PPU College of Engineering and Technology

The Home of Competent Engineers and Researchers

College of Engineering and Technology

Electrical and Computer Engineering Department

Communication and Electronics Engineering

Bachelor Thesis

Graduation Project

Voice Isolation Based on LMS Adaptive Filter

Project Team

Mohammad Abu-Fara Nour Eddin Riziqat

Project Supervisor

Dr. Ghandi Manasrah

Hebron- Palestine
2012

PALESTINE POLYTECHNIC UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Graduation Project

Voice Isolation Based on LMS Adaptive Filter

Project Team

Mohammad Abu-Fara Nour Eddin Riziqat

According to the system of the College of Engineering and Technology, and to the recommendation of the Project Supervisor, this project is presented to Electrical and Computer Engineering Department as a part of requirements of B.Sc. degree in Electrical Engineering – Communication and Electronic Engineering.

Project Supervisor signature
Dr. Ghandi Manasrah

…………………………...

Testing Group signature
Dr. Osama Ata Eng. Ahmad Qudaimat

………...…..……….. ………..…………….

Department Headmaster signature
Dr. Ramzi Qawasmeh

………………………………….

Hebron-Palestine

Dedication

To our fathers

To our mothers

To our brothers and sisters

To our teachers

To our friends

To everyone who helped us

To whom we love

We dedicate our humble effort

Project Team

Acknowledgment

We would like to thank all people who helped us and have a direct or indirect contributions in our project.

Our deepest gratitude goes to our supervisor, Dr. Ghandi Manasrah for his enlightening guidance, supports, encouragement and unending patience throughout the entire period of the project.

Special thanks to our parents, who always encourage, support and care for our throughout our life. We are also grateful to all our friends in the Communications engineering field at the Department of Electrical and Computer Engineering of Palestine Polytechnic University.

Abstract

This project aims to build a way portable (or easily made portable) headset system that effectively filters out the undesired noise while preserving the desired voice signal, allowing two people in close proximity to be able to communicate in a noisy environment. Our system implements a Least-Mean Squares (LMS) adaptive filter to identify and remove the noise from the desired audio signal before it is sent through wireless transmitter to a receiver. Our filter is effective in removing sufficiently loud non-white noise sources from 200 Hz to 4000 Hz, which is the range of most human created noise sources in human environments.

List of contents

Chapter One Introduction

Chapter Two Theoretical Background

Chapter Three Project Design

Chapter Four Design Details and Implementation

Chapter Five Results and Future Works

List of Tables

List of Figures

List of Abbreviations

DSP	: Digital Signal Processing
DFT	: Discrete Fourier Transformation
MDFT	: Modified Discrete Fourier Transformation
LMS	: Least Mean Square
ANC	: Active Noise Cancelation
NRF	: Noise Reconstruction Filter
ALE	: Adaptive Line Enhancer
S/N	: Signal to Noise ratio
SNIR	: Signal to Noise-Interference ratio
A/D	: Analog to Digital
D/A	: Digital to Analog
RLS	: Recursive Least Square
CCS	: Code Composer Studio
IDE	: Integrated Development Environment
DSK	: Digital Signal processing Kite
USB	: Universal Serial Bus
SDRAM	: Synchronous Dynamic RAM

1

Chapter One

Introduction

1.1 Introduction

1.2 Project Objectives

1.3 General Idea

1.4 Literature Review

1.5 Motivations

1.6 Time Plan

1.7 Estimated Cost

1.8 Project Risk Management

1.9 Project Contents

1.1 Introduction

Keeping up a conversation while surrounded by other conversations can be a challenge. I hear some words, while others ‘cut out’ due to overlapping sounds. I then try to guess the meaning based on the words that made it through, and most of the time I get most of it right enough to keep up a conversation that sounds reasonable. However, it is hard work and not enjoyable.

Conversations can be challenging even without noise for a number of reasons. With distracting and painful noise added and words missing, they are a pest. Phone calls tend to be sudden, the sound quality may be poor, the person on the other end tends to be a stranger with an unfamiliar voice and accent, random polite chat with strangers is expected, and there are no visual cues to hint how the person thinks and what she/he means.

In this project we will remove the background noise as much as possible.

1.2 Project Objectives

1.2.1 Project Features

1. Produce a significant gain in SNR.

To attain SNR, we would measure the absolute magnitude of noise from the noise-microphone and compare it to the noise component in the filtered signal. To do so, we can use a known quantity of noise (perhaps a sine wave or an audio file), and measure the presence of it in the output signal with the filter on and off. The logarithmic decease of the magnitude of this signal is proportional to the noise reduction in decibels.

2. Multiple non-interfering channels.

We need to ensure that the frequencies do not overlap, and that a single pair of frequencies can be uniquely selected without interfering with the other frequencies.

1.3 General Idea

Our project’s goal is to isolate the background noise from voice in noisy environment by designing a headset system that can selectively isolate voice from a sum of voices and background noise.

1.4 Literature Review

wireless voice isolation is the modern technique that uses to isolate the background unwanted voice, and there is some of research and papers about this field, there is some of it:

1. Speech Noise Reduction System Using MDFT Pair - Formulation of Noise Subtraction Coefficient.

In this paper, they propose a formulation of noise subtraction modified DFT pair. An input spectrum and a noise spectrum are coefficient using speech and noise levels. The noise detected at each sample using modified DFT and then noise subtraction coefficient is obtained automatically at each reduction is achieved by subtracting the detected noise spectrum from the input one.[1]

2. Application of Optimal settings of the LMS Adaptive filter for speech signal processing.

This paper describes a proposition of the method for optimal adjustment parameters of the adaptive filter with LMS algorithm in the practical application of suppression of additive noise in a speech signal for voice communication with the control system.[2]

3. The Multidimensional Characterization of Active Noise Cancellation headphone reception**.**

This paper explores the perceptual characteristics associated with the application of active noise cancelation (ANC) technologies to headphones.[3]

4. A New Noise Reduction System Based on ALE and Noise Reconstruction Filter.

The new noise reduction system uses two types of adaptive line enhancer (ALE) and noise reconstruction filter (NRF). First, two ALEs estimate the speech components.[4]

1.5 Motivations

The motivations of this project is to apply all the engineering knowledge that we have gained in communication engineering to produce a product that can benefit from it as much as possible by the people and take advantages of it. we ended up with a decision on the project, which allows wireless communication easy between two people in a noisy environment.

We plan to implement this by designing a headset system that can selectively isolate voice from a sum of voice and background noise. We chose this project because we feel that this is a common problem and because a solution like ours has not been implemented in this manner.

This is significantly different from noise-canceling headsets, as traditionally, these use destructive interference to remove noise from incoming signals. Our goal is to use basic audio DSP to remove a known noise pattern from an outgoing audio signal.

1.6 Time Plan

The following tables define the main tasks in the project introduction and project itself:

Table 1.1: Time scheduled table for project introduction

T1	Project Definition	**1 Week**
T2	Collecting Data	**11 Week**
T3	Analysis	**7 Week**
T4	Theoretical Calculation	**4 Week**
T5	Documentation	**10 Week**
T6	Prepare for Presentation	**2 Week**

The time of the project introduction is scheduled over 16 weeks, table 1.2 shows how the work was scheduled over this time:

Table 1.2: Time plan table for project introduction

Week Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
T1	▨															
T2		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨				
T3				▨	▨	▨	▨	▨	▨	▨						
T4						▨	▨	▨	▨							
T5					▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
T6															▨	▨

The following table defines the main tasks in the project:

Table 1.3: Time scheduled table for project

T1	Collecting Data	**3 Week**
T2	Implementations	**10 Week**
T3	Analysis	**5 Week**
T4	Building and Testing the System	**8 Week**
T5	Documentation	**10 Week**
T6	Prepare for Presentation	**2 Week**

The time of the project is scheduled over 16 weeks, table 1.4 shows how the work was scheduled over this time:

Table 1.4: Time plan table for project

Week Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
T1	▨	▨	▨													
T2			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨				
T3				▨	▨	▨	▨	▨								
T4							▨	▨	▨	▨	▨	▨	▨	▨		
T5					▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
T6															▨	▨

1.7 Estimated Costs

Table 1.5: Estimated Costs

Number	**Object**	**Cost\Unit ($)**	**Quantity**	**Cost ($)**
1	Digital signal processing Kit	**400**	**2**	**800**
5	Microphones	**25**	**2**	**50**
6	Headphone	**10**	**2**	**20**
				870 $

1.8 Project Risk Management

1. **Hardware**: This project contains many of hardware components such as DSP Kit and Microphones. We expect to faced some difficulties in obtaining some of these module.

2. **Software:** Matlab used in signal processing part of this project, and downloads this program on DSP kit to process voice signal practically. One of problems that we will faced is that the DSP kit didn't support all blocks used in Matlab that’s forcing us to build a system as DSP kit support.

3. **Team Risk:** No similarity in the study programs that's effected in connection team members. Also, lack of a suitable places for meetings between team members.

4. **Project Risk:** Team work faced difficulty in determine a good idea for a graduation project and many of ideas that offered on team was reject as it's not applicable here.

1.9 Project Contents

The project is divided up to five chapters, the chapters follow each other logically to get the complete idea about the project:

Chapter 1: Introduction

Discussed the definition of the project from multiple sides, its objectives and importance. Then taking about the time plan and the estimated cost of the system components needed to implement the designed system. Finally, the difficulties that the group had been faced .

Chapter 2: Theoretical Background

This chapter contains some of sections talking about some of signals (basics and processing).

First discussed the basics of the signals and information, discuss the signal processing methods, adaptive noise cancellation and reduction, the type of noise and the sampling theorem processing. Finally discuss the Least Mean Square (LMS) Adaptive Filter.

