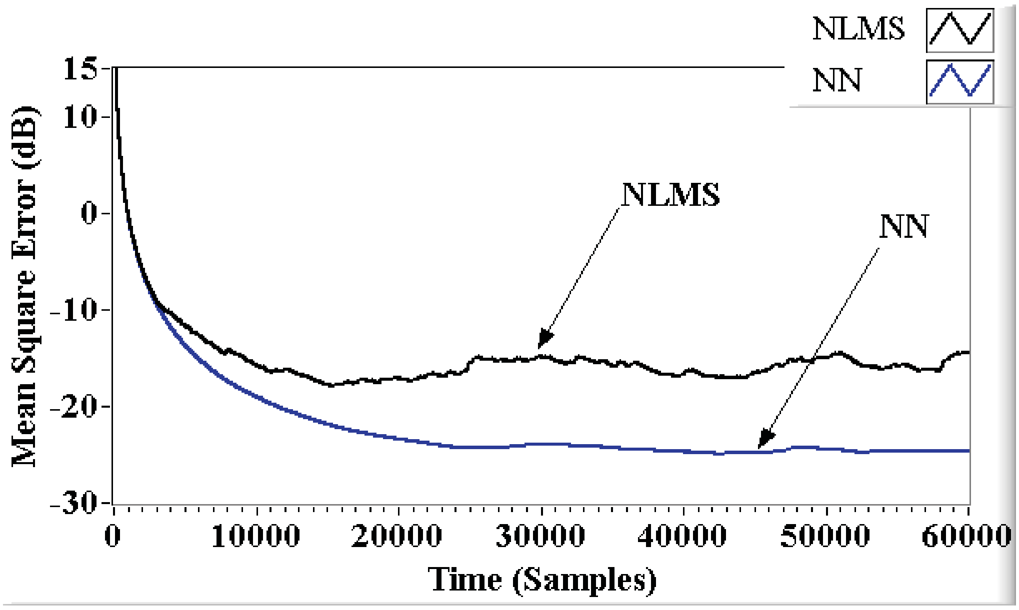


Figure 4.28. Comparison of the MSE for the Three-Channel NLMS and NN Based SGJBF's Output Signals

This graph demonstrates that the NN converged more smoothly than the NLMS. In addition, an informal listening test showed a great improvement in the NN based beamformer output when compared to the NLMS based beamformer.

a) Four-Microphone Array System Experiments

This experimental setup is similar to the above experiment; the target speech signal is generated directly in front of the microphones and the interference signals are generated at an angle to the microphones. Figure 4.29 shows how the four microphones are positioned in a linear array. As illustrated, they are placed approximately 10 cm apart.

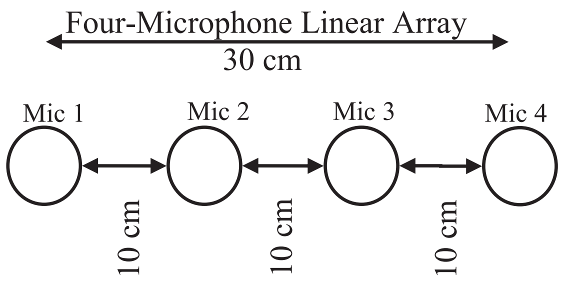


Figure 4.29. Shows the Arrangement of the Linear Four-Microphone Array

The input signal contains the speech signals and the interference signals. The babble noise is positioned at approximately a 35 degree angle on the right and motor noise is positioned at approximately a 60 degree angle on the left. Both noise sources are equally spaced from the microphone. The input signal contains the spoken words "one, two, three, four, five" and the interference signals are a babble noise and a motor noise. All the interference signals used here are taken from the Noise-X database.

The values of the variables are set the same as the ones in the three-microphone experiment (refer to Table 4.4 for more details). The NLMS algorithm's step-size value has to be reduced to 0.2, to get a stable result. Initially, the NN structure was trained with a small section of the interference signal (3000 samples).

Figure 4.30 (a) shows the input signal from the first microphone while Figure 4.30 (b) and Figure 4.30 (c) show the output signal from the NLMS and the NN based beamformer.

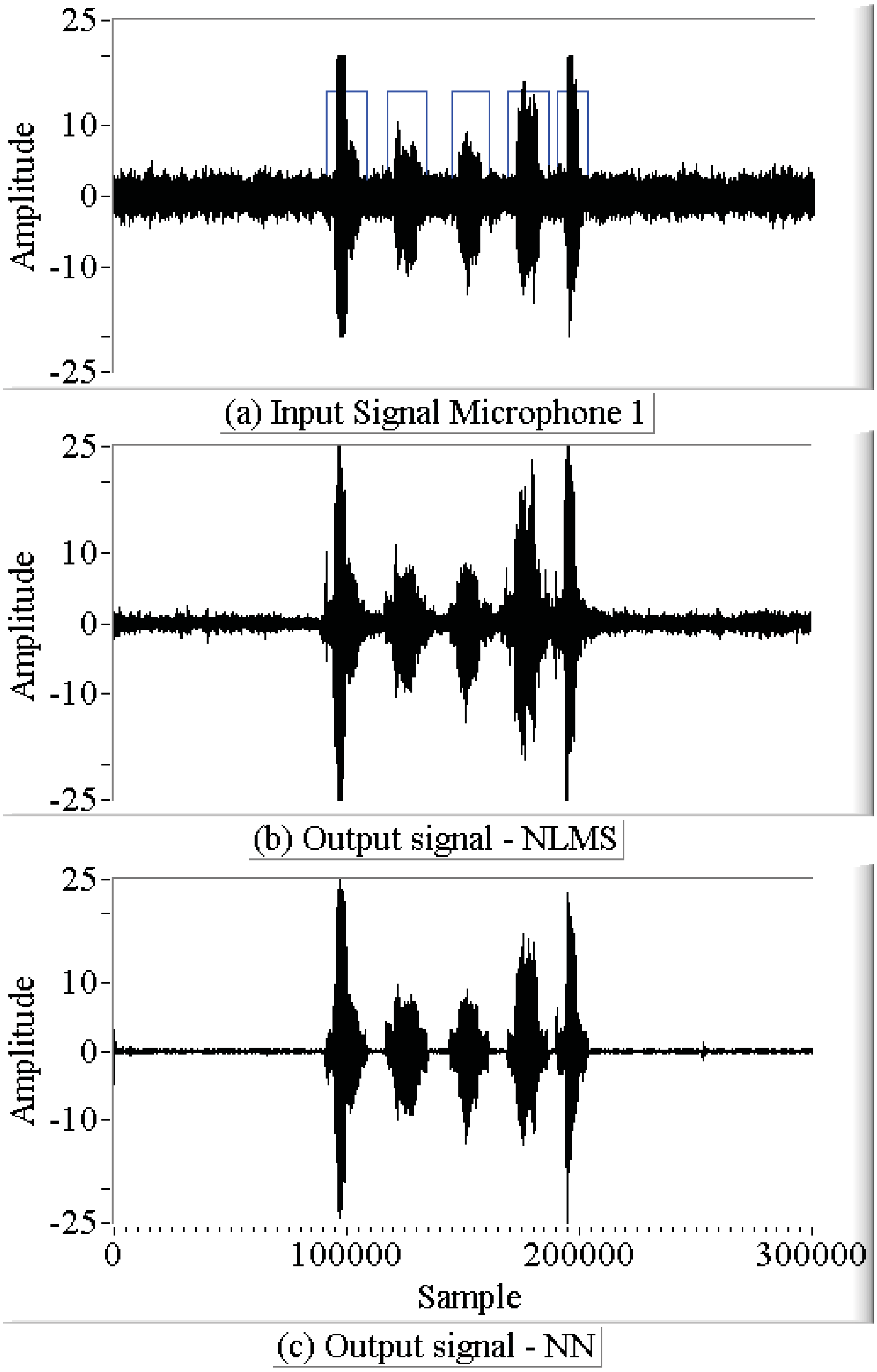


Figure 4.30. (a) Acquired Input Signal and VAD Output (b) Output Signal from the NLMS Based SGJBF (c) Output Signal from the NN Based SGJBF

Table 4.6 shows the SNR of the input and the output signals shown in Figure 4.30. The NLMS based SGJBF showed a SNR improvement of 3.55 dB and the NN based SGJBF showed a SNR improvement of 10.09 dB. The NN based beamformer showed an improvement of 6.54 dB over the NLMS based beamformer.

Table 4.6. SNR of the Input Signals and the Output Signal from the Four-Microphone NLMS and NN Based SGJBF

Primary Input Signal	0.30 *dB*
NLMS-SGJBF Output Signal	3.85 *dB*
NN-SGJBF Output Signal	10.39 *dB*

Figure 4.31 shows the comparison of the MSE of the output signal of the NLMS and NN. This graph demonstrates that the NN converged more smoothly than the NLMS based beamformer.

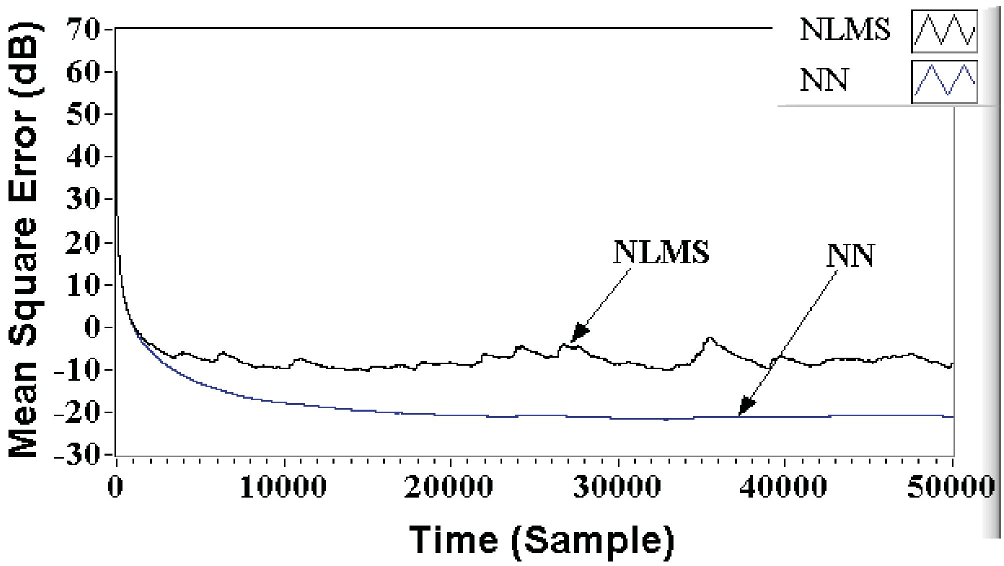


Figure 4.31. Comparison of the MSE for the Four-Channel NLMS and NN Based SGJBF's Output Signals

This chapter introduced the new NN based adaptive filtering for the SGJBF and it showed some promising results when compared to other filters. In the next chapter, the proposed speech enhancement system is compared with the speech-controlled system to evaluate the performance of the system.

CHAPTER 5

EVALUATION OF A SPEECH-CONTROLLED INSTRUMENT WITH THE NEW SPEECH ENHANCEMENT ALGORITHM

This chapter evaluates the performance of the speech enhancement algorithm developed in the earlier chapters with a speech-controlled application. The target speech-based application used here can be any device that can benefit from hands-free control. This chapter will concentrate on a Vibration Monitoring System (VMS) as the target instrumentation, in order to test the beamformer system developed for real-world applications. The first section of this chapter gives a brief introduction to VMS and its use in industry. Furthermore, this chapter will describe the product implementation in the VB.net software. The later section evaluates the performance of the hybrid beamformer algorithm with this target application under different conditions.

5.1 Speech-Controlled Vibration Monitoring System

In industries, the condition of the rotating machine is monitored using the VMS. The machine vibrations are usually measured using an accelerometer, and the results are analysed to predict the performance and reliability of those machines. This system is normally used as part of the industrial maintenance programme to prevent unexpected system breakdowns, by predicting system failures before they occur. By using this system, these industries achieve the following rewards: reduction in unnecessary machine maintenance costs, lower costs due to production loss because the number of unexpected failures is reduced, longer life of machinery because maintenance is carried out regularly (Shikari, 2004).

Traditional VMSs are usually hand-held devices, which often require the user to hold on to the transducer with one hand and control the data acquisition system with the other. However, controlling this system becomes very difficult when both hands are required to perform other tasks. In addition, this device is used in an unfriendly factory environment, so controlling the system without using any hands would be a great help for the user. Currently, there are not many VMSs that have hands-free features. Therefore, this would be an attractive feature in any VMS. The basic idea is not to replace the existing systems with full hands-free control but to improve the system with a hands-free feature, so if the user is not able to use his/her hands at a certain time, then he/she is able to use speech to control the system.

5.2 Proposed Design of the System

The proposed design of the system is shown in Figure 5.1. Arrays of microphones are used to acquire the noisy speech signal. This signal is then transferred to the computer where the chosen speech enhancement algorithm is applied. The improved speech signal is then used to control the speech-controlled VMS.

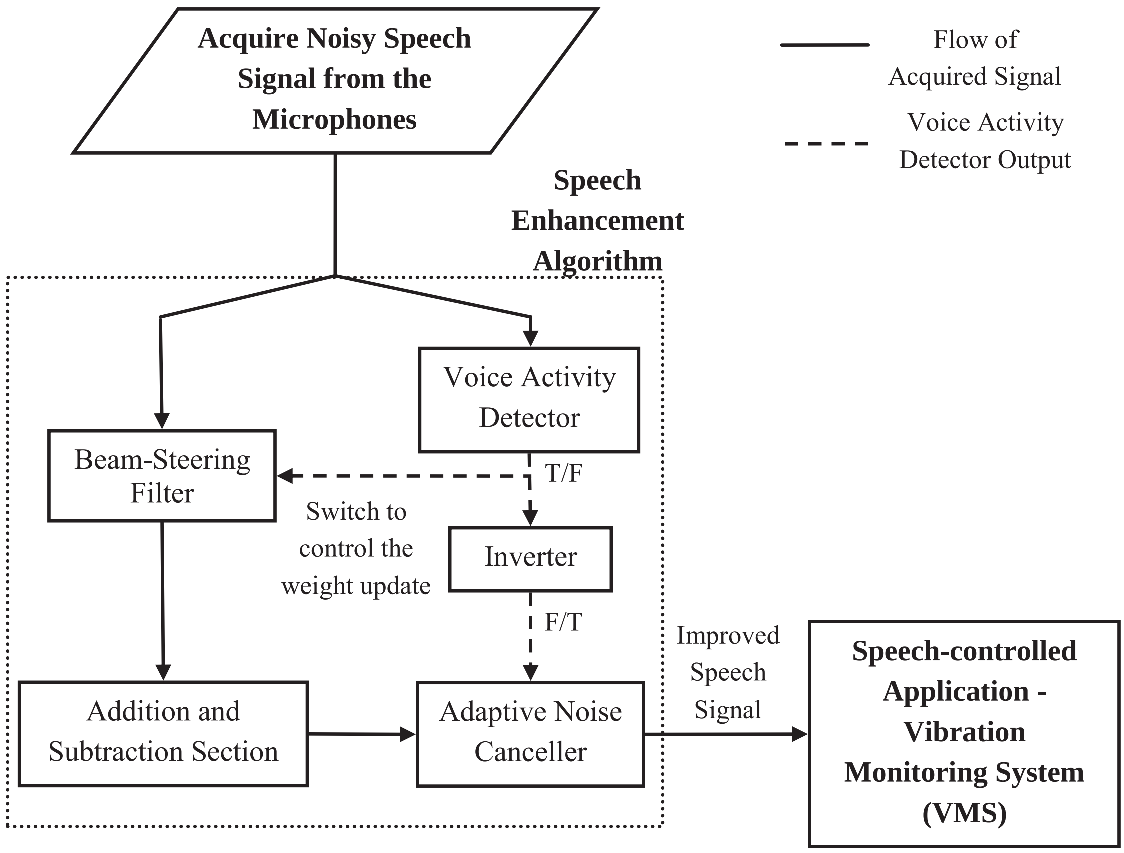


Figure 5.1. Design Layout of the Proposed System

5.2.1 Hardware Configuration

The following hardware equipment's are used here:

- Computer
  – required to run the software and do the processing
  Minimum requirement of Microsoft Windows XP SP3, Intel core2 Duo processor with 2.99 GHz and 2.98 GB of RAM
- Omnidirectional microphones (ECM8000)
  – used to acquire the speech signal
- Microphone Pre-amplifier
  – used to add additional gain to the microphone signals
- Accelerometers (Shinkawa CA-301)
  – used to measure motor's vibration signal
  Specifications:
  - Sensitivity 100 mV/g
  - Resonance frequency 30 kHz
  - Frequency response 2-5,000 Hz
- Data acquisition board (DAQ card - NI USB-9233)
  – Transfers the acquired vibration signal from accelerometer to the computer
  Specifications:
  - 4 Analog input channels
  - 50kS/s max sampling rate per channel
  - IEPE (Integrated Electronic Piezoelectric) signal conditioning for accelerometers.

Figure 5.2 shows the setup of the hardware equipment used in this thesis. The setup can be divided into two main sections: the first section is the speech enhancement system, and the other section is the speech-controlled application. Speech-controlled VMS is the target application here, but this could be replaced with any other application that requires speech enhancement.

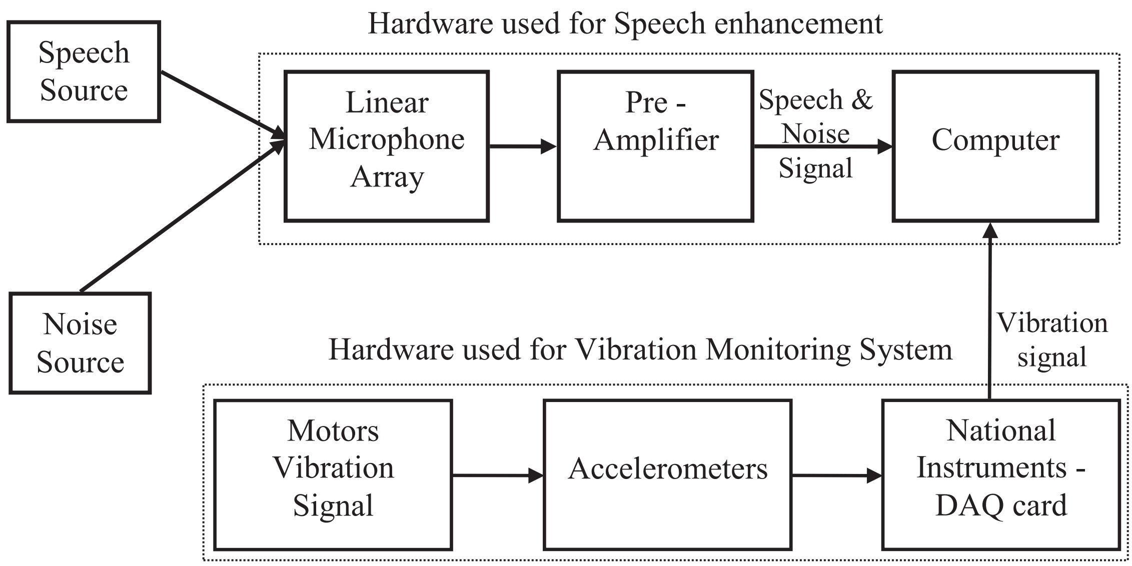


Figure 5.2. Hardware Setup of a Distant Talking Speech-Controlled VMS

Microphones are placed next to each other in a linear array facing the speaker. An external microphone preamplifier is used to add additional gain to the microphone signals. The acquired speech signal is then transferred to the computer via the pre-amplifier (through a line in the input of the sound card) for noise reduction. A neural based beamformer algorithm (explained in Chapter 3) is used here to reduce the background noise. Then the improved speech signal is used to control the speech-controlled VMS. The VMS uses a DAQ card to acquire the motor's vibration signal from an accelerometer. This acquired vibration signal is used in the software program to calculate frequency content and to predict the condition of the machine.