Chapter 3: Project Design

This chapter talks and discusses all the module needed in this project, then it discuss the design procedure, and drawing the block diagram for the project.

Chapter 4: Design Details and Implementation

This chapter is talking about the Design Details and Implementation of the project and block testing for each part in the system..

Chapter 5: Results and Future Works

This chapter shows the final results and future work.

2

Chapter Two

Theoretical Background

2.1 Introduction

This chapter contains a theoretical background related to the project, and contains many parts, it is starting by providing some details about signals and information, and defines the sampling theory and listing the important sampling rates, and typing some kinds of noise, finally it shows an overview to the least mean square (LMS) adaptive filter and describe its process which represents the core of this project.

2.2 Signals and Information

A signal is the variation of a quantity by which information is conveyed regarding the state, the characteristics, the composition, the trajectory, the evolution, the course of action or the intention of the information source. A signal is a means of conveying information regarding the state(s) of a variable.

The information conveyed in a signal may be used by humans or machines for communication, forecasting, decision-making, control, geophysical exploration, medical diagnosis, forensics, etc. The types of signals that signal processing deals with include textual data, audio, ultrasonic, subsonic, image, electromagnetic, medical, biological, financial and seismic signals.

Figure 2.1 illustrates a communication system composed of an information source, i[t],followed by a system, T[.], for transformation of the information into variation of a signal, x[t], a communication channel, h[.], for propagation of the signal from the transmitter to the receiver, additive channel noise, n(t), and a signal processing unit at the receiver for extraction of the information from the received signal.

In general, there is a mapping operation that maps the output, i[t], of an information source to the signal, x(t), that carries the information; this mapping operator may be denoted as T[.] and expressed as

$$x(t) = T[i(t)] \tag{1}$$

Where:

i[t]: Information source.
T[.]: System for transform information into variation signal.
x[t]: Variation of a signal.
h[.]: Communication channel.
n(t): Additive channel noise.

The information source i(t) is normally discrete-valued, whereas the signal x(t) that carries the information to a receiver may be continuous or discrete. For example, in multimedia communication the information from a computer, or any other digital communication device, is in the form of a sequence of binary numbers (ones and zeros), which would need to be transformed into voltage or current variations and modulated to the appropriate form for transmission in a communication channel over a physical link.

As a further example, in human speech communication the voice-generating mechanism provides a means for the speaker to map each discrete word into a distinct pattern of modulation of the acoustic vibrations of air that can propagate to the listener. To communicate a word, the speaker generates an acoustic signal realization of the word, x(t); this acoustic signal may be contaminated by ambient noise and/or distorted by a communication channel, or impaired by the speaking abnormalities of the talker, and received as the noisy, distorted and/or incomplete signal y(t), modeled as

$$y(t) = h[x(t)]+n(t) \qquad (2)$$

In addition to conveying the spoken word, the acoustic speech signal has the capacity to convey information on the prosody (i.e. pitch, intonation and stress patterns in pronunciation) of speech and the speaking characteristics, accent and emotional state of the talker. The listener extracts this information by processing the signal y(t).

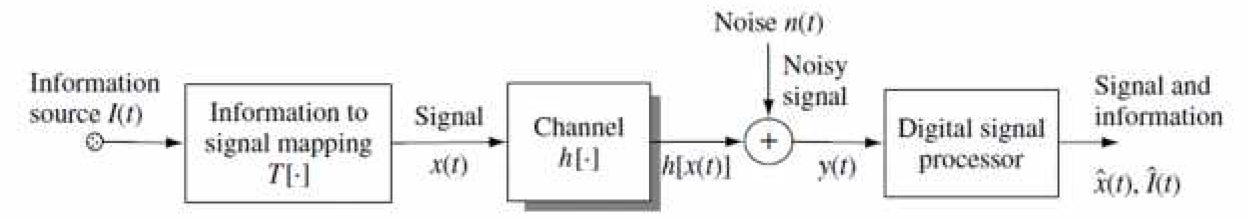


Figure 2.1 Illustration of a communication and signal processing system.

2.3 Signals Processing Methods

Signal processing methods have evolved in algorithmic complexity, aiming for optimal utilization of the information in order to achieve the best performance. In general the computational requirement of signal processing methods increases, often exponentially, with the algorithmic complexity. However, the implementation cost of advanced signal processing methods has been offset and made affordable by the consistent trend in recent years of a continuing increase in the performance, coupled with a simultaneous decrease in the cost, of signal processing hardware.

Digital signal processing algorithms can be categorized into one or a combination of four broad categories. These are transform-based signal processing, model-based signal processing, Bayesian statistical signal processing and neural networks, as illustrated in Figure 2.2. These methods are briefly described below.

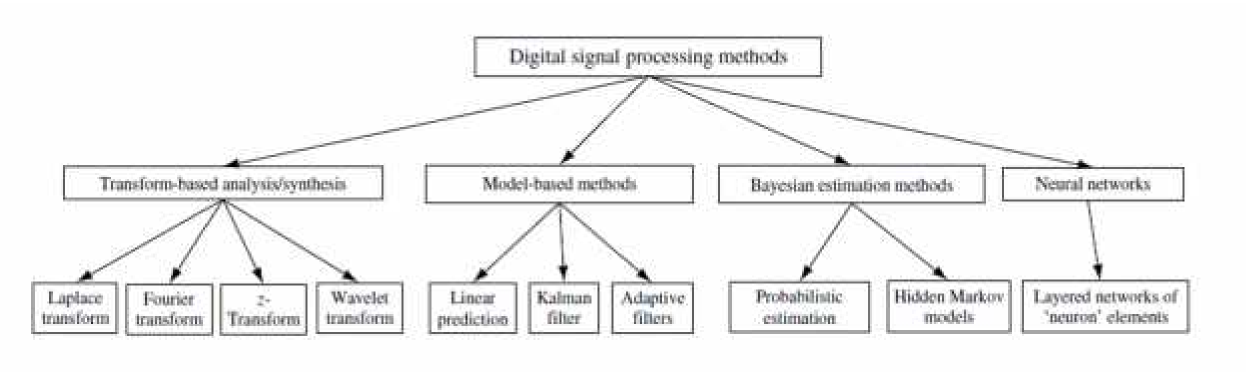


Figure 2.2 A broad categorization of some of the most commonly used signal processing methods.

2.3.1 Model-Based Signal Processing

Model-based signal processing methods utilize a parametric model of the signal generation process. The parametric model normally describes the predictable structures and the expected patterns in the signal process, and can be used to forecast the future values of a signal from its past trajectory. Model-based methods normally outperform nonparametric methods, since they utilize more information in the form of a model of the signal process. However, they can be sensitive to the deviations of a signal from the class of signals characterized by the model. Linear prediction models have facilitated the development of advanced signal processing methods for a wide range of applications such as low-bit-rate speech coding in cellular mobile telephony, digital video coding, high-resolution spectral analysis, radar signal processing and speech recognition.

2.4 Applications of Digital Signal Processing

In recent years, the development and commercial availability of increasingly powerful and affordable digital computers has been accompanied by the development of advanced digital signal processing algorithms for a wide variety of applications such as noise reduction, telecommunications, radar, sonar, video and audio signal processing, pattern recognition, geophysics explorations, data forecasting, and the processing of large databases for the identification, extraction and organization of unknown underlying structures and patterns.

Figure 2.3 shows a broad categorization of some digital signal processing (DSP) applications.

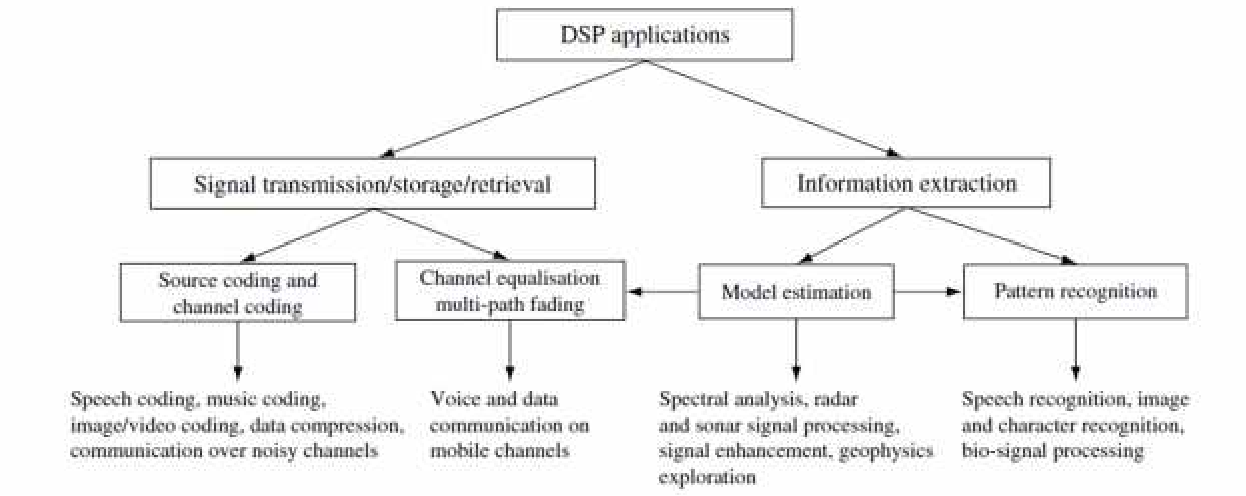

DSP applications
Signal transmission/storage/retrieval
Information extraction
Source coding and channel coding
Channel equalisation multi-path fading
Model estimation
Pattern recognition
Speech coding, music coding, image/video coding, data compression, communication over noisy channels
Voice and data communication on mobile channels
Spectral analysis, radar and sonar signal processing, signal enhancement, geophysics exploration
Speech recognition, image and character recognition, bio-signal processing

Figure 2.3 A classification of the applications of the digital signal processing.