5.2.2 Software Implementation

The main idea is to use speech to control this system, which will allow the users hands-free to do other tasks. In order to understand the speech commands, a speech recognition system is required to convert the speech signals into words. There is a large selection of software development kits (SDK) for developing speech recognition applications that is available in the market. Some examples of speech recognition software include: Dragon Naturally Speaking, Microsoft Speech SDK, Phillips Speech SDK, and Sensory's FluentSoft SDK, to name a few. Speech recognition software that is speaker independent and available at low cost is required here. Since Microsoft Speech SDK does not require any training and is freely available, it is selected here. However, any of this software could be used instead; the choice depends on the personal preference and availability.

Figure 5.3 shows an overview of the software implementation. There are two main parts to this speech-controlled VMS: one is the speech recognition of the spoken command, and the other is the vibration analysis section. The speech enhancement system proposed in Chapter 3 is used as a pre-processing unit to this system. Microsoft Speech SDK toolkit is used with the VB.net software to do the speech recognition. The signal acquisition and analysis section of the VMS is carried out in Labview software. The programs developed in the Labview software are created as Direct Link Library (DLL) files, and they are called from the VB.net software. The acquired noisy speech signal is processed in the speech enhancement system, to enhance the speech signal. This signal is then passed to the speech recognition system, to identify the commands spoken. This command is then used in the VMS to process the required action.

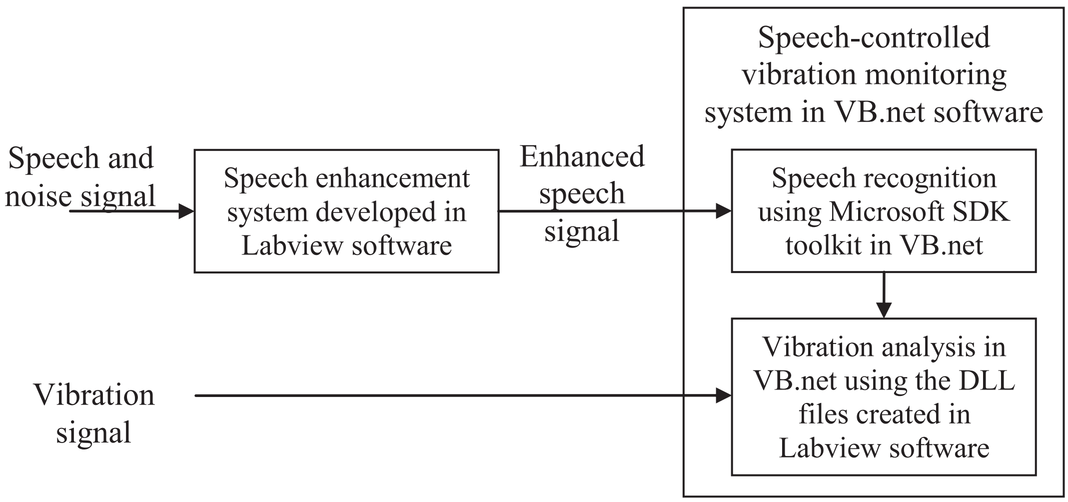


Figure 5.3. Overview of the Software Implementation

The speech-controlled vibration monitoring system was developed using the VB.net software and Figure 5.4 shows the front panel of the developed program.

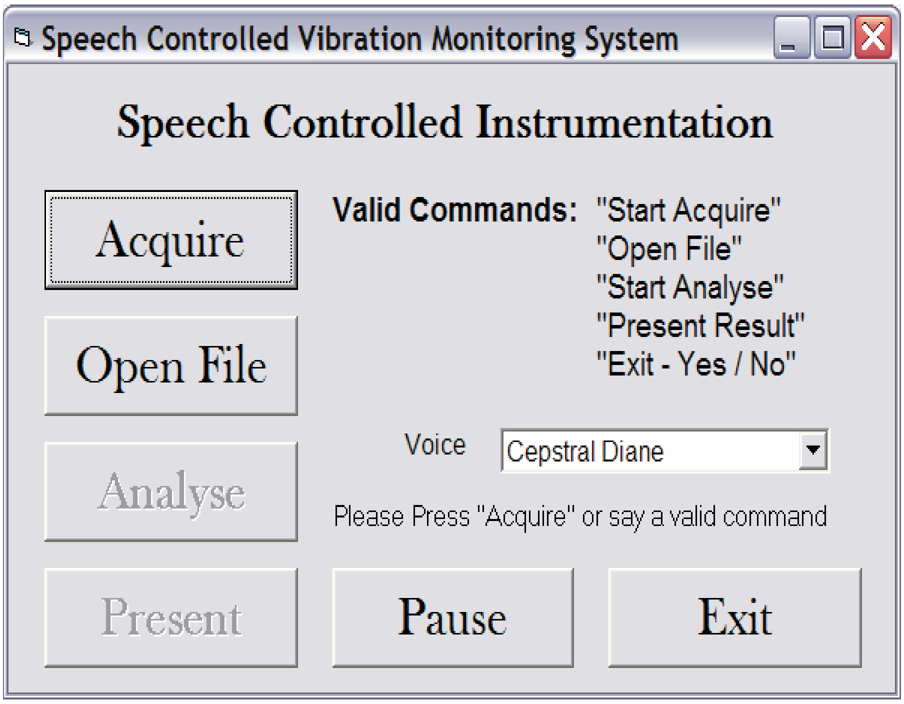


Figure 5.4. Speech-Controlled Vibration Monitoring System

A limited grammar file is used for this software as this will increase the accuracy of the system and allow multiple users to use the system without training. An example of a grammar file used here is shown in Figure 5.5. This system allows the following speech commands to control the system: “start acquire”, “open file”, “start analyse” and “present result”. This program is capable of recognising just a few simple commands as it was only developed to verify the performance of the speech enhancement system. These commands can be easily modified or extended as required by the user of the application.

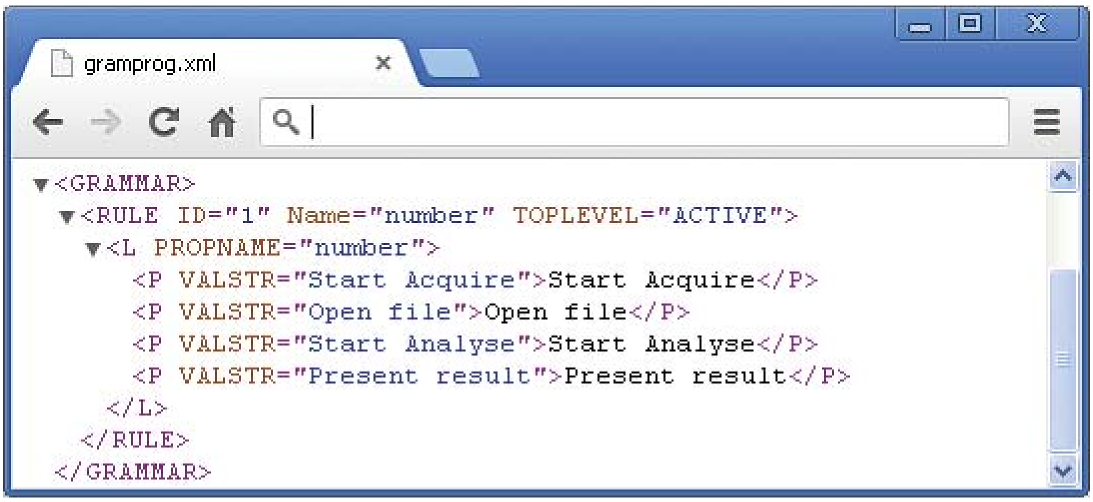


Figure 5.5. Example of a Grammar File used with the Speech Recognition System

In order to start using this system, the user has the option of either using the "start acquire" command or the "open file" command. The "start acquire" command records real-time vibration data to a file and displays it in a separate window. Labview implementation of the acquisition function is shown in Figure 5.6. The DAQmx functions in the Labview software are used to acquire the signal from the DAQ card. This Labview program is created as a DLL and is called in the VB.net software platform.

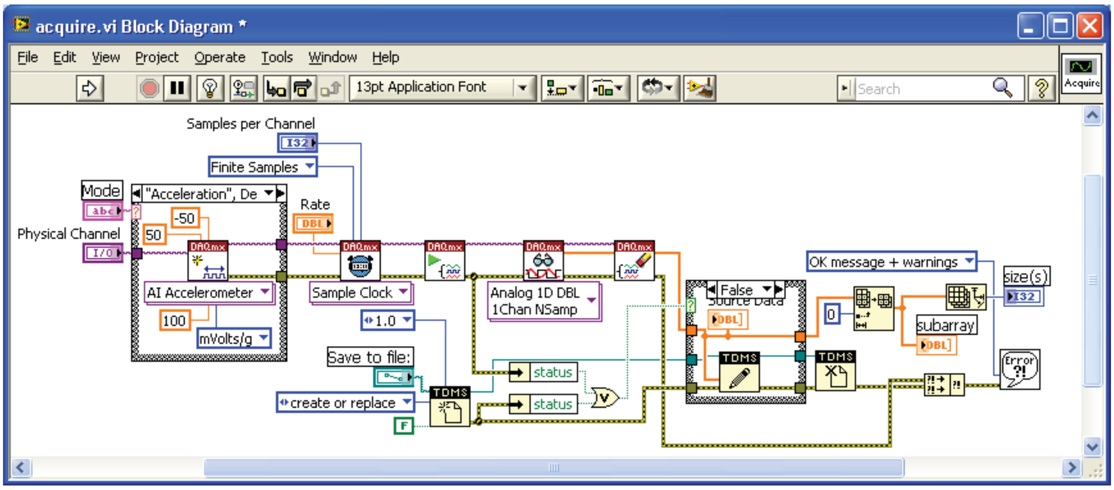


Figure 5.6. Labview Implementation of the Data Acquisition of the Vibration Signal

An example of the "start acquire" command GUI window is shown in Figure 5.7. This window allows the user to choose the number of channels (ai0 to ai3), sample size, sample rate and the mode of measurement. The user has the choice of measuring in "accelerometer" or "voltage". There is an option to save the acquired data in a file, to be analysed at a later time. The "open file" command can be used at a later time, to open an existing file recorded earlier, using this program.

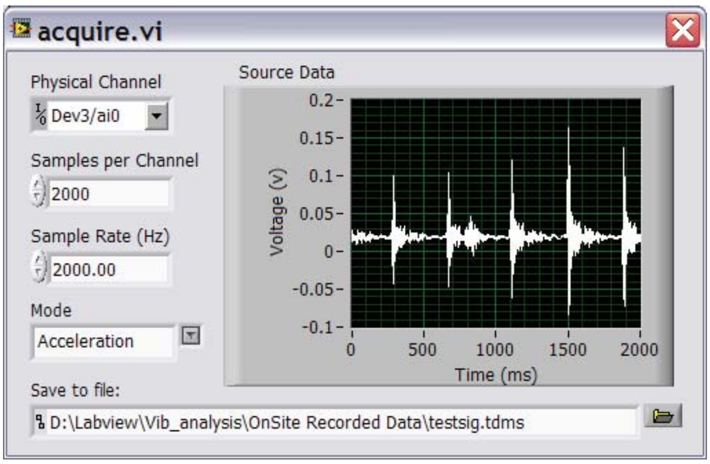


Figure 5.7. *Start Acquire* Command

The "start analyse" command applies a power spectrum to the original data and presents the results in a separate window. An example of the analysis program created in the Labview software is shown in Figure 5.8. This function uses the power spectrum to analyse the signal, but it can be modified with any vibration analysis technique that is required by the user.

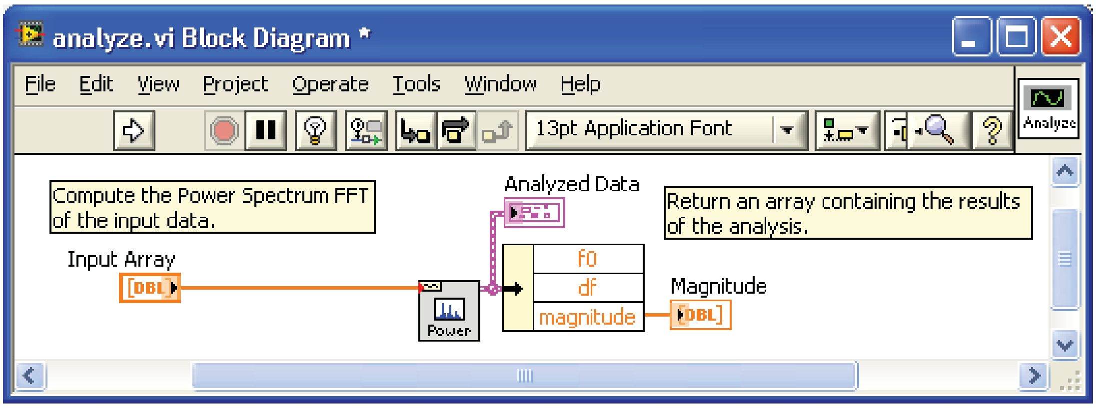


Figure 5.8. Labview Implementation of the Analyse Command

The "present result" command can be used to show the original measured data and analysed data. An example of a GUI window of this command is shown in Figure 5.9.

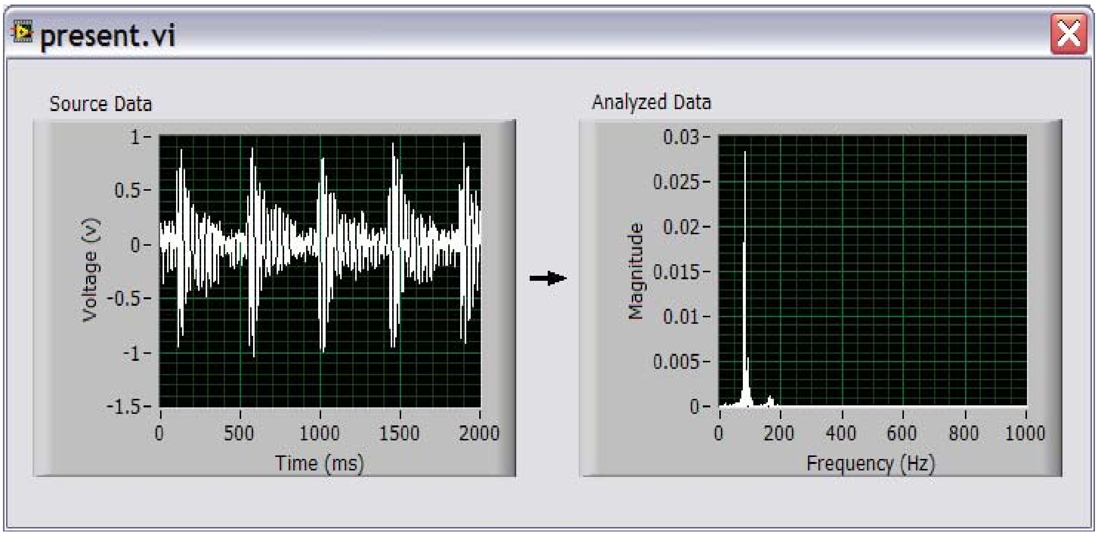


Figure 5.9. *Present Result* Command

The software program developed here shows the feasibility of the speech-controlled idea, but it can be improved later with other features, as necessary for this application.

As the vibration analysis area is not the main focus of this research, the current functions in the system are enough to evaluate the performance of the beamformer algorithm.

5.3 Evaluation of this System in a Noisy Environment

This section will evaluate the performance of the speech enhancement system with the speech-controlled VMS. Several experiments are carried out in real-world surroundings, to test the performance of this speech-controlled VMS under different noisy conditions, and also to test if using the beamformer algorithm helps to improve the performance. This system is tested using two speech enhancement algorithms; they are traditional NLMS based SGJBF and the neural SGJBF algorithm.

A similar setup to that in Figure 4.9 in Chapter 4 is used here. The speech signal is positioned directly in front of the two microphones. The microphones are positioned in a linear array on the table, which will allow the user to move freely and use the system. The background noise is played through a speaker at about a 30 degree angle to the speech signal. This arrangement simulates a factory environment for this experiment. The background noises used to test the system are factory noise and white noise (from the "Noise-X" database).

Several experiments are conducted to test the performance of the speech enhancement system proposed in Chapter 3, with the speech-controlled VMS introduced here. Figure 5.10 shows an example of the input signal and output signal from the Neural SGJBF and the traditional NLMS based SGJBF. The input signal contains the speech command "open File" and the "factory noise" in the background. Figure 5.10 (a) shows the input primary signals and Figure 5.10 (b) shows the input reference signals. The output result from the VAD function is also shown in Figure 5.10 (a). Figure 5.10 (c) shows the output signal from the NLMS based SGJBF and Figure 5.10 (d) shows the output signal from the NN based SGJBF.

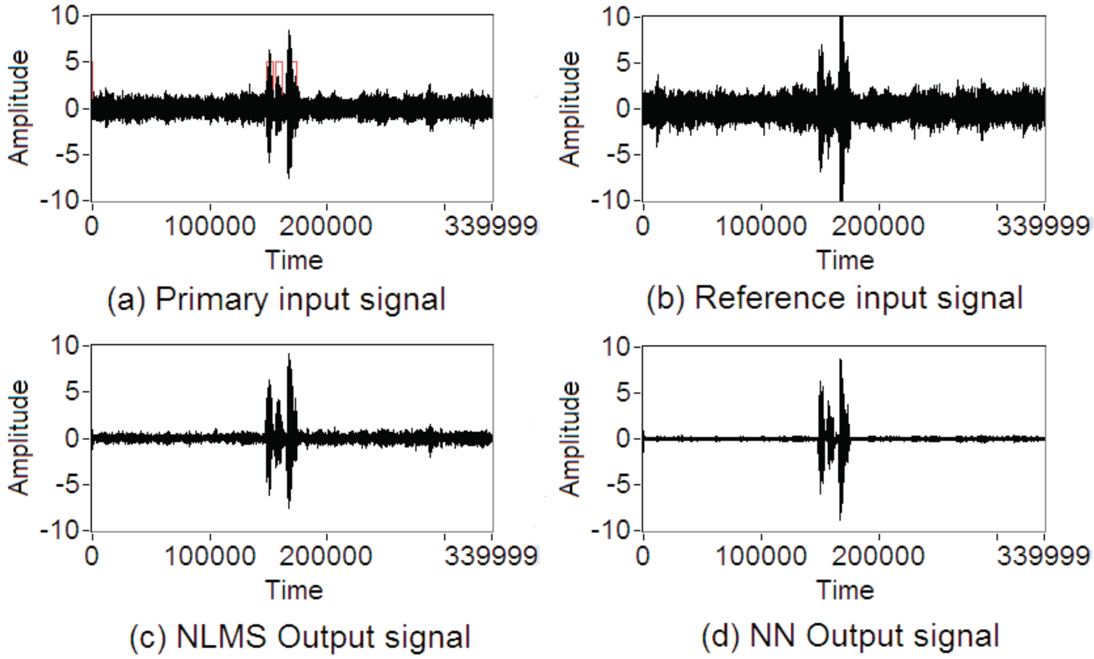


Figure 5.10. Input and Output Signal from the Factory Noise Interference Experiment

Table 5.1 shows the SNR of the input and the output signals shown in Figure 5.10. It shows that the NN based beamformer reduced the noise by about 8.6 dB more than the traditional NLMS based beamformer.

Table 5.1. SNR of the Input Signals and the Output Signal from the NLMS and NN Based SGJBF

	SNR (dB)
Primary Input Signal	10.30
Reference Input Signal	10.40
NLMS-SGJBF Output Signal	19.63
NN-SGJBF Output Signal	28.23

This input and output signal are tested with the speech-controlled VMS to show the performance of the system. The success of the speech recognition system was measured using the Speech Recognition Hit Rate (SRHR) and it is calculated by:

$$\text{SRHR} = \frac{\text{Number of accurately recognised commands}}{\text{Total number of commands given}} \qquad \textbf{(5-1)}$$

The output signals are not tested using a formal listening test, as this is irrelevant to the current study. The main aim is only to improve the performance of the speech recognition system; therefore it is not necessary for the output signal to sound better to the human ear.

Table 5.2 shows the comparison of the input and output signal from the NLMS and NN based SGJBF for different commands under the factory noise signal. It is evident from the results that both beamformers improve the performance of the speech-controlled system. However, the NN based beamformer out-performs the traditional NLMS based beamformer by an average of about 6 dB. Almost perfect recognition is achieved with the NN based beamformer output signal. In some cases, the NLMS beamformer also achieves good recognition rate. Even with a SNR of about 15 dB the speech recognition program achieves a reasonably good performance.

Table 5.2. Evaluation of the Speech-Controlled VMS with the Speech Enhancement System in Factory Noise Environment

	Input SNR (dB)		Output SNR (dB)		Speech Recognition Hit Rate		
					Input Signal	Output Signal	
Factory Noise	**Primary**	**Reference**	**NLMS**	**NN**	**Primary**	**NLMS**	**NN**
Open File	10.3	10.4	19.63	28.23	40%	95%	100%
Present Result	5.55	5.95	14.15	22.84	15%	40%	90%
Start Acquire	6.08	7.15	15.19	19.79	20%	60%	100%
Start Analyse	7.95	8.32	14.96	21.52	35%	90%	100%

Table 5.3 shows the comparison of the input and output signal from the NLMS and NN based SGJBF for different commands under the white noise signal. The performance of the speech-controlled system under white noise is much better than under factory noise.

Table 5.3. Evaluation of the Speech-Controlled VMS with the Speech Enhancement System in White Noise Environment

	Input SNR (dB)		Output SNR (dB)		Speech Recognition Hit Rate		
					Input Signal	Output Signal	
White Noise	**Primary**	**Reference**	**NLMS**	**NN**	**Primary**	**NLMS**	**NN**
Open File	15.7	16.7	26.73	32	80%	100%	100%
Present Result	10.2	11.9	18.16	24.28	10%	30%	70%
Start Acquire	12.5	13.3	20.02	25.51	60%	85%	95%
Start Analyse	14.3	16.3	20.76	25.65	70%	95%	100%

This result shows that when there is non-stationary background noise the NN based beamformer performs much better than the NLMS based beamformer. It also shows that using a speech enhancement algorithm significantly improves the performance of the speech recognition system. Occasionally, there are problems with some commands being recognised incorrectly, which is due to low SNR. The SRHR of the "present result" is bit low compared to the other commands; this was possibly due to the clarity of the voice and incorrect pronunciation of the words. It is found in the experiments that using a predefined grammar file increases the speech recognition hit rate.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The goal of this thesis was to improve the current performance of the speech recognition system in a noisy environment by developing and applying a new neural network (NN) based noise reduction algorithm. Another goal was to apply the new algorithm to an industrial environment and measure the performance.

6.1 Conclusions

A speech enhancement algorithm is an essential component in improving the performance of the speech recognition system in an adverse acoustic environment. This thesis has presented a new multi-microphone speech enhancement system based on NN. A new method for noise reduction based on a non-linear NN based adaptive filter was developed and applied here. The proposed neural adaptive filter algorithm was used as part of the noise cancellation section in the SGJBF structure. This hybrid method of using a neural adaptive filter with this beamformer structure for speech enhancement has not been used before, to the author's knowledge.

The novel neural SGJBF algorithm consisted of three main sections: beam-steering section, noise cancelling section and voice activity detector section. The beam-steering filter used a NLMS algorithm to align the microphone signals. This filter was only updated during the presence of a speech signal. The adaptive noise cancelling filter used a novel non-linear IDNN based adaptive filter algorithm to reduce the background noise. This filter was only updated during the noise-alone signal. The error back-propagation algorithm was used here as the learning algorithm to train the NN filters. As it is a slow gradient algorithm, a momentum term was added with the new weight calculation to improve the performance.

A VAD based on variance of the input signal was used to differentiate between "speech and noise" signal and "noise-alone" signal. An improved version of the automatic variance estimator was introduced in this thesis. It used a moving average technique to avoid the start and end of the signal from being cut off. The output of this VAD function was used to control the updates of the beam-steering and noise cancelling sections. Only one of these sections was updated at any given time. The proposed algorithm was implemented in Labview software.

The neural adaptive filter with a fully connected and partially connected three-layer IDNN was developed here for noise reduction. Several other adaptive filters such as LMS, NLMS, APA and Volterra Filters were implemented in the Labview software, in order to compare the neural adaptive filter with these methods. The rate of convergence of these methods was analysed with different simulation setups. The LMS algorithm struggled to function efficiently under a high eigenvalue spread of the input signal.

A comparison of the NLMS based SGJBF and Neural SGJBF was carried out with real-world recordings. A partially connected network structure was chosen for rapid real-time processing. The proposed neural SGJBF algorithm showed promising noise reduction improvement over the previous NLMS based SGJBF structure. However, the performance of the NN greatly depends on several variables, such as the amount of training given and the size of the NN structure. Improving the dual microphone structure to three and four microphones gave the option of reducing more noise sources. The results showed that the number of noise sources that are being minimised has to be less than the number of microphones. As a result, a much higher noise reduction was achieved using more than two microphones.

This thesis also introduced a computer based speech-controlled vibration monitoring system. The main focus was to allow the users to keep their hands free to complete other tasks while using the system for measurement. This system was developed using the VB.net software. The data acquisition of the vibration signals and the processing was done using Labview software. The speech recognition feature was implemented using the Microsoft Speech SDK toolkit. The main commands used to control the system were: “start acquire”, “open file”, “start analyse” and “present results”. Currently, there is no VMS that has a hands-free feature. Therefore, this will be a great advantage to this type of instrumentation.

The output signals from the beamformer were tested with the speech-controlled system to analyse their performance. It was proven that using the NN based beamformer with the speech recognition system notably improved the recognition rate of the system. This shows that the user of this system can benefit greatly by using the speech enhancement system, with the speech-controlled system. The speech enhancement system allows the user to operate the system in a noisy environment, while keeping their hands free to do other tasks.

6.2 Suggestions for Future Work

The thesis investigates the application of IDNN architecture to the adaptive noise cancelling structure. It concentrates on the single hidden layer NN structure, as it was enough to approximate our target noisy environment. However, future work can expand this study by adding another hidden layer to the current system and analysing the performance of the system. It will be interesting to see if an extra hidden layer will improve the noise reduction performance and whether it is worth the extra computational cost. The NN filter uses the error back-propagation algorithm to train the structure. This algorithm is considered as a gradient based algorithm, and it has a slower convergence rate. If this is a problem for real-time processing, a faster training algorithm can be used to improve the learning rate. A more sophisticated method like the fast back-propagation algorithms could be used to improve the current structure.

Currently developed speech-controlled VMS only has a few commands to control the system. This system was developed as a prototype to show the feasibility of the study. However, it can be extended to include more functions that are useful in condition monitoring; for example a balancing calculator can also be included with this system. The speech-controlled VMS was mainly tested with a single speaker, with different noisy backgrounds. The results confirm the possibility of using this speech-controlled system in a simulated real-world environment. However, there was not enough time to test this system in a real-world industrial environment. Therefore a more thorough study, using this system in industry, will provide a more significant contribution to this area.

References

Abbas, Q., Ahmad, J., & Bangyal, W. H. (2010, 18-19 Oct.). *Momentum term heals the performance of Back Propagation Algorithm for digit recognition.* Paper presented at the 2010 6th International Conference on Emerging Technologies (ICET), Islamabad.

Aboulnasr, T., & Mayyas, K. (1997). A robust variable step-size LMS-type algorithm: analysis and simulations. *IEEE Transactions on Signal Processing, 45*(3), 631-639. doi: 10.1109/78.558478

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.

Al-Anbaky, T. M. J. (2004). Adaptive Noise Cancellation Based on Multilayer Perceptron Neural Network Approach. *Iraqi Journal of Computers, Communication, Control, and Systems Engineering, 4*(2), 37-46.

Anderson, B., & Montgomery, D. (1990, 17-21 June). *A method for noise filtering with feed-forward neural networks: Analysis and comparison with low-pass and optimal filtering.* Paper presented at the International Joint Conference on Neural Networks, San Diego, CA, USA.

Anusuya, M. A., & Katti, S. K. (2009). Speech Recognition by Machine: A Review. *International Journal of Computer Science and Information Security, 6*(3), 181-205.

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.

Babu, C. G., Vanathi, P. T., Ramachandran, R., Rajaa, M. S., & Vengatesh, R. (2010). Performance analysis of voice activity detection algorithm for robust speech recognition system under different noisy environment. *Journal of Scientific & Industrial Research, 69*(7), 512-522.

Badri, L. (2010). Developement of Neural Networks for Noise Reduction. *The International Arab Journal of Information Technology, 7*(3), 289-294.

Benesty, J., Makino, S., & Chen, J. (2005). *Speech Enhancement*. Netherlands: Springer.

Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., . . . Wellekens, C. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication, 49*(10-11), 763-786. doi: http://dx.doi.org/10.1016/j.specom.2007.02.006

Berkens, M. (2013, 22 May). Google Chrome Now Allows For Voice Search On Laptops & Desktops. Retrieved 18 June 2013, from http://www.thedomains.com/2013/05/22/google-chrome-now-allows-for-voice-search-on-laptops-desktops/

Bilcu, R. C. (2004). *On Adaptive Least Mean Square FIR Filters: New Implementations and Applications.* (Doctor of Technology), Tampere University of Technology, Tampere, Finland. Retrieved from http://dspace.cc.tut.fi/dpub/handle/123456789/85

Bitzer, J., Simmer, K. U., & Kammeyer, K.-d. (1999, 25 - 26 May). *Multi-Microphone Noise Reduction Techniques For Hands-Free Speech Recognition*

- *A Comparative Study*. Paper presented at the Robust Methods for Speech Recognition in Adverse Conditions (ROBUST 99), Tampere, Finland.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing, 27*(2), 113-120. doi: 10.1109/tassp.1979.1163209

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

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

Chan, K. Y., Yong, P. C., Nordholm, S., Yiu, C. K. F., & Lam, H. K. (2014). A hybrid noise suppression filter for accuracy enhancement of commercial speech recognizers in varying noisy conditions. *Applied Soft Computing, 14, Part A*(0), 132-139. doi: http://dx.doi.org/10.1016/j.asoc.2013.05.017

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.

Chen, C. K., & Tzi-Dar, C. (1996, 12-15 May). *Multilayer perceptron neural networks for active noise cancellation*. Paper presented at the IEEE International Symposium on Circuits and Systems, Connecting the World., Atlanta, GA.

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

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.

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

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

Cox, K. S. (1988). *An analysis of noise reduction using back-propagation neural networks*. (Master of Science in Computer Engineering), Air Force Institute of Technology Air University, Ohio, USA.

Davis, G. M. (2002). *Noise reduction in speech applications*: CRC Press.

Davis, K. H., Biddulph, R., & Balashek, S. (1952). Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America, 24*(6), 637-642.

Daw, D. (2011). What Makes Siri Special? *PC World*. http://www.pcworld.com/article/242479/what_makes_siri_special_.html

Diniz, P. S. R. (2008). Introduction To Adaptive Filtering *Adaptive Filtering Algorithms and Practical Implementation* (pp. 1-12): Springer US.

Diniz, P. S. R. (2013). *Adaptive filtering: algorithms and practical implementation*: Springer.

Dobler, S. (2000). Speech recognition technology for mobile phones. *Ericsson Review, 77*(3), 148-155.

Du, K. L., Lai, A. K. Y., Cheng, K. K. M., & Swamy, M. N. S. (2002). Neural methods for antenna array signal processing: a review. *Signal Processing, 82*(4), 547-561.

Edwards, J. (2013). Researchers Push Speech Recognition Toward the Mainstream. *IEEE Signal Processing Magazine, 30*(1), 8-11. doi: 10.1109/msp.2012.2219682

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

Evans, J. B., Ping, X., & Liu, B. (1993). Analysis and implementation of variable step size adaptive algorithms. *IEEE Transactions on Signal Processing, 41*(8), 2517-2535. doi: 10.1109/78.229885

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 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97.

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

Fink, N., Furst, M., & Muchnik, C. (2008, 3-5 Dec.). *The benefit of speech enhancement to the hearing impaired.* Paper presented at the 25th IEEE Convention of Electrical and Electronics Engineers in Israel, Eilat.

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, 23-26 May). *The voice activity detector for the PAN-European digital cellular mobile telephone service.* Paper presented at the International Conference on Acoustic Speech Signal Processing, Glasgow.

Frost, O. L., III. (1972). An algorithm for linearly constrained adaptive array processing. *IEEE Proceedings, 60*(8), 926-935.

Fukane, A. R., & Sahare, S. L. (2011). Different Approaches of Spectral Subtraction method for Enhancing the Speech Signal in Noisy Environments. *International Journal of Scientific & Engineering Research, 2*(5).

Gan, K. B., Zahedi, E., Alauddin, M., & Ali, M. (2011). Application of Adaptive Noise Cancellation in Transabdominal Fetal Heart Rate Detection Using Photoplethysmography. In L. Garcia (Ed.), *Adaptive Filtering Applications*: InTech.

Ganapathiraju, A., Webster, L., Trimble, J., Bush, K., & Kornman, P. (1996, 11-14 Apr 1996). *Comparison of energy-based endpoint detectors for speech signal processing.* Paper presented at the Conference on Bringing Together Education, Science and Technology. IEEE Southeastcon '96, Tampa, FL, USA.

Gaonkar, A., & Savan, M. (2012). Concepts of neural network and its application. *International Journal of Electrical and Electronics Engineers, 2*(2).

Gay, S. L., & Tavathia, S. (1995, 9-12 May). *The fast affine projection algorithm.* Paper presented at the International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95, Detroit, MI.

Gersho, A., & Paksoy, E. (1992, 25-26 Jun). *An overview of variable rate speech coding for cellular networks.* Paper presented at the IEEE International Conference on Selected Topics in Wireless Communications, 1992., Vancouver, BC.

Gong, Y. (1995). Speech recognition in noisy environments: A survey. *Speech Communication, 16*, 261-291.

Griffiths, L., & Jim, C. (1982). An alternative approach to linearly constrained adaptive beamforming. *IEEE Transactions on Antennas and Propagation, 30*(1), 27-34.

Hadei, S. A., & Azmi, P. (2010, May). *A Novel Adaptive Channel Equalization Method Using Variable Step-Size Partial Rank Algorithm.* Paper presented at the 2010 Sixth Advanced International Conference on Telecommunications, AICT' 10, Barcelona, Spain.

Haigh, J. A., & Mason, J. S. (1993, 19-21 Oct). *Robust voice activity detection using cepstral features.* Paper presented at the TENCON '93. IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering.1993., Beijing.

Hamid, N. A., Nawi, N. M., Ghazali, R., & Salleh, M. N. M. (2011). Improvements of Back Propagation Algorithm Performance by Adaptively Changing Gain, Momentum and Learning Rate. *International Journal on New Computer Architectures and Their Applications (IJNCAA), 1*(4), 866-878.

Han, J., Yook, S., Nam, K. W., Lee, S., Kim, D., Hong, S. H., . . . Kim, I. Y. (2012). Comparative evaluation of voice activity detectors in single microphone noise reduction algorithms. *Biomedical Engineering Letters, 2*(4), 255-264.

Hansen, J. H. L., Kim, W., & Angkititrakul, P. (2008, 06-08 May). *Advances in human-machine systems for in-vehicle environments.* Paper presented at the Workshop on Hands-Free Speech Communication and Microphone Arrays Trento, Italy.

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

Haykin, S. (Ed.). (2000). *Unsupervised Adaptive Filtering* (4 ed. Vol. 1: Blind Source Separation). USA: John Wiley & Sons, Inc.

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

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

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

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.

Hoyt, J. D., & Wechsler, H. (1990, 17-21 June). *An examination of the application of multi-layer neural networks to audio signal processing.* Paper presented at the International Joint Conference on Neural Networks, San Diego, CA, USA.

Hu, Y., & Loizou, P. C. (2007). A comparative intelligibility study of single-microphone noise reduction algorithms. *Journal of the Acoustical Society of America, 122*(3), 1777-1786.

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.

Jeub, M., Kolossa, D., Astrudillo, R. F., & Orglmeister, R. (2009, March). *Performance Analysis of Wavelet-based Voice Activity Detection.* Paper presented at the NAG/DAGA 2009, Rotterdam.

Jiju, P. V., Singh, C. P., & Sharma, R. M. (2012). Characterization of Noise Associated with Forensic Speech Samples. In A. Neustein, & H. A. Oatil (Eds.), *Forensic Speaker Recognition*. New York: Springer Science+Business Media: Law Enforcement and Counter-Terrorism.

Jingdong, C., Benesty, J., Yiteng, H., & Doclo, S. (2006). New insights into the noise reduction Wiener filter. *IEEE Transactions on Audio, Speech, and Language Processing 14*(4), 1218-1234. doi: 10.1109/tsa.2005.860851

Jongseo, S., Nam Soo, K., & Wonyong, S. (1999). A statistical model-based voice activity detection. *IEEE Signal Processing Letters, 6*(1), 1-3. doi: 10.1109/97.736233

Jr, J. P. M. (2011). Apple's Siri has Rival in Sensory. *PC World*. http://www.pcworld.com/article/241562/apples_siri_has_rival_in_sensory.html

Juang, B. H., & Tsuhan, C. (1998). The past, present, and future of speech processing. *IEEE Signal Processing Magazine, 15*(3), 24-48. doi: 10.1109/79.671130

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.

Kamm, C., Walker, M., & Rabiner, L. (1997). The role of speech processing in human – computer intelligent communication. *Speech Communication, 23*, 263-278.

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

Kaur, R., & Kaur, S. (2011). Neural Network Approach for Adaptive Noise Cancellation. *International Journal of Information Technology and Knowledge Management, 4*(2), 455-457.

Kavanagh, S. (2012, 5-10 May). *Facilitating Natural User Interfaces through Freehand Gesture Recognition.* Paper presented at the Conference on Human Factors in Computing Systems, CHI12., Austin, Texas, USA.

Kim, S. K., & Chang, J. H. (2012). Voice activity detection based on conditional MAP criterion incorporating the spectral gradient. *Signal Processing, 92*(7), 1699-1705. doi: http://dx.doi.org/10.1016/j.sigpro.2012.01.005

Knapp, C., & Carter, G. (1976). The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing, 24*(4), 320-327. doi: 10.1109/tassp.1976.1162830

Knecht, W. G. (1994). Nonlinear noise filtering and beamforming using the perceptron and its Volterra approximation. *IEEE Transactions on Speech and Audio Processing, 2*(1), 55-62.

Knecht, W. G., Schenkel, M. E., & Moschytz, G. S. (1995). Neural network filters for speech enhancement. *IEEE Transactions on Speech and Audio Processing, 3*(6), 433-438.

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.

Kwong, R. H., & Johnston, E. W. (1992). A variable step size LMS algorithm. *IEEE Transactions on Signal Processing, 40*(7), 1633-1642. doi: 10.1109/78.143435

Landau, L. J., & Taylor, J. G. (Eds.). (1998). *Concepts for neural networks : a survey*. New York: Springer-Verlag

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

Lee, J.-W., & Lee, G.-K. (2005). Design of an Adaptive Filter with a Dynamic Structure for ECG Signal Processing. *International Journal of Control, Automation, and Systems, 3*(1), 137-142.