2.4.1 Adaptive Noise Cancellation

In speech communication from a noisy acoustic environment such as a moving car or train, or over a noisy telephone channel, the speech signal is observed in an additive random noise.

In signal measurement systems the information-bearing signal is often contaminated by noise from its surrounding environment. The noisy observation, y(m), can be modeled as

$$y(m) = x(m)+n(m) \tag{3}$$

where x(m) and n(m) are the signal and the noise, and m is the discrete-time index. In some situations, for example when using a mobile telephone in a moving car, or when using a radio communication device in an aircraft cockpit, it may be possible to measure and estimate the instantaneous amplitude of the ambient noise using a directional microphone.

The signal, x(m), may then be recovered by subtraction of an estimate of the noise from the noisy signal.

Figure 2.4 shows a two-input adaptive noise cancellation system for enhancement of noisy speech. In this system a directional microphone takes as input the noisy signal x(m)+n(m),and a second directional microphone, positioned some distance away, measures the noise $\alpha n(m+\tau)$. The attenuation factor, α , and the time delay, τ , provide a rather over-simplified model of the effects of propagation of the noise to different positions in the space where the microphones are placed. The noise from the second microphone is processed by an adaptive digital filter to make it equal to the noise contaminating the speech signal, and then subtracted from the noisy signal to cancel out the noise. The adaptive noise canceller is more effective in cancelling out the low-frequency part of the noise, but generally suffers from then on-stationary character of the signals, and from the over-simplified assumption that a linear filter can model the diffusion and propagation of the noise sound in the space.

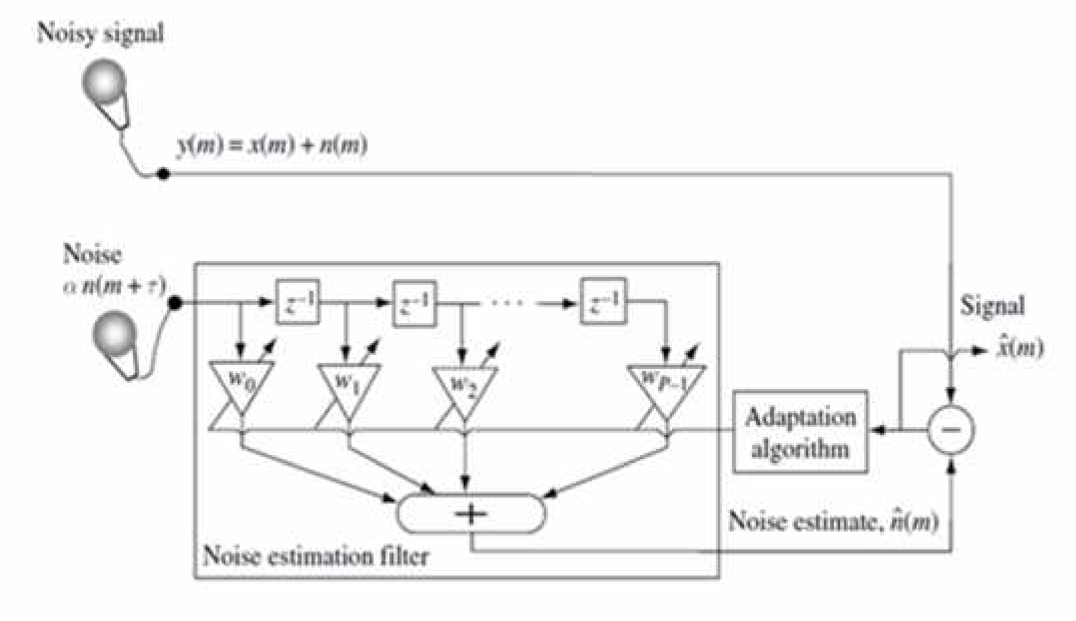


Figure 2.4 Configuration of a two-microphone adaptive noise canceller.

2.4.2 Adaptive Noise Reduction

In many applications, for example at the receiver of a telecommunication system, there is no access to the instantaneous value of the contaminating noise, and only the noisy signal is available. In such cases the noise cannot be cancelled out, but it may be reduced, in an average sense, using the statistics of the signal and the noise process. Figure 2.5 shows a bank of Wiener filters for reducing additive noise when only the noisy signal is available. The filter bank coefficients attenuate each noisy signal frequency in inverse proportion to the signal-to-noise ratio at that frequency.

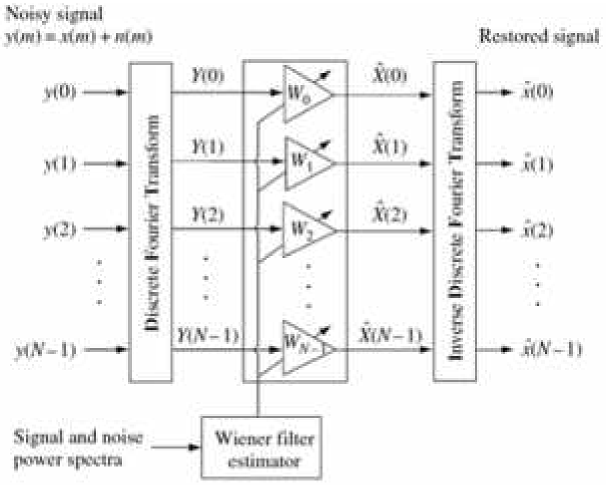


Figure 2.5 A frequency-domain Wiener filter for reducing additive noise.

2.5 Noise and Interference

Speech is a very basic way for humans to convey information to one another with a bandwidth of only 4kHz; speech can convey information with the emotion of a human voice. The speech signal has certain properties: It is a one-dimensional signal, with time as its independent variable, it is random in nature, it is non-stationary, i.e. the frequency spectrum is not constant in time. Although human beings have an audible frequency range of 20Hz to 20kHz, the human speech has significant frequency components only up to 4kHz. The most common problem in speech processing is the effect of interference noise in speech signals. Interference noise masks the speech signal and reduces its intelligibility. Interference noise can come from acoustical sources such as ventilation equipment, traffic, crowds and commonly, reverberation and echoes. It can also arise electronically from thermal noise, tape hiss or distortion products. If the sound system has unusually large peaks in its frequency response, the speech signal can even end up masking itself.

One relationship between the strength of the speech signal and the masking sound is called the signal-to-noise ratio, expressed in decibels. Ideally, the S/N ratio is greater than 0dB, indicating that the speech is louder than the noise. Just how much louder the speech needs to be in order to be understood varies with, among other things, the type and spectral content of the masking noise. The most uniformly effective mask is broadband noise. Although, narrow-band noise is less effective at masking speech than broadband noise, the degree of masking varies with frequency.

High-frequency noise masks only the consonants, and its effectiveness as a mask decreases as the noise gets louder. But low-frequency noise is a much more effective mask when the noise is louder than the speech signal, and at high sound pressure levels it masks both vowels and consonants.

In general, noise that affects the speech signals can be modeled using different types, White noise, Colored noise or Impulsive noise.

2.5.1 White Noise

White noise is defined as an uncorrelated random noise process with equal power at all frequencies. A random noise that has the same power at all frequencies in the range of $\pm\infty$ would necessarily need to have infinite power, and is therefore only a theoretical concept. However a band-limited noise process, with a flat spectrum covering the frequency range of a band-limited communication system, is to all intents and purposes from the point of view of the system a white noise process. For example, for an audio system with a bandwidth of 10kHz, any flat-spectrum audio noise with a bandwidth of equal to or greater than 10kHz looks like white noise.

2.5.1.1 Band-Limited White Noise

Pure white noise is a theoretical concept, since it would need to have infinite power to cover an infinite range of frequencies. Furthermore, a discrete-time signal by necessity has to be band-limited, with its highest frequency less than half the sampling rate. A more practical concept is band-limited white noise, defined as a noise with a flat spectrum in a limited bandwidth.

2.5.2 Colored Noise

Although the concept of white noise provides a reasonably realistic and mathematically convenient and useful approximation to some predominant noise processes encountered in telecommunications systems, many other noise processes are nonwhite. The term ‘colored noise’ refers to any broadband noise with a nonwhite spectrum. For example most audio frequency noise, such as the noise from moving cars, noise from computer fans, electric drill noise and people talking in the background, has a nonwhite predominantly low-frequency spectrum. Also, a white noise passing through a channel is ‘colored’ by the shape of the frequency response of the channel.

2.5.3 Impulsive Noise

Impulsive noise consists of random short-duration on/off noise pulses, caused by a variety of sources, such as switching noise, electromagnetic interference, adverse channel environment in a communication system, drop-outs or surface degradation of audio recordings, clicks from computer keyboards, etc.[5]

2.5.4 Interference

The interference happened when signal or more affect on the desired signal, and this caused attenuation in the amplitude and phase of the desired signal, wasted the information that contained, interference could happen in time or in frequency domain, in time by send data at same time for two station or more, and its happened in frequency by using the same frequency between stations ,we can consider two type of interference that are: intentional interference for example a base stations broadcast information on the same frequency band, or non-intentional that comes from two signal can't knowing the frequency of each other, in a communication system we can measure the quality of a signal, when its face interference by (Signal to Noise-Interference ratio) as the following [6]**:**

$$\mathrm{SNIR} = \frac{\text{Signal Power (P)}}{\text{Noise power } \ \mathrm{N} \ + \text{Interference power } \ \mathrm{I}} \tag{4}$$

2.6 Sampling Process

2.6.1 Definition and Theory

Sample rate, or sampling frequency (f_s) defines the number of samples per unit of time (usually seconds) taken from a continuous signal to make a discrete signal. For time-domain signals, the unit for sampling rate is hertz (inverse seconds, 1/s, s^{-1}), sometimes noted as Sa/s (samples per second). The inverse of the sampling frequency is the sampling period or sampling interval, which is the time between samples.