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

Liew Ban, F., Hussain, A., & Samad, S. A. (2000, 24-27 Sep). *Speech enhancement by noise cancellation using neural network.* Paper presented at the Proceedings of TENCON 2000, Kuala Lumpur.

Maganti, H. K., Motlicek, P., & Gatica-Perez, D. (2007, 15-20 April). *Unsupervised Speech/Non-Speech Detection for Automatic Speech Recognition in Meeting Rooms.* Paper presented at the IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007, Honolulu, HI.

Makino, S., Lee, T.-W., & Sawada, H. (Eds.). (2010). *Blind Speech Separation*: Springer.

Malik, G., & Sappal, A. S. (2011). Adaptive Equalization Algorithms:An Overview. *(IJACSA) International Journal of Advanced Computer Science and Applications, 2*(3).

Martinez, J. A., Ubeda, A., Ianez, E., Azorin, J. M., & Perez-Vidal, C. (2013). Multimodal System Based on Electrooculography and Voice Recognition to Control a Robot Arm. *International Journal of Advanced Robotic Systems, 10*, 283-283.

Mathews, V. J. (1991). Adaptive polynomial filters. *IEEE Signal Processing Magazine, 8*(3), 10-26.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the idea immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115-133.

Mistry, D., & Kulkarni, A. V. (2013). Noise Cancellation using Adaptive Filter Base On Neural Networks. *ITSI Transactions on Electrical and Electronics Engineering (ITSI-TEEE), 3*, 1.

Moir, T. J. (2001a). Automatic variance control and variance estimation loops. *Circuits Systems and Signal Processing, 20*, 1-10.

Moir, T. J. (2001b). Cancellation of noise from speech using Kepstrum analysis. *Research Letters in the Information and Mathematical Sciences, 2*, 101-111.

Moir, T. J. (2012). Filtering, smoothing and prediction using a control-loop spectral factorization method for coloured noise. *International Journal of Adaptive Control and Signal Processing, 27*(3), 153-165. doi: 10.1002/acs.2285

Montazeri, M., & Duhamel, P. (1995). A set of algorithms linking NLMS and block RLS algorithms. *IEEE Transactions on Signal Processing, 43*(2), 444-453.

Murugesan, M., & Sukanesh, R. (2009, 28-29 Dec). *Automated Detection of Brain Tumor in EEG Signals Using Artificial Neural Networks.* Paper presented at the International Conference on Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT '09., Trivandrum, Kerala, India.

Nagi, J., Ahmed, S. K., & Nagi, F. (2008, March 7-9). *A MATLAB based Face Recognition System using Image Processing and Neural Networks.* Paper presented at the 4th International Colloquium on Signal processing and its Applications, Kuala Lumpur, Malaysia.

NOISEX-92. (1996). from DRA Speech Research Unit http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html

Noyes, J., Haigh, R., & Starr, A. (1989). Automatic speech recognition for disabled people. *Applied Ergonomics, 20*(4), 293-298.

Omologo, M., & Svaizer, P. (1994). *Acoustic event localization using a cross-power spectrum phase based technique.* Paper presented at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP - 1994), Adelaide, Australia.

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

Ozeki, K., & Umeda, T. (1984). An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electronics & Communications in Japan, 67*(5), 19-27.

Papp, I. I., Saric, Z. M., & Teslic, N. D. (2011). Hands-free Voice Communication with TV. *IEEE Transactions of consumer electronics, 57*(2), 606-614.

Portet, F., Vacher, M., Golanski, C., Roux, C., & Meillon, B. (2013). Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects *Personal and Ubiquitous Computing* (Vol. 17, pp. 127-144).

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

Qi, T. Z. (2008, 2-4 Dec.). *Automotive Speech Control in a Non-Stationary Noisy Environment.* Paper presented at the 15th International Conference on Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. , Auckland.

Qidwai, U., & Shakir, M. (2012). *Ubiquitous Arabic voice control device to assist people with disabilities.* Paper presented at the 4th International Conference on Intelligent and Advanced Systems New York.

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., & Sambur, M. (1977, May 1977). *Voiced-unvoiced-silence detection using the Itakura LPC distance measure.* Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '77., Hartford, CT, USA.

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

Ramírez, J., & Górriz, J. M. (Eds.). (2011). *Recent Advances in Robust Speech Recognition Technology*: Bentham e-Books.

Ramirez, J., Górriz, J. M., & Segura, J. C. (2007). *Voice Activity Detection. Fundamentals and Speech Recognition System Robustness* M. Grimm, & K. Kroschel (Eds.), *Robust Speech Recognition and Understanding* Retrieved from http://www.intechopen.com/books/robust_speech_recognition_and_understanding/voice_activity_detection__fundamentals_and_speech_recognition_system_robustness

Rao, S. S., & Pisharam, P. M. (1990, 1-3 May). *A noise-reduction neural network as a preprocessing stage in the SVD based method of harmonic retrieval.* Paper presented at the IEEE International Symposium on Circuits and Systems, 1990., New Orleans, LA, USA

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

Reed, M. J., & Hawksford, M. O. (2000). Efficient implementation of the Volterra filter. *IEE Proceedings of Vision, Image and Signal Processing, 147*(2), 109-114.

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.

Revathy, N., & Guhan, T. (2012). Face recognition system using back propagation artifical neural networks. *International Journal of Advanced Engineering Technology, III*(I), 321-324.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1. Foundations.). Cambridge, MA, USA: MIT Press.

Ryan, J. G., & Goubran, R. A. (2003). Application of near-field optimum microphone arrays to hands-free mobile telephony. *IEEE Transactions on Vehicular Technology, 52*(2), 390-400.

Sangwan, A., Chiranth, M. C., Jamadagni, H. S., Sah, R., Prasad, R. V., & Gaurav, V. (2002, 3-5 July). *VAD techniques for real-time speech transmission on the Internet.* Paper presented at the International Conference on High-Speech Networks and Multimedia Communication, Jeju Island, South Korea.

Sarkar, D. (1995). Methods to speed up error back-propagation learning algorithm. *ACM Computing Surveys, 27*(4), 519-544. doi: http://doi.acm.org/10.1145/234782.234785

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

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 International Conference on Spoken Language Processing, ICSLP-98, Sydney, Australia.

Shikari, B. (2004). Automation in condition based maintenance using vibration analysis. *Maintenance world*.

Sinha, N. K., Gupta, M. M., & Rao, D. H. (2000, 19-22 Jan). *Dynamic neural networks: an overview.* Paper presented at the Proceedings of IEEE International Conference on Industrial Technology 2000, Goa, India.

Slock, D. T. M. (1993). On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing, 41*(9), 2811-2825. doi: 10.1109/78.236504

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, ICASSP-98, Seattle, Washington, USA.

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

Srinwasan, S., & Brown, E. (2002). Is speech recognition becoming mainstream? *Computer, 35*(4), 38-41.

Stella, M., Begusic, D., & Russo, M. (2006). *Adaptive Noise Cancellation Based on Neural Network.* Paper presented at the International Conference on Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006., Split, Croatia.

Stinchcombe, M., & White, H. (1989). *Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions.* Paper presented at the Proceedings of the International Joint Conference on Neural Networks, Washington, DC, USA.

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

Tamura, S. (1989). *An analysis of a noise reduction neural network.* Paper presented at the International Conference on Acoustics, Speech, and Signal Processing, 1989. ICASSP-89, Glasgow, UK.

Tamura, S., & Waibel, A. (1988). *Noise reduction using connectionist models.* Paper presented at the International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88, New York, USA.

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

Thumchirdchupong, H., & Tangsangiumvisai, N. (2013, 15-17 May). *A two-microphone noise reduction scheme for hands-free telephony in a car environment.* Paper presented at the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Krabi, Thailand.

Tian, Z.-M., & Wang, A.-Z. (2012, 7-9 July). *The Research of Adaptive Noise Cancellation Technology Based on Neural Network.* Paper presented at the 2012 International Conference on Computing, Measurement, Control and Sensor Network (CMCSN), Taiyuan.

Topolsky, J. (2013). I used Google Glass: the future, but with monthly updates. *The Verge.* http://www.theverge.com/2013/2/22/4013406/i-used-google-glass-its-the-future-with-monthly-updates

Tsuda, Y., & Shimamura, T. (2002). *An improved NLMS algorithm for channel equalization.* Paper presented at the IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002, Scottsdale, Arizona, USA.

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

Urmila, S., & Vilas, T. (2013). A hybrid method for automatic speech recognition performance improvement in real world noisy environment. *Journal of Computer Science, 9*(1), 94 -104.

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

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

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 Gerven, S., & Xie, F. (1997, September). *A comparative study of speech detection methods*. Paper presented at the Proc. 5th Eur. Conf. Speech Communication Technology, Eurospeech' 97., Rhodes, Greece.

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

Vanden Berghe, J., & Wouters, J. (1998). An adaptive noise canceller for hearing aids using two nearby microphones. *Journal of the Acoustical Society of America, 103*(6), 3621-3626.

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.

Weber, M., Crilly, P. B., & Blass, W. E. (1991). Adaptive noise filtering using an error-backpropagation neural network. *IEEE Transactions on Instrumentation and Measurement, 40*(5), 820-825. doi: 10.1109/19.106304

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

Weiyuan, L. (2010, 17-19 November). *Natural user interface- next mainstream product user interface*. Paper presented at the 2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design (CAIDCD), Hangzhou, China.

Widrow, B. (2005). Thinking about thinking: the discovery of the LMS algorithm. *IEEE Signal Processing Magazine, 22*, 100 -106.

Widrow, B., Glover, J. R. J., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., . . . Goodlin, R. C. (1975). Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE, 63*(12), 1692-1716.

Widrow, B., & Lehr, M. A. (1993). *Backpropagation and its applications*. Mahwah: Lawrence Erlbaum Assoc Publ.

Widrow, B., & Luo, F.-L. (2003). Microphone arrays for hearing aids: An overview. *Speech Communication, 39*(1-2), 139-146.

Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York: John Wiley

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.

Wormek, A. K., Ingenerf, J., & Orthner, H. F. (1997). SAM: Speech-aware applications in medicine to support structured data entry. *Journal of the American Medical Informatics Association*, 774-778.

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.

Ye-Yi, W., Dong, Y., Yun-Cheng, J., & Acero, A. (2008). An introduction to voice search. *IEEE Signal Processing Magazine, 25*(3), 28-38. doi: 10.1109/msp.2008.918411

Yoganathan, V., & Moir, T. J. (2008, 2-4 Dec.). *Switched Griffiths-Jim beamformer using the affine projection algorithm*. Paper presented at the 15th International Conference on Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008, Auckland, New Zealand.

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

Zaknich, A. (2005). *Principles of adaptive filters and self-learning systems*. Germany: Springer.

Zhongguang, Q., & Zongyuan, M. (2000). *A new algorithm for neural network architecture study*. Paper presented at the 3rd World Congress on Intelligent Control and Automation, Hefei, China.

Appendix A

Published Papers

A - I

APA based SGJBF Paper Presented at the M2VIP08 Conference

15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08),2-4 Dec 2008,Auckland,New-Zealand

Switched Griffiths-Jim beamformer using the affine projection algorithm

V. Yoganathan and T. J. Moir
School of Engineering and Advanced Technology
Massey University, Auckland, New Zealand
Email: v.yoganathan@massey.ac.nz, tjmoir@massey.ac.nz.

Abstract - **Speech controlled applications are now becoming more and more practical due to advances in technology. These applications vary from command and control instruments, video conferencing to robotics. However, their performance decreases when the acquired speech signal is corrupted by background noise. Numerous research has been done in the last two decades to improve their performance. The switched Griffiths-Jim beamformer is one of the well known methods used to reduce background noise (or interference). This algorithm makes use of two adaptive filters and a voice activity detector. The first filter is used as a beam-steering filter which is updated during speech signals and the second filter is used as a noise cancellation filter which is updated during a noise-alone signal. A voice activity detector is used to control the updates of these adaptive filters. Generally, the normalised least mean squares (NLMS) algorithm is used for the adaptive filters. However, it is found that the convergence rate of the NLMS deteriorates for coloured noise under certain non-stationary conditions where the correlation matrix is "stiff". It has been found that the affine projection algorithm will perform better under these conditions. Therefore, a comparison of these two adaptive filters in this beamformer structure will be discussed in this paper.**

I. INTRODUCTION

In recent years, speech recognition in noisy environments has received a considerable amount of attention due to increased demand for speech based applications [1, 2]. These applications vary from dictation based systems to command-and-control systems. In order for the speech recogniser to perform well, they require a high signal to noise ratio (typically around 20dB). However, in some situations background noise cannot be controlled by the user. Therefore, the accuracy of the speech recognition system needs to be improved by reducing the background noise.

The adaptive noise cancelling method was first introduced in the 1970s to estimate the speech signal corrupted by the background noise [3]. Fig. 1 shows the basic structure of the adaptive noise canceller. As shown in the figure, the first microphone is positioned near the noise source and the second microphone is positioned so that it can pickup the desired speech signal and the noise signal. The basic idea behind it is to use an adaptive filter to suppress the noise components whilst leaving the desired speech signal unchanged. In order for the noise cancellation to work efficiently this algorithm requires the noise signal in both microphones to be coherent. However, it is very difficult to achieve that while keeping the reference signal speech-free. The onset of this is that the filter could very well cancel desired speech as well as the noise.

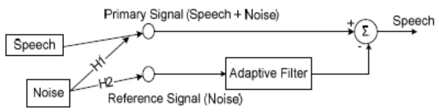


Figure 1. Adaptive noise canceller

To overcome this problem Griffith-Jim proposed a new and improved version of this method in the 1980's [4]. Fig. 2 shows the basic structure of a two microphone Griffiths-Jim beamformer (though more microphones can be used with this structure). The assumption is made that the speech signal is straight ahead, or else a beam-steering algorithm can be used to steer towards the speaker in the so-called "look" direction. A preprocessor to add and subtract the microphone signals is used to obtain the speech-free reference signal.

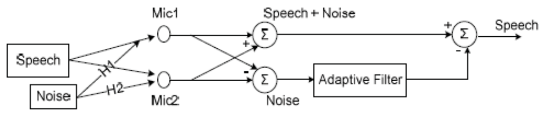


Figure 2. Two-Microphone Griffiths-Jim beamformer

This algorithm works well as long as the target signal arriving at the two microphones are time-aligned. However, this algorithm is prone to target signal cancellation in the presence of steering vector errors. This error can be caused by errors in microphone positions, microphone gains, reverberations, and target direction. This problem was noticed by some researchers and new improved algorithms have been proposed in an attempt to solve it.

In particular, Van Compernolle proposed a new speech beamformer called switched Griffiths-Jim beamformer in 1990 to solve this problem raised here [5]. He showed that signal cancellation can be reduced somewhat by adapting the noise canceller filter parameters only during the noise-alone regions (thus when no speech is present in the received signals). Fig. 3 shows the basic structure of the two microphone switched Griffiths-Jim beamformer. This algorithm uses two adaptive filters and a voice activity detector. These will be explained in the later sections.

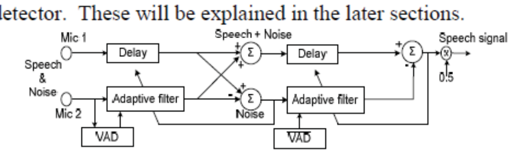


Figure 3. Two-Microphone Switched Griffiths-Jim beamformer

 ISBN: 978-0-473-13532-4

II. ADAPTIVE FILTER

This beamformer algorithm uses two adaptive filters. The first filter is used as a beam-steering filter and is used to obtain an optimal phase alignment between the input speech signals. This filter creates an adaptive look direction and cues in on the desired speech. Therefore, it is only updated during speech signals. The second filter is used as a noise cancellation filter and it is updated during noise-alone signals. A speech detector algorithm is used to control the updates of these adaptive filters.

A normalised least mean squares (NLMS) algorithm is most commonly used adaptive filter with this beamformer structure due to its simplicity. This algorithm is summarised as [6]:

$$e(n) = d(n) - \vec{x}(n)^T \vec{w}(n-1) \tag{1}$$

$$\vec{w}(n) = \vec{w}(n-1) + \mu \vec{x}(n) \left[\vec{x}(n)^T \vec{x}(n) + \gamma \right]^{-1} e(n) \tag{2}$$

where $d(n)$ is the desired signal, $x(n)$ is the vector of regressors, $e(n)$ is the error signal, $w(n)$ is the estimated tap-weight vector and μ is the set-size. Also, → denotes vector signal. The gamma (γ) value is a small positive constant introduced to prevent division by zero (this is generally chosen as 0.0001).

The convergence speed of the NLMS is known to deteriorate under certain coloured noise conditions [7]. The beam-steering filter is not necessary to track all the head movements, so NLMS will be an appropriate choice. However, the noise canceller requires an adaptive filter with the faster convergence rate as it needs to reduce the background noise quickly.

To overcome the problems with NLMS, Ozeki and Umeda proposed a new method called affine projection in 1984 [8]. The affine projection algorithm (APA) is summarised from reference [9] as:

$$\vec{d}_P(n) = \begin{pmatrix} d(n) \\ d(n-1) \\ \vdots \\ d(n-P+1) \end{pmatrix} \text{ and } \vec{x}(n) = \begin{pmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{pmatrix} \tag{3}$$

$$\boldsymbol{X}_P = \left(\vec{x}(n) \vec{x}(n-1) \cdots \vec{x}(n-P+1) \right) \tag{4}$$

$$\vec{e}(n) = \vec{d}_P(n) - \boldsymbol{X}_P(n)^T \vec{w}(n-1) \tag{5}$$

$$\vec{w}(n) = \vec{w}(n-1) + \mu \boldsymbol{X}_P(n) \left[\boldsymbol{X}_P(n)^T \boldsymbol{X}_P(n) + \delta \boldsymbol{I} \right]^{-1} \vec{e}(n) \tag{6}$$

where the subscript T denotes transpose, $\boldsymbol{I}$ denotes the identity matrix, P is the projection order, N is the number of filter weights, and δ is the regularisation parameter. Also, $\boldsymbol{X}_P$ is the P by N excitation matrix, $\vec{d}_P(n)$ is a vector of desired signal, and $e(n)$ is a vector of error signals.

Generally, the projection order is selected to be less than the filter weights $(P < N)$. If $P = 1$, this algorithm is equivalent to the NLMS algorithm and if $P = N$ it is equivalent to the windowed recursive least squares (RLS) algorithm [10]. The APAs computational complexity and convergence rate is considered to be in-between the NLMS and the RLS algorithms. Due to the high computational complexity of the RLS it is not used widely for real-time applications though low-order APA is more appropriate for real-time implementation. Comparison of NLMS and APA has been carried out by many researchers and APA is found to perform better under certain conditions [11, 12].

Often APA is used for echo cancellation [13] and system identification [12] applications. Recently, a faster version of APA has been used with linear constrained minimum variance beamformer structure by [10] to improve the convergence rate. A comparison analysis of the NLMS and APA adaptive filters with the switched Griffith-Jim beamformer is presented here.

III. VOICE ACTIVITY DETECTOR

A voice activity detector (VAD) is used to distinguish between "speech and noise" signals and "noise-alone" signals. It is used to control the updates of the adaptive filter. Voice activity detection algorithms mainly 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 VAD algorithms use energy distribution, timing, pitch, zero crossing rates, cepstral features, spectral information, and periodicity measures.

This paper will concentrate on a direction based method proposed by [14]. This method assumes 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 originating outside of this zone is considered background interference. If the user happens to be outside this predefined area, they could easily move themselves in to this area to use the microphone system. These types of methods restrict the user to stay within this zone of activity where the speech is expected to be.

IV. SPEECH CONTROLLED VIBRATION MONITORING SYSTEM

A vibration monitoring system is used as our target application for this paper. Vibration monitoring systems are used in industrial maintenance programs to monitor the condition of rotating machinery. In recent years, this technique has become a critical component of the routine monitoring program. Machine vibration is usually measured using an accelerometer and the result is analysed to predict the machines performance and reliability. Most unexpected system breakdowns are prevented by this. As a result, this saves money and time for industry.

Vibration monitoring systems are usually handheld devices. This often requires the user to hold the transducer with one hand while controlling the device with the other hand. However, this becomes difficult when the users hands are required to do other tasks, for instance holding on to a ladder

15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08),2-4 Dec 2008,Auckland,New-Zealand

while taking the measurement. Therefore, hands-free control of this system would be a great assistance to the user. At present, there aren't many devices that use hands-free features so this would be an attractive feature in this system. However, our challenge will be to make this instrument work in a difficult noisy environment.

V. SIMULATION

A computer based speech controlled vibration monitoring system was created in Visual Basic and Labview software. This program uses the Microsoft speech recognition engine to translate from a speech signal to valid commands. Fig. 4 shows the front panel of this program. This program uses a National Instruments DAQ card (NI-9233) to acquire the vibration signal from an accelerometer.

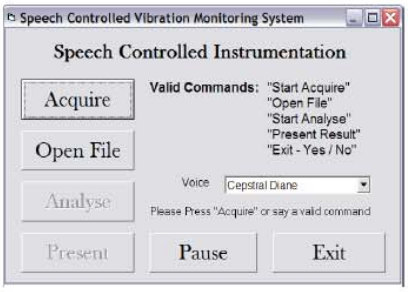


Figure 4. Speech controlled vibration monitoring system

This system allows the following speech commands: "Start acquire", "Open file", "Start analyse" and "Present result". The "Start acquire" command records real time vibration data to a file and displays it in a separate window (Example shown in Fig. 5). Otherwise, the "Open file" command can be used to open an existing file recorded earlier using this program. The "Start analyse" command applies a power spectrum to the original data and presents the results in a separate window. The "Present result" command shows the original data and analysed data as shown in Fig. 6.

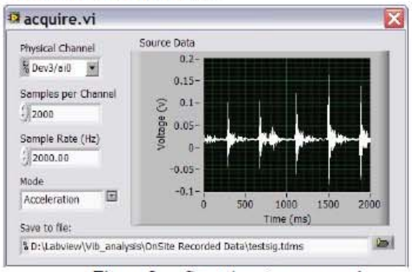


Figure 5. *Start Acquire* command

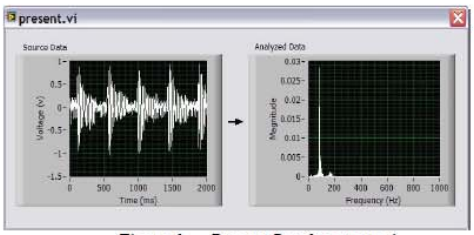


Figure 6. *Present Result* command

Experiment 1:

The switched Griffiths-Jim beamformer structure shown in Fig. 3 is simulated in the Labview software. The input speech signal (file named "kit_001.wav") is used from the "Timit" speech database. Two versions of this beamformer structure is implemented here, one with the APA and another with the NLMS as the noise cancelling filter. The white noise is simulated in the software program and mixed with the speech signal. Since the input signal for this experiment is simulated, the beam-steering filter is not essential.

The primary and reference noise signals are simulated with the transfer function of H1 and H2, respectively. The transfer function of the adaptive noise cancelling filter should converge to $(H1+H2)/(H1-H2)$ provided $(H1-H2)$ has all its root within the unit circle (actually, it is the FIR series expansion of this transfer function, but for $(H1-H2)$ with roots outside the unit circle another solution exists [15]). For this experiment, H1 is set to $2-0.195z^{-1}+0.95z^{-2}$ and H2 is set to 1. According to [7], this transfer function gives an eigenvalue spread of 1.22 for the correlation matrix.

For this simulation, the step-size parameter μ is set to 0.5 and the number of filter weight N is set to 100 (these values are kept the same for NLMS and APA). Also, the regularisation parameter δ is set to 0.1 and the projection order P is set to 10 (these values are used for APA). The signal-to-noise (SNR) ratio in this paper is calculated by:

$$SNR = 10\log_{10}\left(\frac{Power(Speech)}{Power(Noise)}\right)$$

An example of the input primary signal to the switched Griffiths-Jim beamformer (with the SNR of 0dB) is shown in Fig. 7 (a). The simulation results from the NLMS and the APA based beamformer are shown in Fig. 7 (b)-(c). The APA and NLMS based beamformer has an output SNR of 23.79dB and 15.69dB. The output signal from the APA based beamformer clearly shows that it reduces more noise than the NLMS based beamformer.

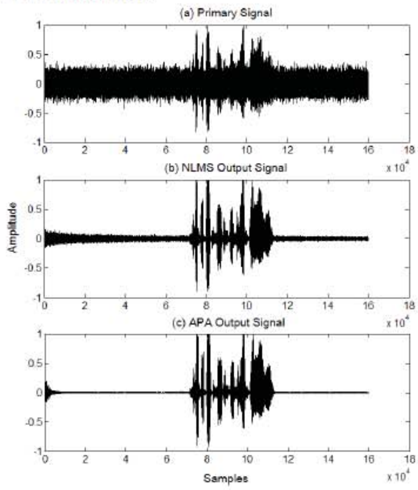


Figure 7. Result from the Switched Griffiths-Jim Beamformer

15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08),2-4 Dec 2008,Auckland,New-Zealand

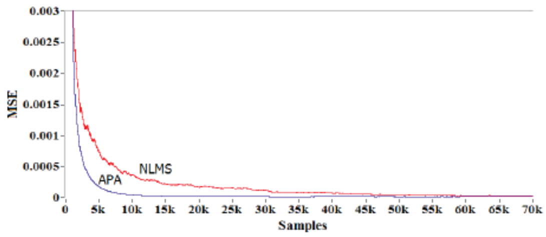


Figure 8. Learning curves of APA and NLMS (eigenvalue spread 1.22)

The mean-square error (MSE) of the noise-alone output signal for the APA and the NLMS is shown in Fig. 8 (it is also known as the "learning curve" in the literature). The APA converged to zero after about 10,000 samples and NLMS is slowly converging to zero after about 65,000 samples. It is evident that APA converged faster than the NLMS algorithm by an order of magnitude and that its minimum MSE is significantly smaller. Fig. 9 shows the comparison analysis of the input SNR vs. the output SNR for the switched Griffiths-Jim beamformer with NLMS and APA. It can be seen from the graph that the overall performance of the APA is better than the NLMS algorithm. Also, APA reduced more noise at low SNRs where it matters most. This is quite important as this is where the speech recognition engine will fail to work.

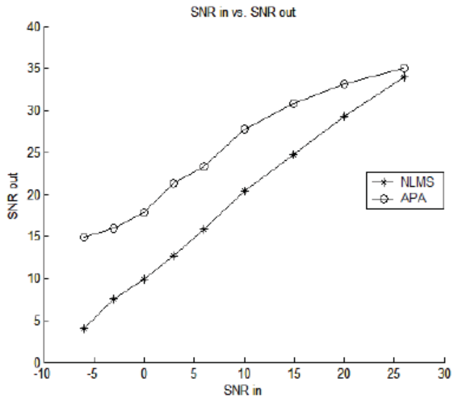


Figure 9. SNR Input vs. Output for the eigenvalue spread of 1.22.

Experiment 2:

A similar set up to experiment one is given here. However for this simulation, H1 is set to $2\text{-}1.9114z^{-1}+0.95z^{-2}$ and H2 is set to 1. This transfer function gives a high eigenvalue spread of 100 for the correlation matrix. Under these conditions the NLMS or LMS has trouble converging. This is because, the correlation matrix has a large condition number (ratio of maximum to minimum eigenvalue) so when forming the least-squares solution it is numerically more difficult to invert [7].

Fig. 10 shows the comparison analysis of input SNR vs. output SNR for the switched Griffiths-Jim beamformer with NLMS and APA ($\delta = 0.1$). Both of these adaptive filters have difficulty reducing the noise when compared with the previous experiment. However, at low SNR the APA still performs better than the NLMS.

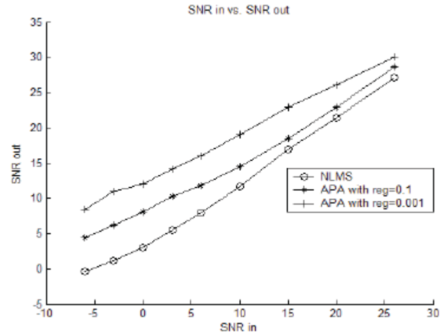


Figure 10. SNR Input vs. Output for the eigenvalue spread of 100.

The MSE of the noise-alone output signal for the RLS, the APA and the NLMS is shown in Fig. 11. In this experiment, RLS converged to zero after 5,000 samples and APA converged after 35,000 samples. As expected, the NLMS is struggling to converge to zero and also it is converging incorrectly to 0.002. The APA converged slightly slower than the previous experiment but still performed better than the NLMS. The RLS algorithm is estimating a FIR filter here, but it can be used to it is full advantage by estimating a pole-zero model.

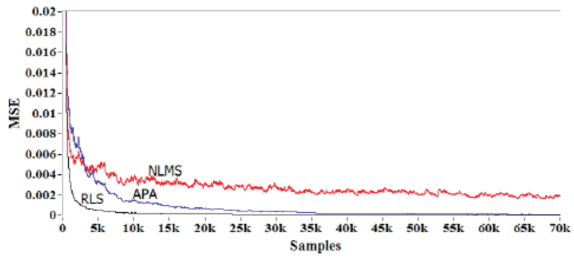


Figure 11. Learning curves of RLS, APA & NLMS (eigenvalue spread 100)

Another simulation is done to show the effect of the regularisation parameter δ on the APA. The regularisation parameter δ is changed to 0.001 and the rest of the values are left unchanged. Fig. 10 shows the comparison of APA with the δ of 0.1 and 0.001. By choosing a smaller regularisation parameter the APA's performance was improved. The MSE of the noise-alone output signal for the APA and the NLMS is shown in Fig. 12. As you can see from the graph, the APA algorithm still performs better under high eigenvalue spread condition whereas the NLMS performance deteriorates. By comparing these two graphs it is clear that APA can perform better when the initial parameters are chosen correctly.

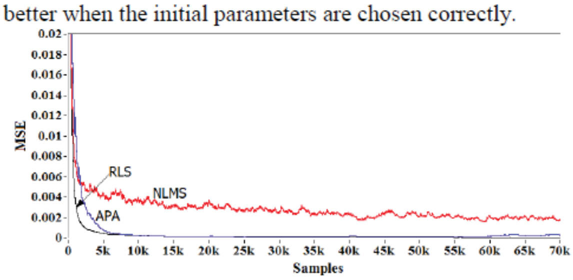


Figure 12. Learning curve of RLS, APA & NLMS (eigenvalue spread 100)

15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08),2-4 Dec 2008,Auckland,New-Zealand

Experiment 3:

This experiment is designed to test the effect of using the proposed beamformer algorithm with the speech controlled vibration monitoring system under a noisy environment. The speech signal is directly in-front of the two microphones. The background noise is from the "NOISEX" database named "Factory floor noise 1". The background noise was played through a speaker at about 30 degree angle to the speech signal. As the recording was performed in real-time the transfer functions are unknown. Here we use the term "real-time" loosely, in that there is some processing delay before the output is heard.

Two versions of the switched Griffiths-Jim beamformer structure is implemented here, one with the APA and another with the NLMS as the noise cancelling filter. The NLMS adaptive filter is used as the beam-steering filter for both of these beamformers. For this filter, μ is set to 0.9 and N is set to 100. For the noise cancelling filter μ is set to 0.9 and N is set to 500 (for the NLMS and APA algorithms). Also, the regularisation parameter δ is set to 0.001 and the projection order P is set to 20 (for APA).

Fig. 13 (a) shows an example input primary signal with the command "Open File" and the factory noise (input SNR is 9.54dB). Fig. 13 (b) shows the output signal from the NLMS based beamformer with the SNR of 20.8dB. Fig. 13 (c) shows the output signal from the APA based beamformer with the SNR of 24.36dB. This shows that APA reduced the noise about 3.56dB more than the NLMS based beamformer. This result is similar to the simulation done in experiment one.

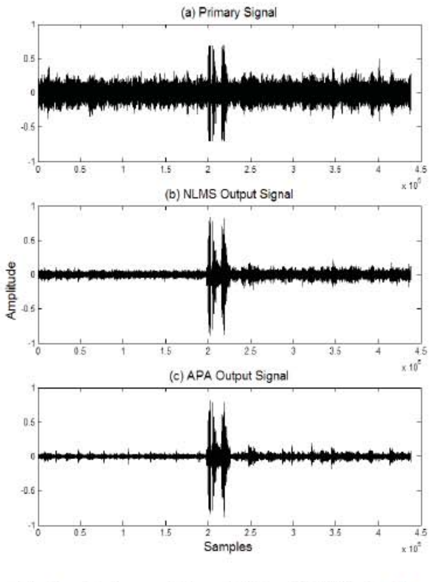


Figure 13. Results from real-time Switched Griffiths-Jim beamformer

The MSE of the noise-alone period from the APA and the NLMS output signal is shown in Fig. 14. This convergence graph is not smooth as the results from the simulations. The reason for this is because the noise signal is continually changing as compared with the simulations. As a result, this adaptive algorithm keeps adapting to the changes to reduce the background noise. However, it is evident that APA converged faster than the NLMS algorithm.

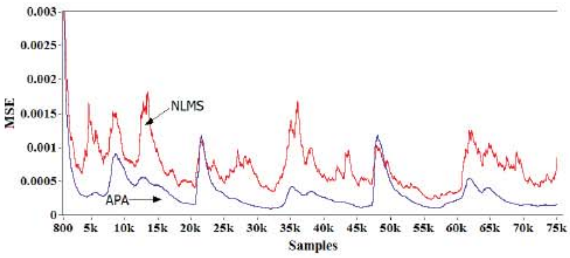


Figure 14. Convergence rate of APA and NLMS (experiment 3)

This results were tested with two different speech controlled vibration monitoring systems; one with defined grammar and another with unlimited grammar. The system with the unlimited grammar file could not recognise any word from the input primary signal or the output signals from both of the beamformers. Alternatively, the system with the defined grammar had 50% recognition with the input primary signal. Also, prefect recognition was achieved with the output signal from both of the beamformers. Since, both of the adaptive filters improved the SNR above 20dB as a result they both performed well with the speech recogniser.

VI. Conclusion

This paper describes a speech controlled vibration monitoring system using the switched Griffiths-Jim beamformer. This system is tested with different levels of SNR. The experimental results show the APA based beamformer reduced the background noise by about 20 dB (in some cases). This is twice the amount of noise reduced by the NLMS based beamformer. These simulations were carefully designed to include examples where the acoustic transfer function identification, resulted in a least-squares problem where the condition number of the correlation matrix is large (hence its inverse is numerically difficult to find).

The real-world experiment also proved that APA gave an improved SNR when applied to the switched Griffiths-Jim beamformer structure than that of the NLMS algorithm. This experiment also shows that using a defined grammar for the speech recognition program improves the recognition rate. It is also proven that using the beamformer with the speech recognition system improves the recognition rate of the system. According to the simulations, at low input SNRs the APA still performed better than NLMS.

References

[1] E. Lleida, J. Fernandez, and E. Masgrau, "Robust continuous speech recognition system based on a microphone array," 1998, pp. 241-244 vol.1.

[2] M. L. Seltzer, B. Raj, and R. M. Stern, "Speech recognizer-based microphone array processing for robust hands-free speech recognition," 2002, pp. I-897-I-900 vol.1.

15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08),2-4 Dec 2008,Auckland,New-Zealand

[3] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE,* vol. 63, pp. 1692-1716, 1975.

[4] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *Antennas and Propagation, IEEE Transactions on [legacy, pre - 1988],* vol. 30, pp. 27-34, 1982.

[5] D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in *1990 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-90)*, Albuquerque, NM, USA, 1990, pp. PP 833-836

[6] P. S. R. Diniz, *Adaptive filtering algorithms and practical implementation*. USA: Kluwer Academic Publishers, 1997.

[7] S. Haykin, *Adaptive Filter Theory*. USA: Prentice-Hall, 1986.

[8] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics & Communications in Japan,* vol. 67, pp. 19-27, 1984.

[9] G. Rombouts, "Adaptive filtering algorithms for acoustic echo and noise cancellation," in *Department of Electrical Engineering* vol. PhD Leuven: Katholieke Universiteit Leuven, 2003, p. 177.

[10] Y. R. Zheng and R. A. Goubran, "Adaptive beamforming using affine projection algorithms," in *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*, 2000, pp. 1929-1932 vol.3.

[11] S. Hyun-Chool, A. H. Sayed, and S. Woo-Jin, "Variable step-size NLMS and affine projection algorithms," *Signal Processing Letters, IEEE,* vol. 11, pp. 132-135, 2004.

[12] K. Ikeda, "Convergence analysis of block orthogonal projection and affine projection algorithms," *Signal Processing,* vol. 82, pp. 491-496, 2002.

[13] J. Benesty, P. Duhamel, and Y. Grenier, "A multichannel affine projection algorithm with applications to multichannel acoustic echo cancellation," *Signal Processing Letters, IEEE,* vol. 3, pp. 35-37, 1996.

[14] H. Agaiby and T. J. Moir, "Knowing the wheat from the weeds in noisy speech," in *5th European Conference on Speech Communication and Technology - EUROSPEECH*, Rhodes, Greece, 1997, pp. 1119-1122.

[15] T. J. Moir, "A z-domain transfer function solution to the non-minimum phase acoustic beamformer," *Inter Journal Systems Science,* vol. 38, pp. 563-575, 2007.

A - II

Paper Presented at the ISSPA 2010 Conference

10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)

Speech Enhancement Using a Nonlinear Neural Switched Griffiths-Jim Beamformer

V. Yoganathan, T. J. Moir

School of Engineering and Advanced Technology, Massey University
Auckland, New Zealand
v.yoganathan@massey.ac.nz, t.j.moir@massey.ac.nz

ABSTRACT

This paper presents a special nonlinear switched Griffiths-Jim beamformer (SGJBF) structure. The main objective of this paper is to reduce the background noise from an acquired speech signal. The interference we considered here is non-stationary in nature and can arrive from a variety of potential sources; for example, competing talkers, radio, TV and so on. In this paper, we propose an adaptive Time Delay Neural Network (TDNN) based nonlinear noise canceller. The proposed structure consists of a three-layer feedforward network with partially connected layers to achieve real-time processing. The error backpropagation learning algorithm is used here to train the TDNN. This system is tested with different types of interference signals from the Noise-X database. A comparison analysis of the proposed structure and the traditional linear adaptive beamformer is presented here. The nonlinear approach investigated here show remarkable improvements over the previous linear based beamforming approach.

Indexing terms – adaptive filter, generalized sidelobe canceller, multilayer perceptron, noise cancellation

1. INTRODUCTION

Acquiring a noise-free speech signal in an adverse acoustic environment has created great difficulties for many speech related applications. Hands-free telephony, hearing aids, video or teleconferencing, speaker identification and speech-controlled devices are some applications that require clean speech to function efficiently. In the past few decades, various algorithms have emerged aimed at reducing the background noise from the acquired signal. These algorithms can vary from single to multi-microphone methods. However, the underlying idea behind most popular algorithms is to use an adaptive filter to reduce the interference signal

Early work on noise reduction can be referred to Widrow and his colleagues for developing the adaptive noise cancelling structure in the 70's [1]. This structure is used largely for applications where the speech signal is isolated from the reference signal and noise signals are correlated in both channel. However, this traditional structure is unsuitable for many applications due to the difficulties in acquiring such a speech-free reference signal.