A microphone converts the acoustic signal into a corresponding analog electrical signal, and an analog-to-digital converter transforms that analog signal into a sampled digital form. The accuracy of the digital approximation of the analog signal depends on its resolution in time (the sampling rate) and its quantization, or resolution in amplitude (the number of bits used to represent each sample). For example, the audio recorded for storage on compact discs is sampled 44100 times per second and represented with 16 bits per sample

To convert an analog signal to a digital form it must first be band-limited and then sampled. Signals must be first filtered prior to sampling. Theoretically the maximum frequency that can be represented is half the sampling frequency. In practice a higher sample rate is used for non-ideal filters. The signal is now represented at multiples of the sampling period, T, as s(nT) which is also written as s(n), where n is an integer.

The input analog signal x(t) is first passed through an input filter (commonly called the anti-aliasing filter) whose function is to band limit the signal to below the Nyquist rate (one half the sampling frequency) to prevent aliasing.

The signal is then digitized by the A/D converter at a rate determined by the sample clock to produce x(n), the discrete-time input sequence. The system transfer function, H (z) ,is typically implemented in the time-domain using a linear difference equation. The sample output, ~ (n) is, then converted back into a continuous-time signal, y(t), by the D/A converter and output low-pass filter. The calculation of the output signal using a difference equation requires a multiply and accumulate operation. This is typically a single-cycle instruction on DSP chips.

Telephone speech is sampled at 8 kHz. 16 kHz is generally regarded as sufficient for speech recognition and synthesis. The audio standard is a sample rate of 44.1 kHz (Compact Disc) or 48 kHz (Digital Audio Tape) to represent frequencies up to 20 kHz.

Audio data is characterized by the following parameters, which correspond to settings of the A/D converter when the data was recorded. Naturally, the same settings must be used to play the data.

* Sampling rate (in samples per second), e.g.8000 or 44100

* Number of bits per sample, e.g.8 or16.

* Number of channels (1 for mono, 2 for stereo, etc.)

2.6.2 Popular Sampling Rates

Some sampling rates are more popular than others, for various reasons. Some recording hardware is restricted to (approximations of) some of these rates, some playback hardware has direct support for some. The popularity of divisors of common rates can be explained by the simplicity of clock frequency dividing circuits.[7]

Table 2.1: Popular Sampling Rates

Sampling Rate	Description and Use
8 KHz	Exactly 8000 samples/sec is a telephony standard that goes together with U-LAW (and also A-LAW) encoding. Some systems use an slightly different rate.
11 kHz	Either 11025, a quarter of the CD sampling rate, or half the Mac sampling rate.
16 KHz	Used by, e.g. the G.722 compression standard. Wideband frequency extension over standard telephone narrow band 8,000 Hz. Used in most modern VoIP and VVoIP communication products
22 kHz	One half the sampling rate of audio CDs; used for lower-quality PCM and MPEG audio and for audio analysis of low frequency energy. Suitable for digitizing early 20th century audio formats such as 78s
32 kHz	Used in digital radio, NICAM (Nearly Instantaneous Compoundable Audio Matrix [IBA/BREMA/BBC]) and other TV work, At least in the UK.
44.1 KHz	The CD sampling rate. DAT players recording digitally from CD also use this rate .
48 KHz	The DAT (Digital Audio Tape) sampling rate for domestic use.

2.7 Least Mean Square (LMS) Adaptive Filter

2.7.1 Introduction

Adaptive algorithms have become a mainstay in DSP. They are used in wide ranging applications including wireless channel estimation, radar guidance systems, acoustic echo cancellations and many others. An adaptive algorithm is used to estimate a time varying signal.

There are many adaptive algorithms like Recursive Least Square (RLS), Kalman filter, etc. but the most commonly used is the Least Mean Square (LMS) algorithm. LMS is a simple but powerful algorithm.

2.7.2 LMS Overview

The LMS algorithm was developed by Windrow and Hoff in 1959. The algorithm uses a gradient descent to estimate a time varying signal. The gradient descent method finds a minimum, if it exists, by taking steps in the direction negative of the gradient. It does so by adjusting the filter coefficients so as to minimize the error. The gradient is the del operator (partial derivative) and is applied to find the divergence of a function, which is the error with respect to the n^{th} coefficient in this case. The LMS algorithm approaches the minimum of a function to minimize error by taking the negative gradient of the function. A LMS algorithm can be implemented as shown in Figure 2.6.

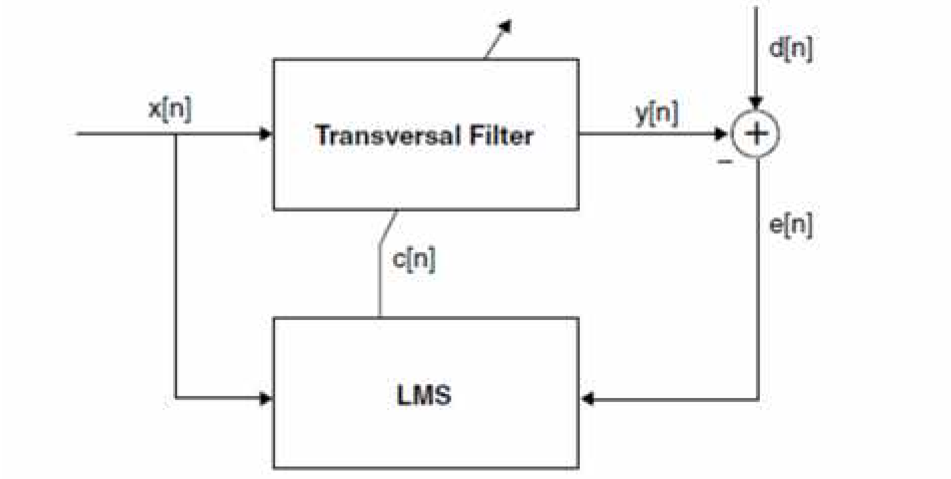


Figure 2.6 LMS Implementation Using FIR Filter

The desired signal d(n) is tracked by adjusting the filter coefficients c(n). The input reference signal x(n) is a known signal that is fed to the FIR filter. The difference between d(n) and y(n) is the error e(n). The error e(n) is then fed to the LMS algorithm to compute the filter coefficients c(n+1) to iteratively minimize the error. The following is the LMS equation to compute the FIR coefficients:

$$c(n+1) = c(n) + \mu * e(n) * x(n) \quad (5)$$

Where:

x[n]: Input reference signal.
c[n]: Filter coefficient.
y[n]: Transversal filter output.
d[n]: Desired signal.
e[n]: Error signal.
µ: Step size (For filter design).

The convergence time of the LMS algorithm depends on the step size µ. If µ is small, then it may take a long convergence time and this may defeat the purpose of using an LMS filter. However if µ is too large, the algorithm may never converge. The value of µ should be scientifically computed based on the effects the environment will have on d(n).

The LMS reference design has the following two main functional blocks:

1. FIR Filter.
2. LMS Algorithm.

2.7.3 FIR Filter

The FIR filter is implemented serially using a multiplier and an adder with a feedback as shown in the high level schematic in Figure 2.7. The FIR result is normalized to minimize saturation.

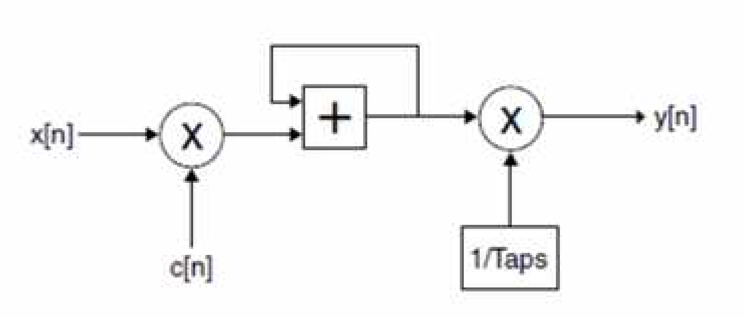


Figure 2.7 FIR Implementation

The LMS algorithm iteratively updates the coefficient and feeds it to the FIR filter. The FIR filter than uses this coefficient c(n) along with the input reference signal x(n) to generate the output y(n). The output y(n) is then subtracted from the desired signal d(n) to generate an error, which is used by the LMS algorithm to compute the next set of coefficients.

2.7.4 LMS Algorithm

The LMS algorithm is implemented as shown in Figure 2.8. The coefficients are calculated according to equation (5). The delay is necessary in the design to separate the current coefficients from the next set of coefficients. The delay size is nearly the same as the tap size.[8]

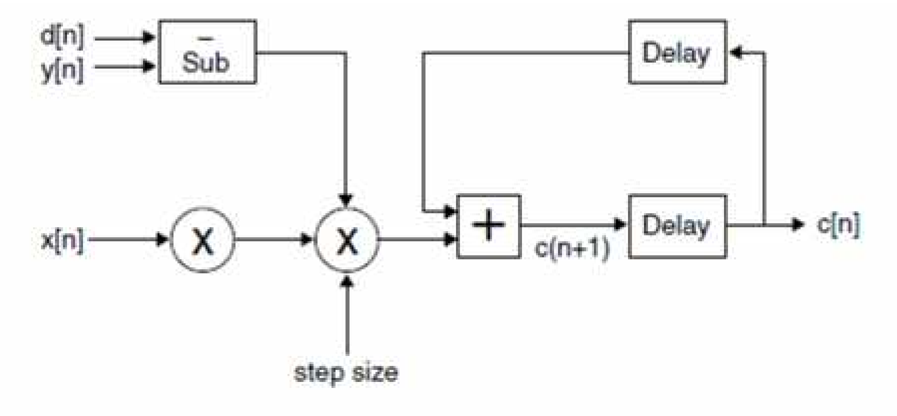


Figure 2.8 LMS Algorithm Implementation

3

Chapter Three

Project Design

3.1 Introduction

This chapter shows the overall voice isolation system in block diagrams, and the modules used by explaining the work of each one of them, then it describes in details the operations that are addressed in the bock diagram by determining the power needed and the input and output of each module, finally it shows how the least mean square adaptive filter (LMS) works.