This problem has been identified by many researchers. In particular, Griffiths and Jim developed a microphone array beamformer algorithm where the signal is added and subtracted to give a new primary and a reference signal [2]. The two-microphone structure of this beamformer is shown in Figure 1. The main drawback with this method is that it requires the speech signal arriving at the microphones to be phase-aligned. Van Compernolle improved this structure by introducing another adaptive filter to align the speech signals and control the updates of these adaptive filters by using a voice activity detector (VAD) [3]. The two-microphone switched Griffiths-Jim beamformer (SGJBF) structure is shown in Figure 2.

This beamformer structure has two sections: a beam-steering section and a noise canceller section. The purpose of the beam-steering section is to obtain a phase alignment between the input signals. The noise canceller section is used to reduce the unwanted background noise. The first section is only updated during "speech and noise period" and the second section is updated during a "noise alone" period. By controlling the update of the first section we can avoid the beam from steering to the noise source and by controlling the update of the second section, speech cancellation can be avoided while noise reduction can be minimized.

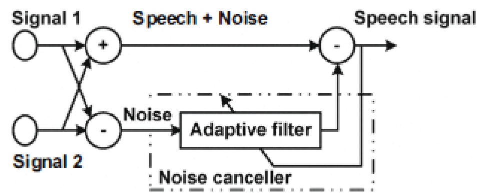


Figure 1. Two-channel GJBF.

978-1-4244-7167-6/10/$26.00 ©2010 IEEE

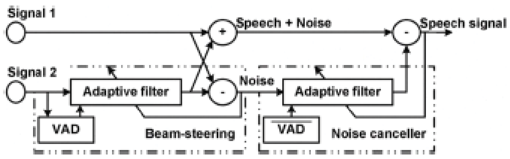


Figure 2. Two-channel switched GJBF.

A VAD is used here to distinguish between the "speech and noise" period and "noise alone" period. Clearly, this is a disadvantage of this kind of approach over other so-called unsupervised adaptive filtering approaches using say Independent Component Analysis (ICA) [4]. However, it is balanced by the need for a system that can potentially work in real-time.

Generally, a linear adaptive filter is used for the beam-steering and the noise cancelling sections. This area of research is exhaustive due to the ease of implementation with today's processing power and it has even been implemented in hearing aids [5]. The idea of using a nonlinear adaptive filter was first introduced by Gabor in 1954 with Volterra series [6]. In recent years, many researchers have proven that using a nonlinear adaptive filter for noise cancellation produces better results than the conventional linear adaptive filters. As a result, there has been a broad interest in this area. Volterra filters, extended Kalman filters and neural filters are few examples of nonlinear adaptive filters. Amongst the existing filters, this paper concentrates on a neural network (NN) based adaptive noise canceller (ANC) that is both beneficial and practical to implement.

An introduction to the NN area can be found in the following references [7, 8]. In recent times, neural networks have been successfully used for a range of applications such as system identification, pattern recognition, image processing, speech recognition and adaptive beamformer for antenna array applications [9]. Early work on NN can be referred to McCulloch and Pitts in the early 40s [10]. Speech enhancement using NN has been a new area for NN researchers [11, 12].

Tamura and Waibel [13, 14] introduced a noise reduction algorithm based on a NN to map from noisy speech to a noise-free speech signal. They used a four-layered feedforward NN to reduce noise. This was tested on an artificially generated noisy Japanese speech data using stationary and non-stationary noise. This method requires a clean speech signal for training the network and that is not possible to obtain it in all cases. Although the output signal from the network was clear, they have concluded that processing did not improve the intelligibility of the speech signal.

Kencht [15] introduced the idea of using nonlinear noise filtering and the perceptron based beamformer. This paper only considers a fixed beamformer, where the weights are estimated in the training period and they are not updated during the test period. Their research lead to work on nonlinear two-microphone based beamformer for speech enhancement in hearing aid applications [16]. They only used the GJBF structure (as shown in Figure 1) with the fixed NN based noise reduction algorithm. Our study will extend this work to SGJBF structure using a partially connected three-layer time delay neural network (TDNN) based ANC.

The rest of the paper is organized as follows. Section 2 introduces the NN algorithm used with the SGJBF algorithm. Section 3 describes the simulation results and discussion whilst section 4 presents our conclusions.

2. NEURAL SWITCHED GRIFFITHS-JIM BEAMFORMER

The error signal $e(k)$ is given by:

$$e(k) = d(k) - y(k) \tag{1}$$

where $d(k)$ is the desired signal, $y(k)$ is the estimated signal from the adaptive filter and it is calculated by:

$$y(k) = f\left(\sum_{j=1}^{P}\left(v_j(k)z_j(k)\right) + a\right) \tag{2}$$

where $v(k)$ is the second weight vector at the hidden layer, P is the number of neurons and a is the bias value at the output layer. $z(k)$ is the neuron value at the hidden layer and it is calculated by:

$$z(k) = f\left(\sum_{i=1}^{N}\left(w_i(k)x_i(k)\right) + b_i\right) \tag{3}$$

where $w(k)$ is the first weight vector at the hidden layer, $x(k)$ is the reference signal, $f(x)$ is the sigmoid function, b is the bias value at the hidden layer and N is the number of inputs to the neuron. The neuron transfer function is chosen to be the hyperbolic tangent nonlinear sigmoid function.

All the weights are initialized to small random value between 1 and -1 and are updated at each iteration using the error backpropagation algorithm. This learning algorithm is most commonly used to train the NN. The new weight values are given by:

$$w(k+1) = w(k) + \left(h1e(k)v(k)z'(k)x(k)\right) + \alpha\left(w(k-1) - w(k)\right) \tag{4}$$

$$v(k+1) = v(k) + \left(h2e(k)y'(k)z(k)\right) + \beta\left(v(k-1) - v(k)\right) \tag{5}$$

where $h1$ and $h2$ are the learning rate, α and β are the momentum factor, $z'(k)$ is the derivative of the hidden layer neuron, and $y'(k)$ is the derivative of the output layer neuron. For more details on choosing an appropriate parameter values for this algorithm and their effect on the performance of this algorithm can be referred to in the following paper [17].

3. RESULTS AND DISCUSSION

The neural based SGJBF (as shown in Figure 3) and the traditional NLMS based SGJBF are simulated in the Labview programming language G. A VAD based on automatic variance control has been used with these beamformer's to control the adaptive filters [18]. With such a simple energy based VAD we must assume that the desired speech has more power than any interference. This does not act as a major drawback since there are other more sophisticated VAD methods that exist [19].

A partially connected multi-layer perceptron with one hidden layer is used here. This three-layer feedforward time delay neural network structure is shown in Figure 4. The NN structure used here has 20 input nodes, 17 hidden nodes and 1 output node. Each hidden neuron has 4 inputs from the input node and all the hidden nodes are connected to the output layer. The NLMS based SGJBF has 85 weight coefficients (this is equivalent to the total number of weights used in the NN). The step-size value and the learning rates are set as 0.8. These parameters are chosen experimentally and they are kept the same throughout the experiments. Initially, the NN structure is trained with a small section of the interference signal.

The quality of the output signal is measured by signal to interference ratio (SIR) and it is calculated by:

$$SIR = 10\log_{10}\left(\frac{Power(Speech+Noise)-Power(Noise)}{Power(Noise)}\right) \quad (6)$$

Several experiments were conducted by varying the level of the interference signal. The experiment is set up as follows: the target speech signal is generated in front of the two microphones and the interference signal is generated at about 45-degree angle to the microphones. The speech signal contains the word "one" and the interference signal consist of white noise and computer fan noise. The interference signals used in the experiments are from the Noise-X database.

Figure 5 shows the comparison of the input SIR verses output SIR for both SGJBFs. The output signal from the NLMS based SGJBF showed only a small improvement in SIR of about 5.5 dB. On the other hand, the NN based SGJBF showed a great improvement in SIR of about 13 dB in most cases and in one particular case an impressive improvement of 22.6 dB was achieved.

Figure 6 shows one example from this experiment. Figure (a) and (b) shows the input primary signal and reference signal that contain the word "one" and the interference signal (with the SIR of 11.49 dB and 12.14 dB). Figure (c) and (d) shows the output signal from the NLMS and the NN based SGJBF (with the SIR of 16.17 dB and 24.89 dB). Figure 7 shows the comparison of the mean square error (MSE) of the output signal of the NLMS and NN (calculated only during the noise alone period). The step size and learning rate are set at 0.8. We can see from the graph that the NN converged more smoothly than the NLMS.

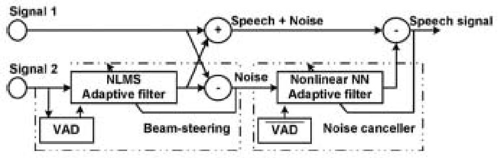


Figure 3. Proposed nonlinear neural SGJBF structure.

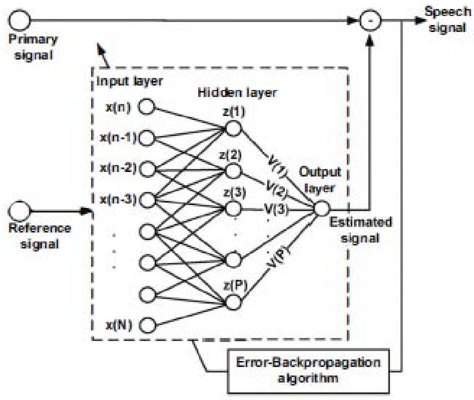


Figure 4. Nonlinear three-layer feedforward TDNN ANC.

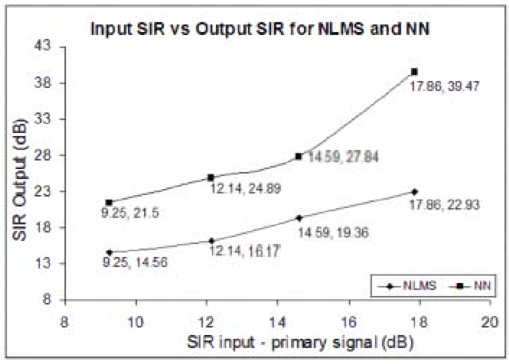


Figure 5. Comparison of SIR input and output graph.

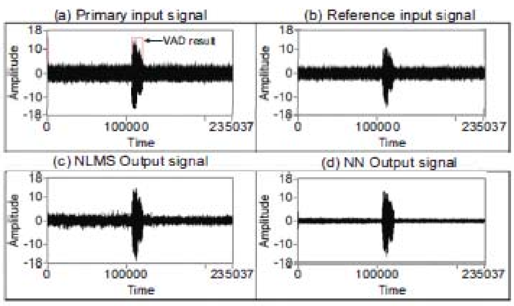


Figure 6. Input and output signal from the white noise interference experiment.

Table 1 shows the comparison of input and output signal for both NLMS and NN based SGJBF for different type of interference signals. As seen from the results the NN based SGJBF outperformed the traditional NLMS based SGJBF by up to 5 dB for different type of interference signals and at different SIR.

TABLE I. COMPARISON OF NN AND NLMS SGJBF WITH DIFFERENT INTERFERENCE SIGNALS.

	Input SIR (dB)		Output SIR (dB)	
Noise Signal type	Primary	Reference	NLMS	NN
Babble noise	19.39	16.71	29.97	34.85
	13.32	10.45	24.23	29.35
Factory noise	14.06	13.56	23.35	26.66

The complexity of the NN is $O(N^2)$ [20] and that of NLMS is $O(N)$. NN certainly requires significantly more computational time than NLMS. This is definitely the main tradeoff in achieving such improved noise reduction.

4. CONCLUSION

This paper described a new method of noise cancellation algorithm based on nonlinear NN adaptive filtering. It is used for the noise cancellation section of the SGJBF structure. A partially connected three-layer TDNN structure was implemented for rapid real-time processing. This algorithm has shown promising noise reduction improvement over the previous linear adaptive filter structure. However, the performance of the NN greatly depends on the amount of training given and also the size of the NN structure. Future work will be to further analyze the performance of the NN beamformer with a speech recognition system.

REFERENCES

[1] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proceedings of the IEEE, vol. 63, 1975.

[2] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," Antennas and Propagation, IEEE Transactions on, vol. 30, pp. 27-34, 1982.

[3] D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in *International Conference on Acoustics, Speech, and Signal Processing*, ICASSP-90., Albuquerque, NM 1990, pp. 833-836 vol.2.

[4] H. Sahlin and H. Broman, "Separation of real-world signals," *Signal Processing*, vol. 64, pp. 103-113, 1998.

[5] B. Widrow and F.-L. Luo, "Microphone arrays for hearing aids: An overview," *Speech Communication*, vol. 39, 2003.

[6] D. Gabor, "Communication theory and cybernetics," in *IRE Transactions on Citcuit theory*. vol. CT-1, 1954, pp. 19-31.

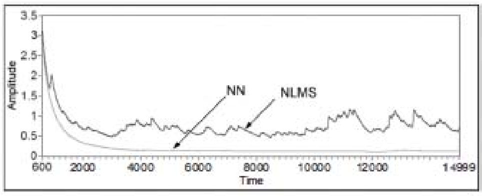


Figure 7. Comparison of MSE of the output signals.

[7] M. Y. Kiang, "Neural Network," in Encyclopedia of information system. vol. 3, D. Bidgoli, Ed.: *Elsevier Science (USA)*, 2003, pp. 303-315.

[8] K. Gurney, An introduction to neural networks. UK,1997.

[9] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing: a review," *Signal Processing*, vol. 82, pp. 547-561, 2002.

[10] S. Hayman, "The McCulloch-Pitts model." in *International Joint Conference on Neural Networks*, 1999.

[11] F. Liew Ban, A. Hussain, and S. A. Samad, "Speech enhancement by noise cancellation using neural network," in *Proceedings of TENCON 2000*, 2000, pp. 39-42 vol.1.

[12] M. Stella, D. Begusic, and M. Russo, "Adaptive Noise Cancellation Based on Neural Network," in Software in *International Conference on Telecommunications and Computer Networks 2006. SoftCOM 2006.* , 2006, pp. 306-309.

[13] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *International Conference on Acoustics, Speech, and Signal Processing. ICASSP-88*, 1988, pp. 553-556 vol.1.

[14] S. Tamura, "An analysis of a noise reduction neural network," in *International Conference on Acoustics, Speech, and Signal Processing 1989. ICASSP-89*, 1989, vol.3.

[15] W. G. Knecht, "Nonlinear noise filtering and beamforming using the perceptron and its Volterra approximation," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 55-62, 1994.

[16] W. G. Knecht, M. E. Schenkel, and G. S. Moschytz, "Neural network filters for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, 1995.

[17] Z. Shi and V. Cherkassky. "Factors controlling generalization ability of MLP networks." in *International Joint Conference on Neural Networks. IJCNN '99*, 1999, vol.1.

[18] T. J. Moir, "Automatic variance control and variance estimation loops," *Circuits Systems and Signal Processing*, vol. 20, pp. 1-10, 2001.

[19] H. Agaiby and T. J. Moir, "Knowing the wheat from the weeds in noisy speech," in *5th European Conference on Speech Communication and Technology*, Rhodes, Greece, 1997.

[20] A. Shareef, et al. "Comparison of MLP neural network and kalman filter for localization in wireless sensor networks," in *Proceedings of the 19th International Conference on Parallel and Distributed Computing and Systems*. 2007. MA, USA.

A - III

Paper Presented at the ICSP 2010 Conference

ICSP2010 Proceedings

Multi-microphone Adaptive Neural Switched Griffiths-Jim Beamformer for Noise Reduction

V. Yoganathan, T. J. Moir
School of Engineering and Advanced Technology, Massey University
Auckland, New Zealand
v.yoganathan@massey.ac.nz, t.j.moir@massey.ac.nz

***Abstract*—This paper proposes a novel multi-microphone nonlinear neural network based switched Griffiths-Jim beamformer structure for speech enhancement. The main objective of this algorithm is to reduce real-world interference signals such as radio, television or computer fan noise from an acquired speech signal. The proposed algorithm improves the current design of the switched Griffiths-Jim beamformer structure by introducing an adaptive nonlinear neural network filter for the noise reduction section. The network topology used here is a partially connected three-layer feedforward neural network structure. The error backpropagation algorithm is used here as the learning algorithm. A comparison analysis of the traditional three-microphone linear beamformer and the proposed three-microphone neural switched Griffiths-Jim beamformer structure is discussed here. They are both tested with different types of interference signal from the Noise-X database. All the experiments are conducted in real-world surroundings. The nonlinear approach introduced here shows remarkable improvement over the previous linear adaptive beamformer approach.**

Keywords-Nonlinear adaptive filter; Speech enhancement; Time delay neural network; multi-layer perceptron; Generalised sidelobe canceller

I. INTRODUCTION

Speech enhancement in a noisy environment has generated many challenges for speech related applications. Desired speech signals are often distorted by interference signals generated from different sources such as computer fans, television, radio and competing speakers, to name but a few. Many researches have emerged through the years with the task of reducing the background noise from an acquired speech signal. These noise reduction techniques generally vary from single microphone to multi-microphone systems. However, the underlying idea behind most widely used algorithms is to use an adaptive filter to minimize the interference signal.

An Adaptive Noise Canceller (ANC) has been used for reducing background noise since the early 70's [1]. The basic structure of this algorithm is shown in Fig. 1. The first microphone is positioned in such a way as to acquire the desired speech as well as the background noise. The second microphone is positioned near the background noise source so (in theory at least) it does not acquire any speech signal. In an ideal situation, subtracting the microphone signals will minimize the noise components while leaving the desired signal unaffected. This algorithm is mostly used for applications where the speech signal can be isolated from the second microphone and at the same time noise signals at the microphones are kept correlated.

978-1-4244-5900-1/10/$26.00 ©2010 IEEE

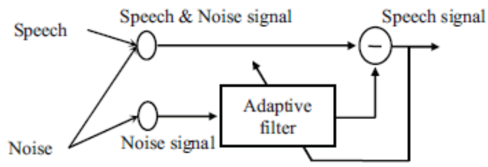


Figure 1. Adaptive noise canceller.

For many speech related applications it is difficult to acquire a noise alone signal while keeping the noise correlated in both microphones. Consequently, this could lead to speech as well as noise cancellation. This problem has been recognized by numerous researchers and different solutions have been proposed. In particular, an adaptive microphone array beamformer algorithm was introduced by Griffiths and Jim where the signals are added to attain the first signal and they are subtracted to attain the second signal [2]. The basic structure of this algorithm is shown in Fig. 2. This design has also been known as the Generalized Sidelobe Canceller (GSC). This algorithm requires the desired speech signal arriving at both microphones to be phase-aligned. If this condition is not met the output speech signal will be distorted due to speech cancellation.

This algorithm was improved by Van Compernolle with another adaptive filter to align the speech signals and also a switch is used to control the updates of these adaptive filters [3]. Fig. 3 shows the basic structure of the two-microphone Switched Griffiths-Jim Beamformer (SGJB) algorithm. This algorithm has a phase alignment section and a noise canceller section. The phase alignment section is used to align the speech signals between the microphones and it is only updated during a speech segment. The noise canceller section is used to minimize the unwanted background noise and it is only

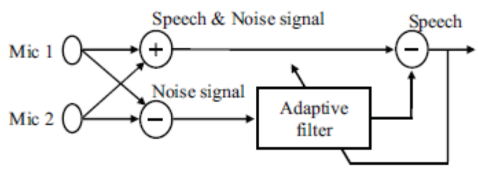


Figure 2. Two-microphone Griffiths-Jim Beamformer.

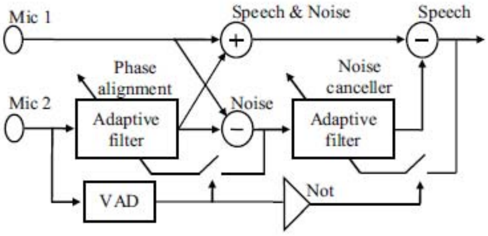


Figure 3. Two-microphone Switched Griffiths-Jim Beamformer.

updated during a noise-alone segment. The update of the adaptive filters is controlled in order to avoid the phase from aligning to the incorrect noise source and also to avoid the speech distortion.

A Voice Activity Detector (VAD) algorithm is used to differentiate between a "speech and noise" segment and a "noise-alone" segment. VAD algorithms vary from energy based systems to direction based systems. Energy based system are widely used due to their simplicity. However, with these systems an assumption is made that the desired speech signal has more power than any background noise. For applications where this condition can not be met more sophisticated VAD methods can be used instead [4].

The phase alignment and the noise canceller sections normally use a linear adaptive filter. The Least Mean Squares (LMS) algorithm is one of the most commonly used linear adaptive algorithms. This area has been researched extensively due to the ease of implementation for practical use with today's processing power. In recent years, there has been a broad interest in nonlinear adaptive filters due to much research that has proven better results than the conventional linear adaptive filters[5]. Early work on nonlinear adaptive filters can be seen in Gabor in 1954 using Volterra series [6]. Examples of nonlinear adaptive filters include: Voltera filters, extended Kalman filters and neural filters, to name but a few. Amongst the existing nonlinear adaptive filters, this paper focuses on a Neural Network (NN) based filter which is both beneficial and practical to implement.

The paper is organized as follows: section 2 summaries pervious work on NN based noise reduction algorithms and introduces the Time Delay Neural Network (TDNN) algorithm. Section 3 describes simulation results and discussion, whilst section 4 presents the conclusion.

II. Neural Network

An introduction to the area neural network area can be found in references [7, 8]. In recent times, neural networks have been used for a range of signal processing applications such as system identification, pattern recognition, image processing, speech recognition and adaptive beamformers for antenna array applications [9]. Early work on neural networks can be referred to Widrow and Hoff [10, 11] whose work lead to the first neuron structure called an Adaptive Linear Element (ADALINE). It is a single neuron linear adaptive filtering structure based on the LMS algorithm.

The research conducted by Tamura and Waibel introduced a noise reduction algorithm based on a NN to map from noisy speech to a noise-free speech signal [12]. A four-layered feedforward NN algorithm was used to reduce the interference signal. This system was tested on artificially generated noisy Japanese speech data using stationary and non-stationary noise. Their method requires a clean speech signal for training the network however which is not possible to obtain it in all cases. Although the output signal from the network was clear, they have concluded that processing did not improve the intelligibility of the speech signal.

Kencht introduced the idea of combining nonlinear noise filtering and the perceptron based beamformer [13]. This research was only limited to a fixed beamformer algorithm. Hence, the adaptive filter weights are estimated in the training period and it is used for the test. This research led to further work on nonlinear dual-microphone based beamformers for speech enhancement in hearing-aid applications [14]. However, they only considered the Griffiths-Jim Beamformer structure (shown in Fig. 2) with the NN based noise reduction algorithm.

The current research presents a three-microphone based adaptive nonlinear neural SGJB. Research has shown that one hidden layer can approximate any measurable function given sufficient amount of hidden units [15]. Therefore, a three-layer feedforward structure with one hidden layer will be used here. The TDNN algorithm is described below.

The error signal $e(k)$ is given by

$$e(k) = d(k) - y(k) \tag{1}$$

where $d(k)$ is the desired signal, $y(k)$ is the estimated signal from the adaptive filter.

Each neuron value at the hidden layer is calculated by

$$z(k) = f\left(\sum_{i=1}^{N}\left(w_i(k) \times x_i(k)\right) + b_i\right) \tag{2}$$

where $w(k)$ is the first weight vector at the hidden layer, $x(k)$ is the reference signal, $f(x)$ is the sigmoid function, b is the bias value at the hidden layer, N is the number of inputs to the neuron. The neuron transfer function is chosen to be the hyperbolic tangent nonlinear sigmoid function.

The output of the network is given by

$$y(k) = f\left(\sum_{j=1}^{P}\left(v_j(k) \times z_j(k)\right) + a\right) \tag{3}$$

where $v(k)$ is the second weight-vector of the hidden layer, P is the number of neurons, a is the bias value at the output layer.

All the weights are initialized to small random values between 1 and -1, and they are updated at each iteration using the error backpropagation algorithm [16]. This learning algorithm is most commonly used to train the NN algorithm. The new weight value is given by:

$$w(k+1) = w(k) + \left(h1 \times e(k) \times v(k) \times z'(k) \times x(k)\right) \tag{4}$$

$$v(k+1) = v(k) + \left(h2 \times e(k) \times y'(k) \times z(k)\right) \tag{5}$$

where $h1$ and $h2$ are the learning rate, $z'(k)$ is the derivative of the hidden layer neuron, and $y'(k)$ is the derivative of the output layer neuron.

III. Results and Disscussion

The following hardware products are used for this experiment: three omni-directional microphones, a pre-amplifier, National Instruments data acquisition card (NI-DAQ 9233) and a personal computer for data analysis. A three-microphone TDNN based SGJB and a three-microphone Normalised Least Mean Squares (NLMS) based SGJB has been simulated in the Labview programming language G. Fig. 4 shows the general structure of the proposed three-microphone neural SGJB algorithm. For the proposed algorithm a NLMS based adaptive filter is used for the phase alignment section and a NN based filter is used for the noise canceller section. For the traditional beamformer both sections use a NLMS based adaptive filter. A VAD based on automatic variance control has been used with these beamformer's to control update of these adaptive filters [17].

The NN structure implemented here is a three-layer feedforward network (one hidden layer) and a partially connected structure. The main advantage of using such a system compared to a fully connected NN structure is to reduce the processing power (this will increase the ability to do real-time processing). Fig. 5 shows the layout of the proposed nonlinear adaptive NN filter used for the noise canceller. This NN structure has 8 input nodes, 5 hidden nodes and 1 output node. Each hidden neuron has 4 inputs from the input node and all the hidden nodes are connected to the output layer.

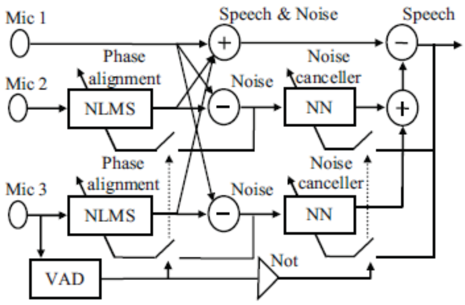


Figure 4. Three-microphone Switched Griffiths-Jim Beamformer.

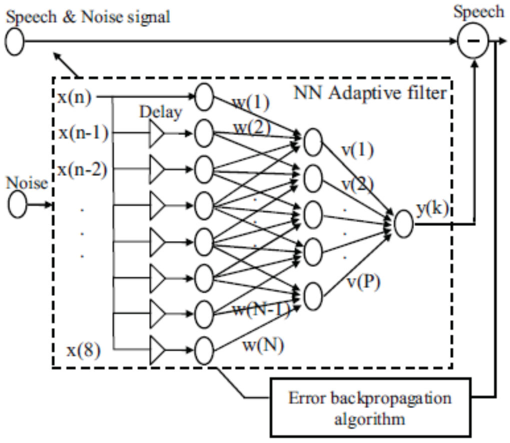


Figure 5. Partially connected three-layer feedforward ANC structure.

Fig. 6 illustrates the experimental arrangement. The three-microphones are positioned in a linear array approximately 15 cm apart. The experiment is set up as follows: the desired speech signal is generated in-front of the microphones and the interference signals are generated at an angle to the microphones (see Fig. 6). The input signal contains the spoken words "One Two Three Four Five" and the interference signals are babble noise and factory noise. The babble noise is positioned at about 65° angle on the right and the factory noise is positioned at about 30° angle on the left. All the interference signals used in this experiment are from the Noise-X database.

After carrying out several experiments the following parameters were chosen for both of the NN filters: the number of inputs to the network was set to 20, the number of neurons was set to 17, the number of inputs to a neuron was set to 4 and the learning rates were set to 0.8. For the NLMS 85 was chosen as the number of weights (this is equivalent to the total weights used for the NN filter in both layers). The step-size value was set as 0.2. Initially, the NN structure was trained with a small section of the interference signal (3000 samples).

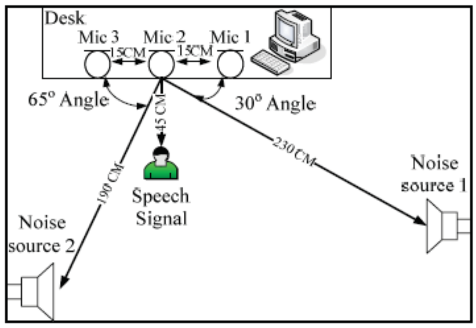


Figure 6. Experiment setup.

The quality of the output signal was measured by Signal to Noise Ratio (SNR) and it was calculated by:

$$SNR = 10\log_{10}\left(\frac{Power(Speech + Noise)}{Power(Noise)}\right) \qquad (6)$$

Fig. 7 (a) shows the input signal from the first microphone (b) and (c) shows the output signal from the NLMS and the NN based beamformer. The NLMS based SGJB showed a SNR improvement of 3.95 dB and the NN based SGJB showed a SNR improvement of 8.32 dB. Fig. 8 shows the comparison of Mean Square Error (MSE) of the output signal of the NLMS and NN. This graph demonstrates that the NN converged more smoothly when compared to NLMS. In addition, an informal listening test showed a great improvement in the NN based beamformer output when compared to the NLMS beamformer.

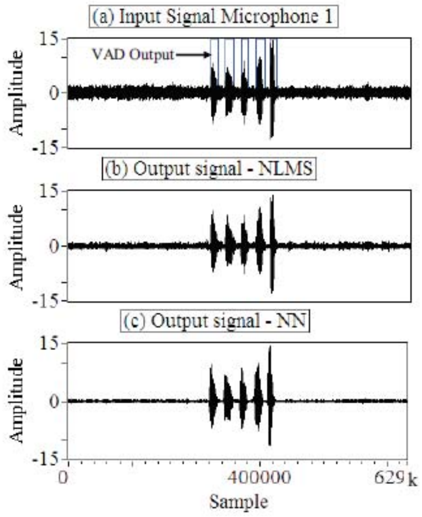


Figure 7. Experiment results (a) input signal (b) output signal from the NLMS SGJB, and (c) the NN SGJB.

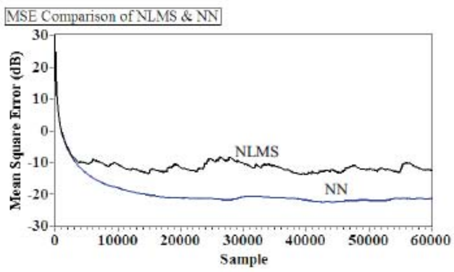


Figure 8. Comparison of the mean square error for the output signals.

IV. CONCLUSION

This paper describes a new method for speech enhancement based on a nonlinear NN based adaptive filtering. A three-microphone neural SGJB structure was analysed with traditional NLMS for the phase alignment section and the NN filter for the noise cancellation section. A partially connected three-layer TDNN structure was implemented here. The error backpropagation algorithm is used as the learning algorithm to train the NN filter. The error backpropagation method is a slow gradient algorithm and it can be easily replaced with other algorithms like the fast backpropagation algorithm [18]. The proposed algorithm has shown promising noise reduction improvement over the linear adaptive filter structure. Future work in this area will consist of combining the NN beamformer with a speech recognition system as a practical application to further analyze the performance of the algorithm.

REFERENCES

[1] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proceedings of the IEEE, vol. 63, pp. 1692-1716, 1975.

[2] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," IEEE Transactions on Antennas and Propagation, vol. 30, pp. 27-34, 1982.

[3] D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in International Conference on Acoustics, Speech, and Signal Processing, ICASSP-90., Albuquerque, NM 1990, pp. 833-836 vol.2.

[4] H. Agaiby and T. J. Moir, "Knowing the wheat from the weeds in noisy speech," EUROSPEECH, Rhodes, Greece, 1997, pp. 1119-1122.

[5] D. Gabor, "Communication theory and cybernetics," in IRE Transactions on Citcuit theory. vol. CT-1, 1954, pp. 19-31.

[6] M. Coker and D. Simkins, "A nonlinear adaptive noise canceller," 1980, pp. 470-473.

[7] M. Y. Kiang, "Neural Network," in Encyclopedia of information system. vol. 3, D. Bidgoli, Ed.: Elsevier Science (USA), 2003, pp. 303-315.

[8] K. Gurney, An introduction to neural networks. UK: UCL Press 1997.

[9] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing: a review," Signal Processing, vol. 82, pp. 547-561, 2002.

[10] F. Liew Ban, A. Hussain, and S. A. Samad, "Speech enhancement by noise cancellation using neural network," in TENCON 2000. Proceedings, 2000, pp. 39-42 vol.1.

[11] M. Stella, D. Begusic, and M. Russo, "Adaptive Noise Cancellation Based on Neural Network," in SoftCOM, 2006, pp. 306-309.

[12] S. Tamura, "An analysis of a noise reduction neural network," in ICASSP, 1989, pp. 2001-2004 vol.3.

[13] W. G. Knecht, "Nonlinear noise filtering and beamforming using the perceptron and its Volterra approximation," IEEE Transactions on Speech and Audio Processing, vol. 2, pp. 55-62, 1994.

[14] W. G. Knecht, M. E. Schenkel, and G. S. Moschytz, "Neural network filters for speech enhancement," IEEE Transactions on Speech and Audio Processing, vol. 3, pp. 433-438, 1995.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[16] D. E. Rumelhart, B. Widrow, and M. A. Lehr, "The basic ideas in neural networks.," in Communications of the ACM. vol. 37, 1994, pp. 87-92.

[17] T. J. Moir, "Automatic variance control and variance estimation loops," Circuits Systems and Signal Processing, vol. 20, pp. 1-10, 2001.

[18] D. Sarkar, "Methods to speed up error back-propagation learning algorithm," ACM Computing Surveys, vol. 27, pp. 519-544, 1995.

A - IV

Paper Presented at the WCSP 2010 Conference

Speech Enhancement using Microphone Array Neural Switched Griffiths-Jim Beamformer

V. Yoganathan, T. J. Moir
School of Engineering and Advanced Technology, Massey University
Auckland, New Zealand
v.yoganathan@massey.ac.nz, t.j.moir@massey.ac.nz

Abstract—**There is a great need for speech enhancement in today's world due to the increasing demand for speech based applications. These applications vary from hearing-aids, hands-free telephony to speech controlled devices. The main goal is to minimize the interference from an acquired speech signal. The interference we considered here could be from any noise source such as competing speaker, radio, TV and so on. This paper proposes a solution to improve the current design of the switched Griffiths-Jim beamformer structure. It introduces an adaptive nonlinear neural network algorithm for the noise reduction section. The network topology used here is a partially connected three-layer feedforward neural network structure. The error backpropagation algorithm is used here as the learning algorithm. Comparison analysis of the traditional four channel linear beamformer and the proposed four-channel neural switched Griffiths-Jim beamformer structure is discussed here. They are both tested with different types of interference signal from the Noise-X database. All the experiments are conducted in real-world surrounding. The nonlinear approach introduced here shows remarkable improvement over the previous linear adaptive beamformer approach.**

Keywords-Nonlinear adaptive filter; noise reduction; Time delay neural network; multi-layer perceptron; Generalised sidelobe canceller

I. INTRODUCTION

Acquiring a clean speech signal in an adverse acoustic environment has created lot of challenges for many applications. Speech is often degraded by background noise originating from different sources such as radio, TV, computer fan or competing speakers, to name but a few. A lot of research has emerged through the years with the task of reducing the background noise from an acquired speech signal. These techniques vary from single microphone to multi-microphone systems; however, the underlying idea behind most popular algorithms is to use an adaptive filter to reduce the interference signal.

The adaptive noise cancelling structure was first introduced in the 70's to estimate the speech signal corrupted by background noise [1]. The reference microphone is positioned near the noise source and the primary microphone is positioned in such a way that it will acquire the speech signal as well as the noise signals. In an ideal situation, subtracting the reference signal from the primary signal will minimize the noise components while leaving the desired speech unchanged.

This structure is used largely for applications where the speech signal is isolated from the reference signal, and the noise signals are correlated in both channels. However, this traditional structure is unsuitable for many applications due to the difficulties in acquiring such a speech-free reference signal. This problem has been identified by many researchers. In particular, Griffiths and Jim developed a microphone array beamformer algorithm where the signals are added to get the new primary signal and they are subtracted to get the reference signal [2]. Their design has been known as the generalised sidelobe canceller or Griffiths-Jim beamformer algorithm. The two-channel structure of this beamformer is shown in Fig. 1. The main drawback with this method is that it requires the speech signal arriving at the microphones to be phase-aligned.

Van Compernolle improved this structure by adding another adaptive filter to align the speech signals and also used a Voice Activity Detector (VAD) to control the updates of these adaptive filters [3]. Fig. 2 illustrates the two-channel Switched Griffiths-Jim Beamformer (SGJB) structure. This beamformer structure has two sections: a beam-steering section and a noise canceller section. The beam-steering section is used to obtain a phase alignment between the microphone signals. The noise canceller section is used to reduce the unwanted background noise.

The first adaptive filter is only updated during a "speech and noise" period and the second adaptive filter is updated during a "noise alone" period. By controlling the update of the first section we can avoid the beam from steering to the noise source and by controlling the update of the second section speech cancellation can be avoided while achieving a maximum noise reduction.

A VAD is used to distinguish between a "speech and noise" period and a "noise alone" period. Clearly this is one major disadvantage of this kind of approach over other so-called unsupervised adaptive filtering approaches using say independent component analysis [4]. However, this is balanced by the need for a system that can potentially work in real-time. With such a simple energy based VAD, it has to be assumed that the desired speech has higher amplitude than any

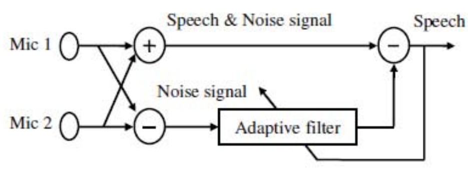


Figure 1. Two-channel Griffiths-Jim beamformer.

978-1-4244-7555-1/10/$26.00 ©2010 IEEE

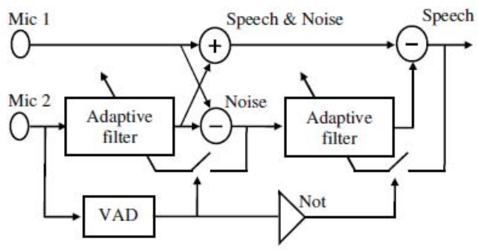


Figure 2. Two-channel Switched Griffiths-Jim Beamformer.

interference. This does not act as a major drawback since other, more sophisticated VAD approaches exist [5].

The beam-steering and the noise cancelling sections generally use a linear adaptive filter like the Least Mean Squares (LMS) algorithm. This area of research is exhaustive due to the ease of implementation with today's processing power and it has even been implemented in applications such as hearing aids [6]. The idea of a nonlinear adaptive filter was introduced originally by Gabor in 1954 using the Volterra series [7]. In recent years, many researchers have proven that using nonlinear adaptive filters for noise cancellation produces better results than the conventional linear adaptive filters [8]. Therefore, there has been a broad interest in this area. Examples of nonlinear adaptive filters include: Voltera filters, extended Kalman filters or neural filters, to name a few.