3.2 Project Block Diagram

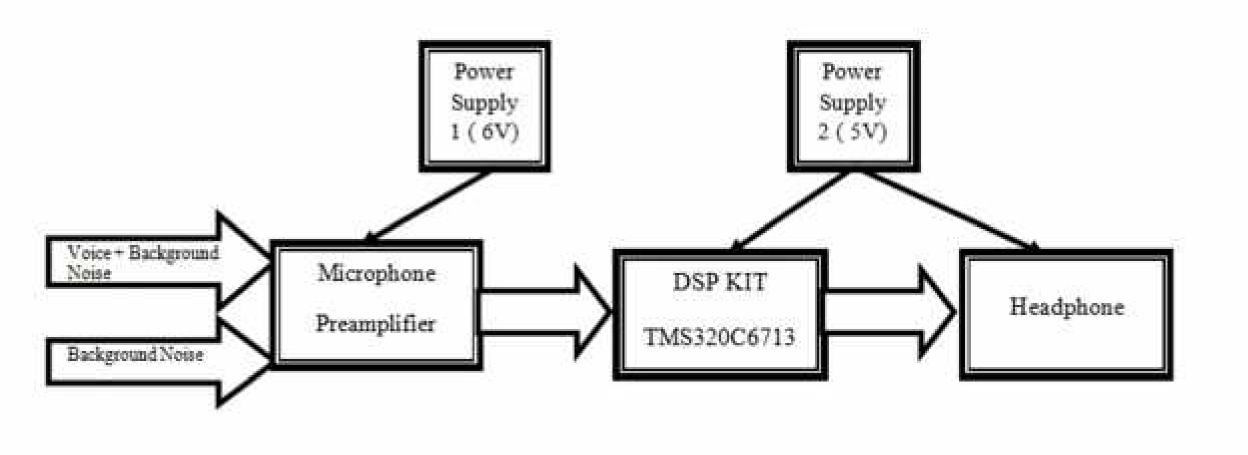


Figure 3.1 Block diagram of the project

3.2.1 Block Description

The microphone preamplifier takes the two voices from two microphones, then it loads the two different channels into one line to the DSP KIT, then the DSP KIT subtract the background noise using the LMS algorithm and finally send the clear voice to the headphones.

3.2.2 Components Description

1. **Power Supply 1**

The power supply 1 is a 6V voltage source powering the Preamplifier.

2. **Power Supply 2**

The power supply is a 5V voltage source powering the DSP KIT and headphone.

3. **Microphone Pre-Amplifier**

The microphone preamplifier module performs several functions on the signal before it reaches the processing board. It supplies power so that the microphones can drive an audio signal. It amplifies the audio signal from the microphones in levels to line in signals. Finally, it combines two separate audio signals so that they act as the two channels of a surround line in signal.

4. Digital Signal Processing (DSP) Chip

The DSP Chip takes in one input. The Line in input is combining of two signals in two channel from preamplifier. The DSP chip then converts both analog audio input signals into digital signals and then filters out the background noise by subtracting the first audio signal from the second signal. This digital signal is the filtered signal to be sent to the listener. The background noise is subtracted out, so only the voice from the speaker is left. After filtering, this digital signal is converted to analog signal by using DSP kit and then outputted to the headphone.

5. Headphones

The headphones take the analog signal from the DSP KIT and output the signal for the listener.

3.3 Project Design and Module Selection

The design of our project as a whole consist of many smaller modules with its own functions:

1. Power Supply Module 1.
2. Power Supply Module 2.
3. Microphone Array Module.
4. Microphone Preamplifier Module.
5. Digital Signal Processing (DSP) Module.
6. Headphones Module.

3.3.1 Power Supply Module 1

This module supplies power to the microphone pre-amplifier module which is equal to 6V DC as in the pre-amplifier data sheet.

3.3.2 Power Supply Module 2

This module supplies power to the DSP chip module and the headphones which is equal to 5V DC.

3.3.3 Microphone Array Module

The microphone module consists of two microphones. The two microphones serve different purposes. One microphone picks up the signal of the noise while the other microphone picks up the signal of the voice as well as the noise.

The microphone array consists of two Microphones. Electrets microphones create a signal by allowing audio waves to modulate a fixed voltage maintained across a capacitor. This voltage is then sent as the microphone signal after going through a small amplifier within the actual microphone. The microphones require a power supply in order to operate the internal amplifier. Microphone placement was important to ensure that the signal integrity was maintained and that the voice signal was not compromised by having the noise only microphone pick up the voice signal as well. The cupping devices also helped to isolate the voice signal to only the voice and noise microphone.

3.3.4 Microphone Preamplifier Module

The microphone preamplifier module performs several functions on the signal before it reaches the processing board. It supplies power so that the microphones can drive an audio signal. It amplifies the audio signal from mike in levels to line in signals. Finally, it combines two separate audio signals so that they act as the two channels of a surround line in signal.

One of the obstacles that we faced was the fact that the DSP board has only two audio inputs: one mike in and one line in. We previously assumed that we can insert microphone plugs to both inputs and manipulate the signal using the DSP board. That was not a problem, First, the microphones required a power supply to drive a signal. Secondly, the microphone signals are much smaller in amplitude than line in signals. So it is needed to use only one of the both DSP Kit inputs "mike in" or "Line in", so we used line in.

3.3.5 Digital Signal Processing (DSP) Module

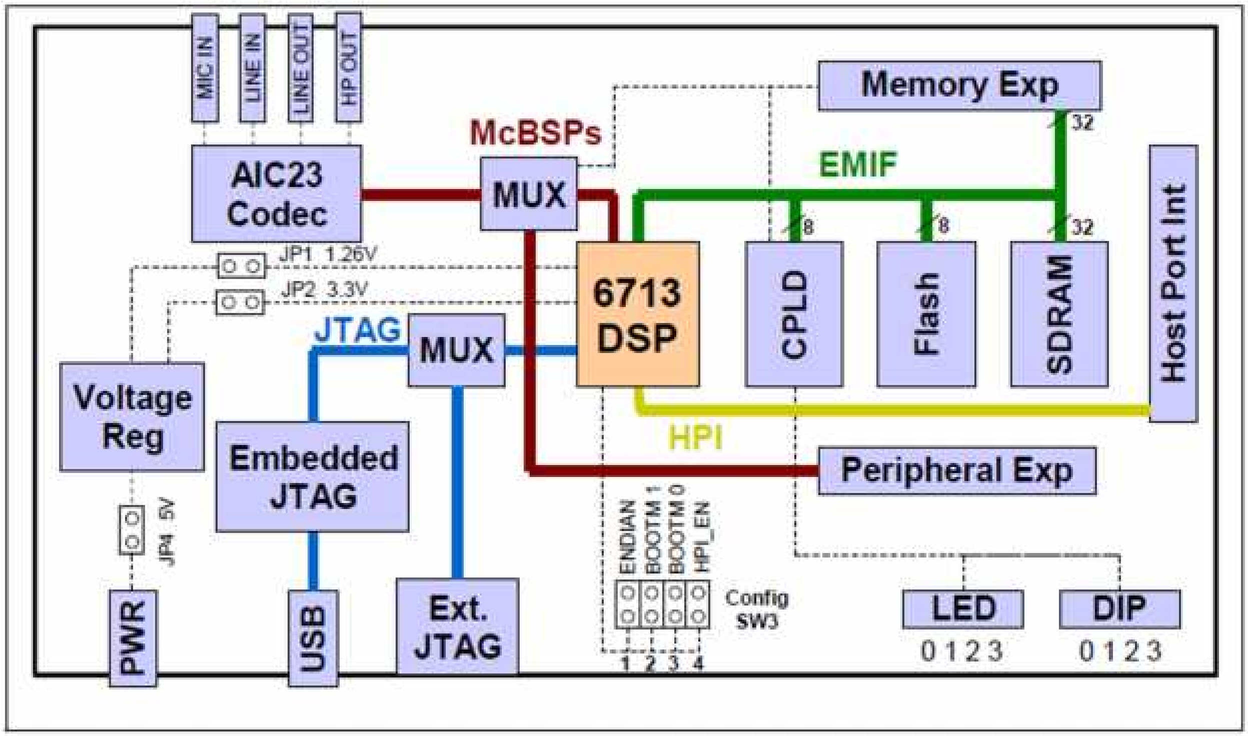


Figure 3.2 Block Diagram of C6713 DSK Board

The Digital Signal Processing (DSP) module formed the heart of our audio analysis and manipulation. It received two single-channel analog audio inputs, performed all audio functions pertaining to our filter, and outputted the sanitized audio to our headphones.

Digital Signal Processing Starter Kit comprises a small circuit board containing a TMS320C6713 floating point digital signal processor, and a TLV320AIC23 analog interface circuit (codec), can connects to a host PC via a USB port. PC software in the form of Code Composer Studio (CCS), is provided in order to enable software written in C, assembly language or MATLAM Simulink, to be compiled and/or assembled, linked, and downloaded to run on the DSK.

DSP processer's very suitable for the audio-frequency range. Common applications using these processors have been for frequencies from 0 to 96 kHz. Speech can be sampled at 8 kHz (the rate at which samples are acquired), which implies that each value sampled is acquired at a rate of 1/(8 kHz) or 0.125ms.

3.3.5.1 DSK Support Tools

1. A Texas Instruments DSP starter kit (DSK). The DSK package includes:

a. Code Composer Studio (CCS), which provides the necessary software support tools. CCS provides an integrated development environment (IDE), bringing together the C compiler, assembler, linker, debugger.

b. A circuit board (TMS320C6713 DSK) , its containing a digital signal processor, and a 16 - bit stereo codec for analog signal input and output.

c. A universal Serial bus (USB), its cable that connects the DSK board to a PC.

d. A +5 V universal power supply for the DSK board.

2. A PC. The DSK board connects to the USB port of the PC through the USB cable included with the DSK package.

3. An oscilloscope, spectrum analyzer, signal generator, headphones, microphone, and speakers.[10]

3.3.5.2 Audio Connectors

The C6713 DSK has 4 audio connectors. They are described in the following sections.

3.3.5.2.1 Microphone Connector

The input is a 3.5 mm. stereo jack. Both inputs are connected to the microphone so it is monaural. The signals on the plug are shown in the figure below.

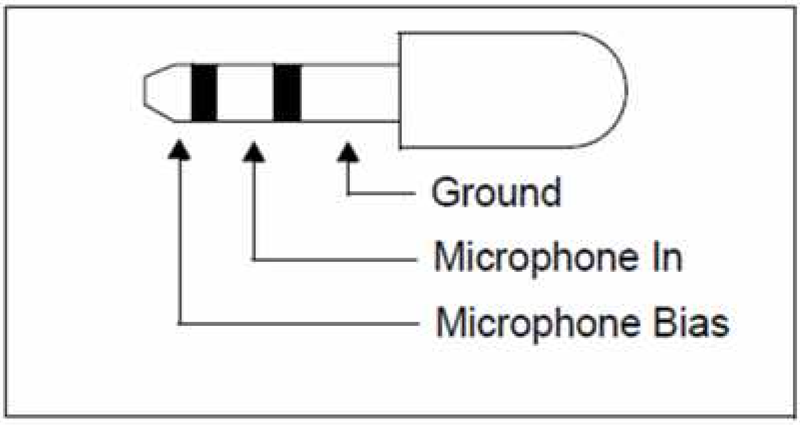


Figure 3.3 Microphone Stereo Jack

3.3.5.2.2 Audio Line In Connector

The audio line in is a stereo input. The input connector is a 3.5 mm stereo jack. The signals on the mating plug are shown in the figure below.

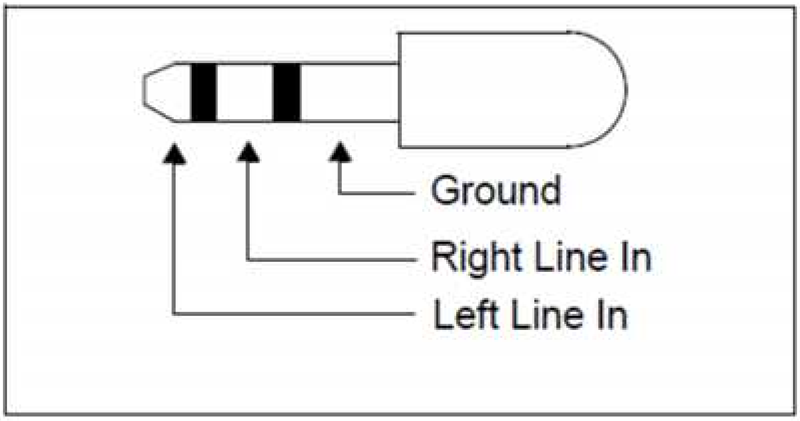


Figure 3.4 Audio Line In Stereo Jack

3.3.5.2.3 Audio Line Out Connector

The audio line out is a stereo output. The output connector is a 3.5 mm stereo jack. The signals on the mating plug are shown in the figure below.

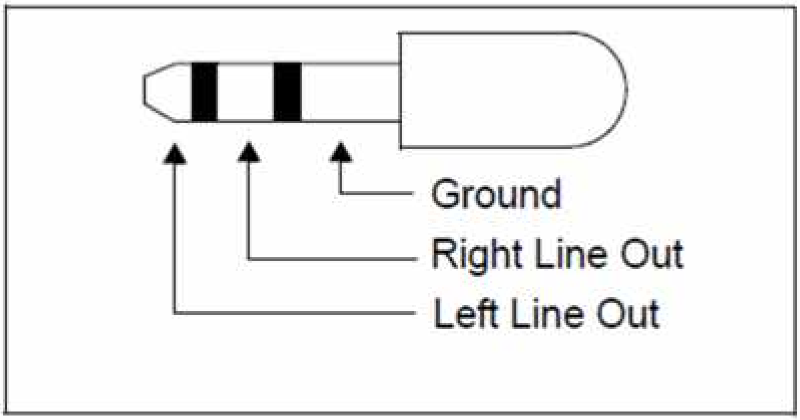


Figure 3.5 Audio Line Out Stereo Jack

3.3.5.2.4 Headphone Connector

Connector J4 is a headphone/speaker jack. It can drive standard headphones or a high impedance speaker directly. The standard 3.5 mm jack is shown in the figure below.

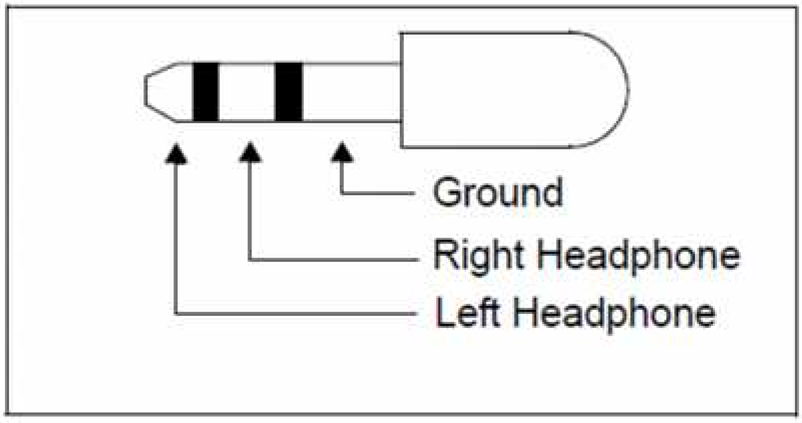


Figure 3.6 Headphone Jack

3.3.5.3 Code Composer Studio:

CCS is a powerful integrated development environment that provides a useful transition between a high-level (C or assembly) DSP program and an on-board machine language program. CCS consists of a set of software tools, and libraries for developing DSP programs, compiling and linking them into machine code, and writing them into memory on the DSP chip, and on-board external memory. It also contains diagnostic tools, for analyzing and tracing algorithms, as they are being implemented on-board. We will use CCS to develop, compile, and link programs that will be downloaded from a PC to DSP hardware.[11]

3.3.5.4 DSP Module Design

The DSP module consisted of several separate yet interdependent components. These were the hardware that performed the actual processing, the software environment that ran and controlled the hardware, and the audio software itself that ran the signal analysis and manipulation. During our design of the overall DSP Module, several revisions were made to each component.

3.3.5.4.1 Hardware/Software - Texas Instruments TMS320C6713 DSP Starter Kit

The 6713 DSK housed the core processing unit as well as all the analog audio inputs and outputs required for the DSP module to interface with the headphones. The starter kit is shown as Figure above and includes the core DSP chip, as well as interfaces for various DSP applications.

3.3.5.4.1.1 Matlab DSP Toolkits

Matlab Simulink, is a visual development platform which provides programmers an environment to create a blocks that take in signal inputs, perform functions, and provide signal outputs. The DSP Toolkit for matlab abstracted away many of the low-level development details (such as C and assembly syntax) allowing us to focus on the development of the filter itself.[11]

3.3.5.4.1.2 Filter Design—Least Mean Square (LMS) Adaptive Filter

The design of the filter is essentially the core of our project. We used matlab to design this filter. The purpose of this need to show that we could use matlab to perform the necessary computations to the audio streams.

- Simple white noise: (Noise signal)
- Desired Signal
- Signals added: (Noise) + (Desired Signal)
- Noise subtracted: (Noise) + (Desired Signal) – (Noise) = (Desired Signal)

Two adaptive filter designs were readily available to us, the Least Mean Squares (LMS) filter, and the Recursive Least Squares (RLS) filter. We chose to pursue the LMS filter as it was simpler and less computationally intensive than the RLS filter. Both adaptive filters follow the same general outline, the filters are set at a certain finite order; they start arbitrary values for their coefficients; and the filter adjusts them at each sample based on a reference signal.[12]

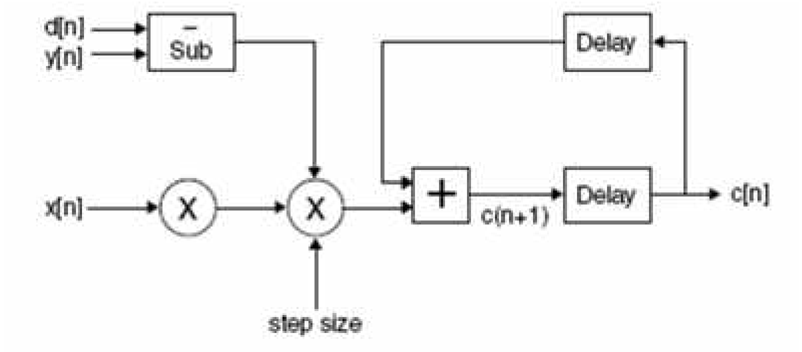


Figure 3.7 LMS Algorithm Implementation

3.3.5.4.1.3 Reference Design Features

The LMS reference design can be configured to meet user specifications. This can be achieved by setting the following user-configurable parameters:

- Input data bit width
- Output data bit width
- Tap size
- Step size

Since the FIR filter is implemented serially, the Tap size will dictate the number of clock cycles (cc) required to generate the FIR output. For example, a 128 TAP Fir will take 128 cc to produce each result. This implies that the input sampling rate should be selected as follows:

m = Clock speed of the reference design , (Tap size * Input sampling rate). Therefore, the clock speed of the reference design should be at least Tap size * Input sampling rate. Using the following specifications as an example;

Input sampling rate = 8KHz Tap size = 256

The reference design should at least be clocked at 256 x 8 = 2.048 MHz for m =1.
A higher value for m will result in better estimation.[8]

4

Chapter Four

Design Details and Implementation

4.1 Introduction

4.2 Implementation using MATLAB

4.3 Downloading on DSP KIT

4.1 Introduction

Through our studying and researching in the introduction of project course, we were able to make a complete image about the system which will be built.

First of all , we need to build the core of our project which is the voice isolation by using the LMS adaptive filter using matlab simulink and download it into DSP kit.

4.2 Implementation using MATLAB

We have used matlab simulink to implement our project before building it to make sure the system is working or not, and to determine the LMS adaptive filter parameters.

After we built the simulating system, the desired results for voice isolation (separation) have occurred.

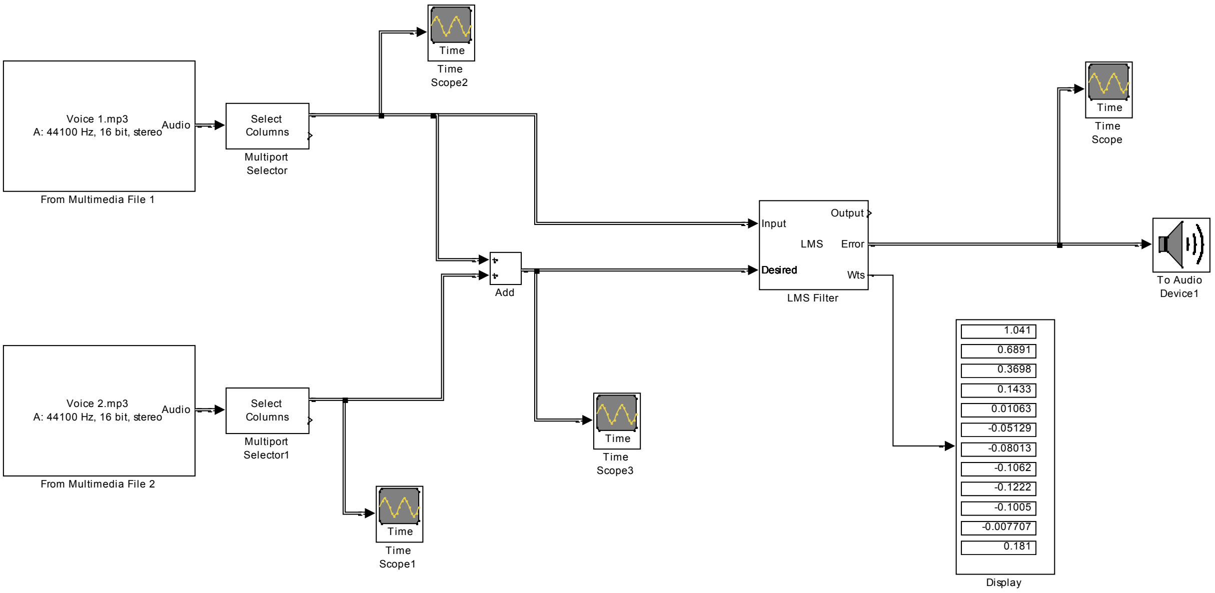


Figure 4.1 Voice Isolation System Implementation

As shown in the figure above, two different voices are supported to two different multiport selectors, the voice from Multimedia File1 is the background noise it is supported to the first channel of the first multiport selector, and the voice from Multimedia File 2 which is supposed to be the desired voice that we need to get rid of is supported to the first channel of the second multiport selector. Then the voice from Multimedia File 1 is entered directly to the input port of the LMS filter, and the sum of the two voices from Multimedia File 1 and from Multimedia File 2 to the desired port of the LMS filter. In order to get the filtered signal the headphone must be directly connected to the error port.

To ensure the design is clearly understood, the used matlab blocks is explained in details in the following sections showing parameters of each one in its own interface.

4.2.1 From Multimedia File block

The From Multimedia File block reads audio frames, video frames, or both from a multimedia file. The block imports data from the file into a simulink model.

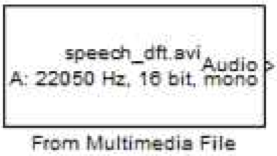
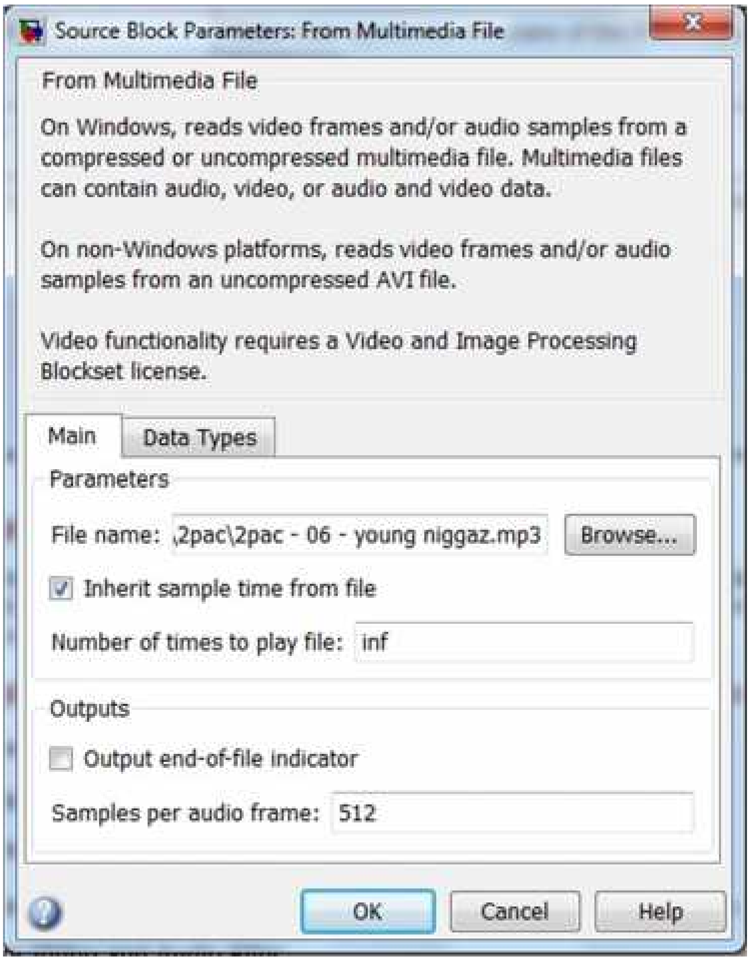


Figure 4.2 From Multimedia File Block


Source Block Parameters: From Multimedia File

From Multimedia File

On Windows, reads video frames and/or audio samples from a compressed or uncompressed multimedia file. Multimedia files can contain audio, video, or audio and video data.

On non-Windows platforms, reads video frames and/or audio samples from an uncompressed AVI file.

Video functionality requires a Video and Image Processing Blockset license.

Main | Data Types

Parameters

File name: ,2pac\2pac - 06 - young niggaz.mp3 [Browse...]

☑ Inherit sample time from file

Number of times to play file: inf

Outputs

□ Output end-of-file indicator

Samples per audio frame: 512

Figure 4.3 From Multimedia File Source Block Parameters

4.2.2 Multiport Selector block

The Multiport Selector block extracts multiple subsets of rows or columns from *M*-by-*N* input matrix *u*, and propagates each new sub-matrix to a distinct output port. A length-*M* 1-D vector input is treated as an *M*-by-1 matrix.

The Indices to output parameter is a cell array whose *k*th cell contains a one-dimensional indexing expression specifying the subset of input rows or columns to be propagated to the *k*th output port. The total number of cells in the array determines the number of output ports on the block.

When the Select parameter is set to Rows, the specified one-dimensional indices are used to select matrix rows, and all elements on the chosen rows are included. When the Select parameter is set to Columns, the specified one-dimensional indices are used to select matrix columns, and all elements on the chosen columns are included. A given input row or column can appear any number of times in any of the outputs, or not at all.

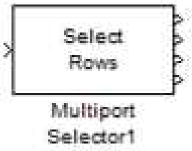
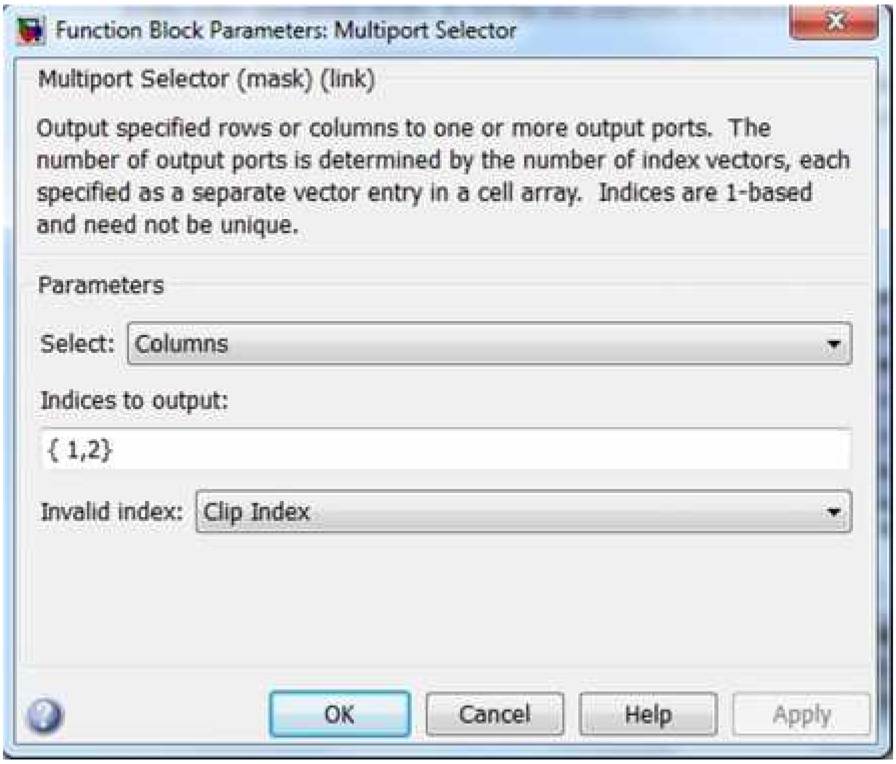


Figure 4.4 Multiport Selector Block



Multiport Selector (mask) (link)

Output specified rows or columns to one or more output ports. The number of output ports is determined by the number of index vectors, each specified as a separate vector entry in a cell array. Indices are 1-based and need not be unique.

Parameters

Select: Columns

Indices to output:

{ 1,2}

Invalid index: Clip Index

Figure 4.5 Multiport Selector Function Block Parameters

You can use the Multiport Selector block in the Indexing library to extract the individual channels of a multichannel frame-based signal. These signals from single-channel frame-based signals that have the same frame rate and size of the multichannel signal. The figure below is a graphical representation of this process.

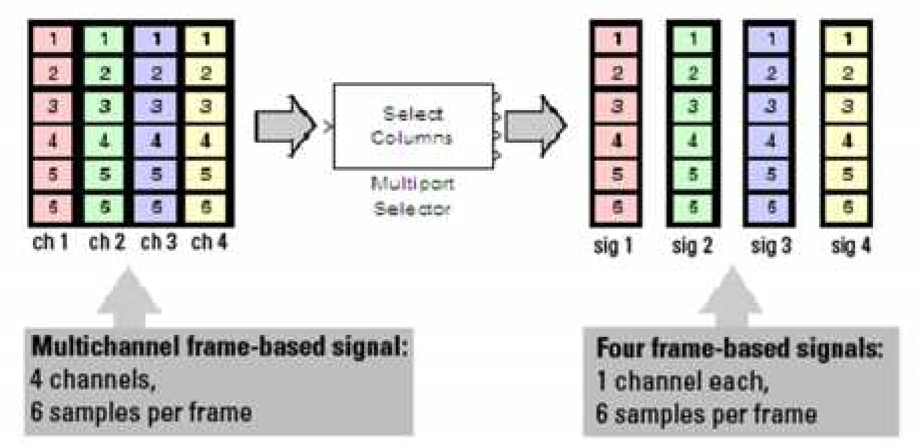


Figure 4.6 Multiport Selector Work

4.2.3 LMS Filter

The LMS Filter block can implement an adaptive FIR filter using five different algorithms. The block estimates the filter weights, or coefficients, needed to minimize the error, *e(n),* between the output signal *y(n)* and the desired signal, *d(n).* Connect the signal you want to filter to the Input port. This input signal can be a sample-based scalar or a single-channel frame-based signal. Connect the desired signal to the Desired port. The desired signal must have the same data type, frame status, complexity, and dimensions as the input signal. The Output port outputs the filtered input signal, which is the estimate of the desired signal. The output of the Output port has the same frame status as the input signal. The Error port outputs the result of subtracting the output signal from the desired signal

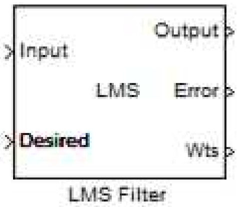
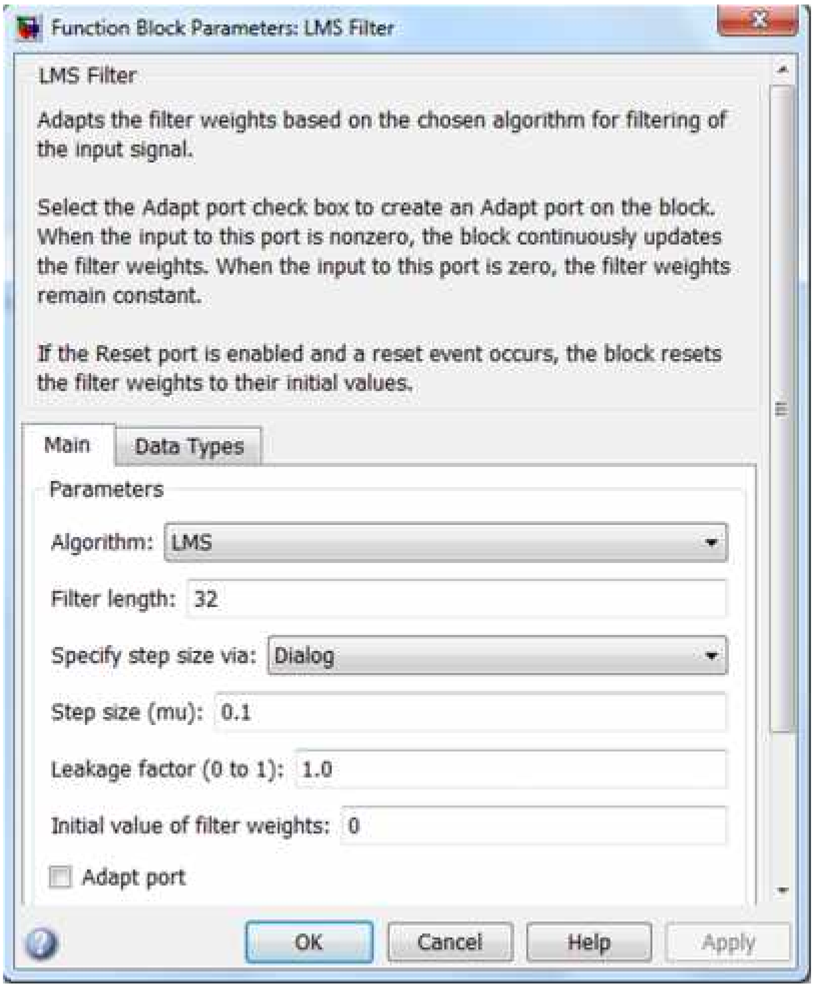


Figure 4.7 LMS Filter Block



Function Block Parameters: LMS Filter

LMS Filter

Adapts the filter weights based on the chosen algorithm for filtering of the input signal.

Select the Adapt port check box to create an Adapt port on the block. When the input to this port is nonzero, the block continuously updates the filter weights. When the input to this port is zero, the filter weights remain constant.

If the Reset port is enabled and a reset event occurs, the block resets the filter weights to their initial values.

Parameters

Algorithm: LMS

Filter length: 32

Specify step size via: Dialog

Step size (mu): 0.1

Leakage factor (0 to 1): 1.0

Initial value of filter weights: 0

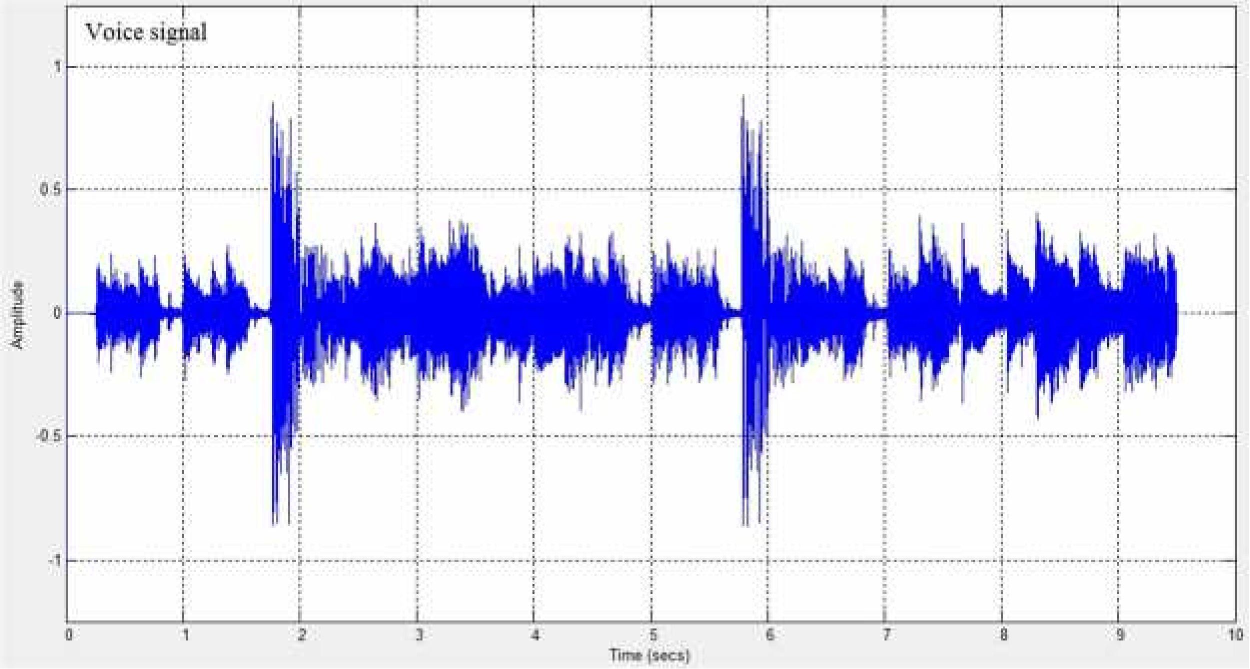
□ Adapt port