Amongst the existing nonlinear adaptive filters this paper concentrates on a Neural Network (NN) based adaptive noise cancelling filter which is both beneficial and practical to implement. The paper is organized as follows: section 2 summarizes pervious work on NN based noise reduction algorithms and introduces the Time Delay Neural Network (TDNN) algorithm. Section 3 presents and discusses the simulation results while section 4 presents the conclusion.

II. Artificial Neural Network for Noise Reduction

An introduction to the area of Artificial Neural Network (ANN) can be found in [9, 10]. In recent times, neural networks have been used for a range of signal processing applications such as system identification, pattern recognition, image processing, speech recognition and adaptive beamformer for antenna array applications [11]. Early work on neural networks can be referred to Widrow and Hoff; and their work lead to the first neuron structure called an Adaptive Linear Element (ADALINE) [12, 13]. It is a single neuron linear adaptive filtering structure based on the LMS algorithm.

Tamura and Waibel [14, 15] introduced a noise reduction algorithm based on a NN to map from noisy speech to a noise-free speech signal. They used a four-layered feedforward NN to reduce the background noise. This system was tested with an artificially generated noisy Japanese speech signal. The main drawback with this system is that it requires a clean speech signal to train the network. Although the output signal from the network was clear, the authors concluded that the processing did not improve the intelligibility of the speech signal.

Kencht [16] introduced the idea of combining nonlinear noise filtering and the perceptron based beamformer. Their research was only limited to a fixed beamformer structure. Hence, the weight values of the adaptive filter are estimated in the training period and it is used for the testing. Their research led to further work on nonlinear two-channel beamformers for speech enhancement in hearing-aid applications [17]. However, they only considered the GJBF structure (shown in Fig. 1) with the NN based noise reduction algorithm.

The current research presents a four-microphone based adaptive nonlinear neural SGJB. Research has shown that one hidden layer can approximate any measurable function given sufficient amount of hidden units [18]. Therefore a three-layer feedforward structure with one hidden layer will be used here. The TDNN algorithm is described below.

The error signal $e(k)$ is given by

$$e(k) = d(k) - y(k) \tag{1}$$

where $d(k)$ is the desired signal, $y(k)$ is the estimated signal from the adaptive filter.

Each neuron value at the hidden layer is calculated by

$$z(k) = f\left(\sum_{i=1}^{N}\left(w_i(k)\times x_i(k)\right) + b_i\right) \tag{2}$$

where $w(k)$ is the first weight vector at the hidden layer, $x(k)$ is the reference signal, $f(x)$ is the sigmoid function, b is the bias value at the hidden layer, N is the number of inputs to the neuron. The neuron transfer function is chosen to be the hyperbolic tangent nonlinear sigmoid function.

The output of the network is given by

$$y(k) = f\left(\sum_{j=1}^{P}\left(v_j(k)\times z_j(k)\right) + a\right) \tag{3}$$

where $v(k)$ is the second weight vector at the hidden layer, P is the number of neurons, a is the bias value at the output layer.

All the weights are initialized with a small random value between 1 and -1, and they are updated at each iteration using the error back propagation algorithm [19, 20]. This learning algorithm is most commonly used to train the NN. The new weight values are given by:

$$\mathrm{w}(k+1) = w(k) + \left(h1\times e(k)\times v(k)\times z'(k)\times x(k)\right) \tag{4}$$

$$v(k+1) = v(k) + \left(h2 \times e(k) \times y'(k) \times z(k)\right) \quad (5)$$

where $h1$ and $h2$ are the learning rate, $z'(k)$ is the derivative of the hidden layer neuron, and $y'(k)$ is the derivative of the output layer neuron.

III. RESULTS AND DISSCUSSION

Hardware products used for this experiment include: four omni-directional microphones, a pre-amplifier, National Instruments data acquisition card NI-DAQ 9233 and a personal computer for data analysis. The four-microphones are positioned in a linear array approximately 10 cm apart. A four-channel TDNN based SGJB and the traditional Normalised Least Mean Squares (NLMS) based SGJB have been simulated in the Labview programming language G. Fig. 3 shows the general structure of the four-channel neural SGJB algorithm. A VAD based on automatic variance control has been used with these beamformer's to control the adaptive filters [21].

The neural network structure implemented here is a three-layer feedforward network (one hidden layer) and it is a partially connected structure. The main advantage of using such a system compared to a fully connected neural network is to reduce the processing power (increasing the ability to do real-time processing). Fig. 4 shows the layout of the proposed nonlinear adaptive neural network filter. This NN structure has 8 input nodes, 5 hidden nodes and 1 output node. Each hidden neuron has 4 inputs from the input node and all of the hidden nodes are connected to the output layer.

A. *Experiment 1*

The experiment is set up as follows: the target speech signal is generated directly in-front of the microphones and the interference signals are generated at an angle to the microphones. Fig. 5 illustrates the experimental arrangement. The input signal contains the spoken words "One Two Three Four Five" and the interference signals are a babble noise and a motor noise. The babble noise is positioned at approximately 35° angle on the right and motor noise is positioned approximately 60° angle on the left. Both noise sources are equally spaced from the microphone. All the interference signals used were taken from the Noise-X database.

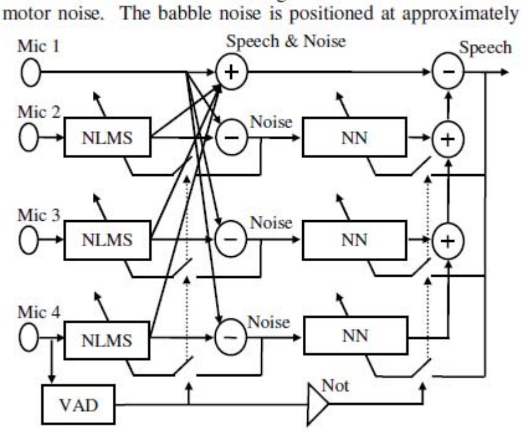


Figure 3. Four-microphone Switched Griffiths-Jim Beamformer.

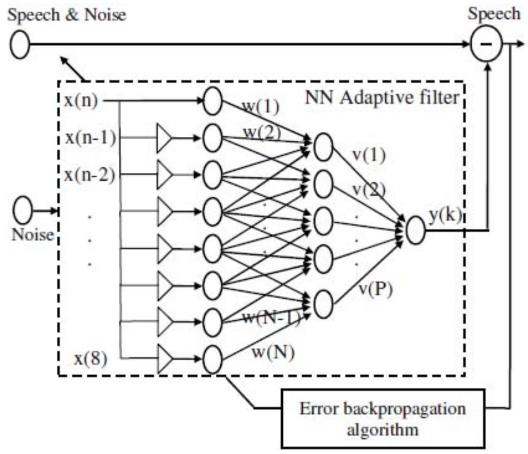


Figure 4. Partially connected three-layer feedforward ANC structure.

After carrying out several experiments the following parameters were chosen for the NN: the number of inputs to the network was set to 20, the number of neurons was set to 17, and the number of inputs to a neuron was set to 4. For the NLMS 85 was chosen as the number of weights (this is equivalent to the total number of weights used for the NN in both layers). The step-size value and the learning rates were chosen as 0.8. Initially, the NN structure was trained with a small section of the interference signal (3000 samples) for a single iteration.

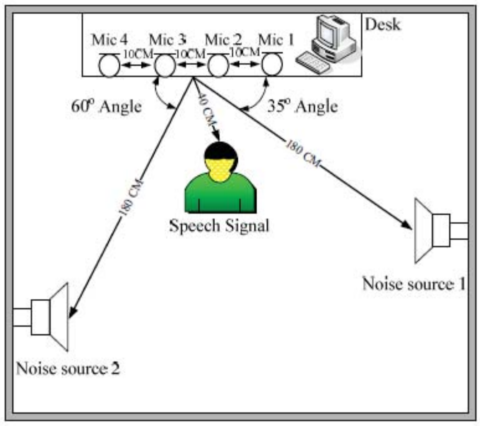


Figure 5. Experimental arrangement.

The quality of the output signal was measured by signal to interference ratio (SIR), and it was calculated by:

$$SIR = 10\log_{10}\left(\frac{Power\left(Speech + Noise\right)}{Power\left(Noise\right)}\right) \qquad (6)$$

Fig. 6 (a) shows the input signal from microphone 1 (b) and (c) shows the output signal from the NLMS and the NN based beamformer. The NLMS based SGJB showed a SIR improvement of 2.88 dB and the NN based SGJB showed a SIR improvement of 9.23 dB. Fig. 7 shows the comparison of the Mean Square Error (MSE) of the output signal of the NLMS and NN. This demonstrates that the NN converged more smoothly when compared to NLMS.

B. Experiment 2

For this experiment a similar setup to the previous experiment was used. The difference being that the distance between the motor noise and the microphones was increased (240 cm from the microphone).

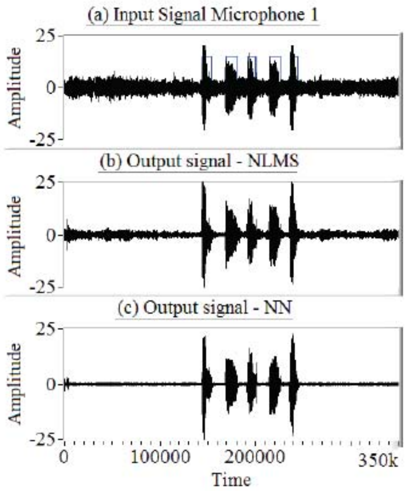


Figure 6. Experiment 1 results (a) input signal from microphone 1, (b) shows the output signal from the NLMS SGJB, and (c) the NN SGJB

As in experiment 1, the input signal contains the spoken words "One Two Three Four Five" and the interference signals contain babble noise and motor noise. Fig. 8 (a) presents the input signal from microphone 1 (b) and (c) show the output signal from the NLMS and the NN based beamformer. The NLMS beamformer showed a SIR improvement of 4.10 dB and the NN beamformer showed a SIR improvement of 10.67 dB. Fig. 9 shows the comparison of the MSE of the output signal of the NLMS and NN. A similar result to the previous experiment was seen here, where the NN converged more smoothly than the NLMS.

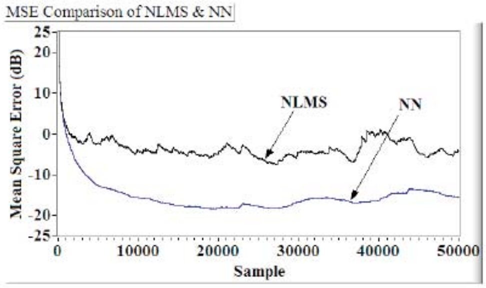


Figure 7. Comparison of the mean square error of the output signal of the NLMS and NN based SGJB

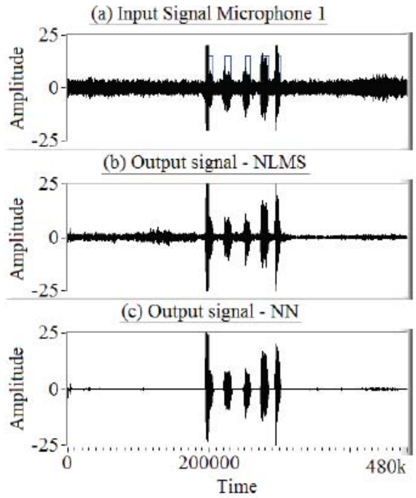


Figure 8. Experiment 2 results (a) input signal from microphone 1, (b) shows the output signal from the NLMS SGJB, and (c) the NN SGJB

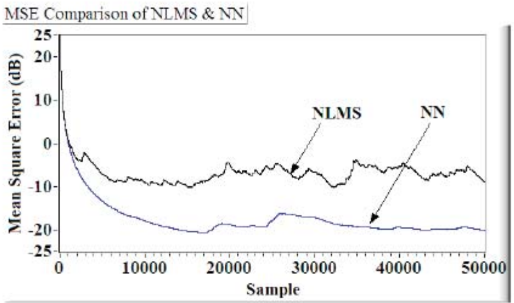


Figure 9. Comparison of the mean square error of the output signal of the NLMS and NN based SGJB

IV. Conclusion

This paper describes a new method for speech enhancement based on nonlinear neural network adaptive filtering. A four channel switched Griffiths-Jim beamformer structure was analysed with the traditional NLMS adaptive filter and the nonlinear neural network filter for the noise cancellation section. A partially connected three-layer time delay neural network structure was implemented for rapid real-time processing. The error back propagation algorithm is used as the learning algorithm to train the neural network. However, it is a slow gradient algorithm and it can be easily replaced with other algorithms such as the fast back propagation algorithm. The experimental results of the proposed algorithm show promising noise reduction improvement over the previous linear adaptive filter structure. However, the performance of the NN greatly depends on the size of the network structure. Future work in this area will consist of combining the neural network beamformer with a speech recognition system as a practical application to further analyze the performance of the algorithm.

References

[1] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proceedings of the IEEE, vol. 63, pp. 1692-1716, 1975.

[2] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," Antennas and Propagation, IEEE Transactions on, vol. 30, pp. 27-34, 1982.

[3] D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in International Conference on Acoustics, Speech, and Signal Processing, ICASSP-90., Albuquerque, NM 1990, pp. 833-836 vol.2.

[4] H. Sahlin and H. Broman, "Separation of real-world signals," Signal Processing, vol. 64, pp. 103-113, 1998.

[5] H. Agaiby and T. J. Moir, "Knowing the wheat from the weeds in noisy speech," in 5th European Conference on Speech Communication and Technology - EUROSPEECH, Rhodes, Greece, 1997, pp. 1119-1122.

[6] B. Widrow and F.-L. Luo, "Microphone arrays for hearing aids: An overview," Speech Communication, vol. 39, pp. 139-146, 2003.

[7] D. Gabor, "Communication theory and cybernetics," in IRE Transactions on Citcuit theory. vol. CT-1, 1954, pp. 19-31.

[8] M. Coker and D. Simkins, "A nonlinear adaptive noise canceller," 1980, pp. 470-473.

[9] M. Y. Kiang, "Neural Network," in Encyclopedia of information system. vol. 3, D. Bidgoli, Ed.: Elsevier Science (USA), 2003, pp. 303-315.

[10] K. Gurney, An introduction to neural networks. UK: UCL Press Limited 1997.

[11] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing: a review," Signal Processing, vol. 82, pp. 547-561, 2002.

[12] F. Liew Ban, A. Hussain, and S. A. Samad, "Speech enhancement by noise cancellation using neural network," in TENCON 2000. Proceedings, 2000, pp. 39-42 vol.1.

[13] M. Stella, D. Begusic, and M. Russo, "Adaptive Noise Cancellation Based on Neural Network," in Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006. International Conference on, 2006, pp. 306-309.

[14] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, 1988, pp. 553-556 vol.1.

[15] S. Tamura, "An analysis of a noise reduction neural network," in Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, 1989, pp. 2001-2004 vol.3.

[16] W. G. Knecht, "Nonlinear noise filtering and beamforming using the perceptron and its Volterra approximation," Speech and Audio Processing, IEEE Transactions on, vol. 2, pp. 55-62, 1994.

[17] W. G. Knecht, M. E. Schenkel, and G. S. Moschytz, "Neural network filters for speech enhancement," Speech and Audio Processing, IEEE Transactions on, vol. 3, pp. 433-438, 1995.

[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[19] D. E. Rumelhart, B. Widrow, and M. A. Lehr, "The basic ideas in neural networks.," in Communications of the ACM. vol. 37, 1994, pp. 87-92.

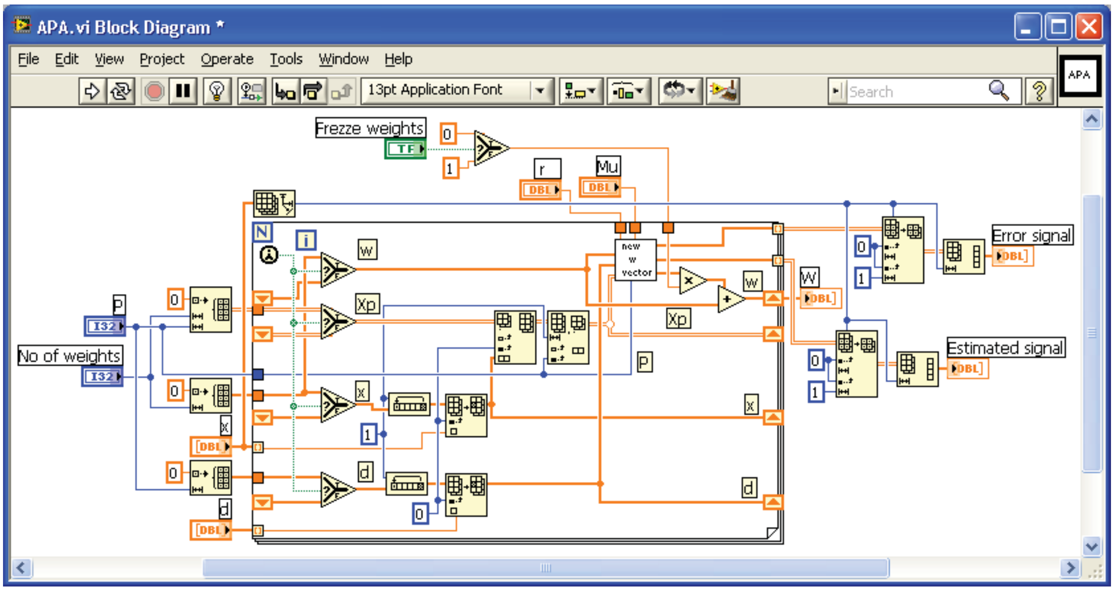
[20] B. Widrow and M. A. Lehr, Backpropagation and its applications. Mahwah: Lawrence Erlbaum Assoc Publ, 1993.

[21] T. J. Moir, "Automatic variance control and variance estimation loops," Circuits Systems and Signal Processing, vol. 20, pp. 1-10, 2001.

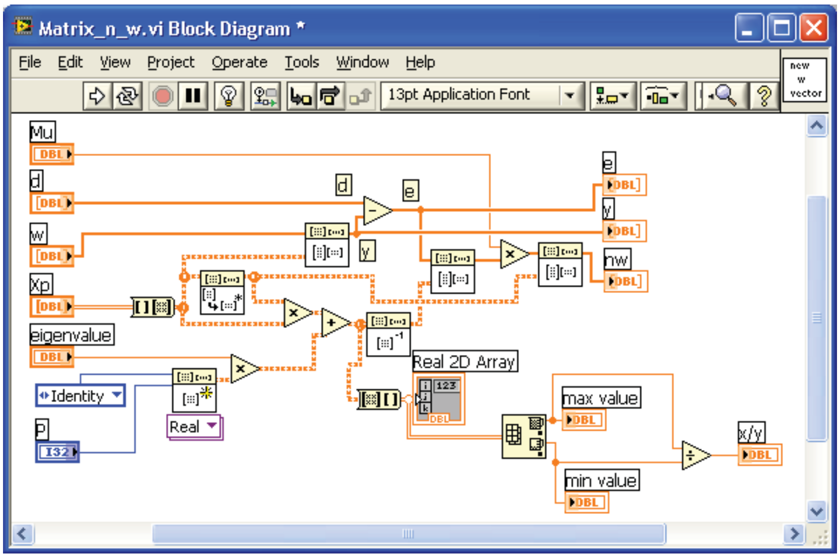
Appendix B

Labview Implementation of Other Adaptive Filter Algorithms

Labview implementation of the APA algorithm is given below.



Labview program for calculating the New W vector function is given below.



Labview implementation of the Volterra filter is given below: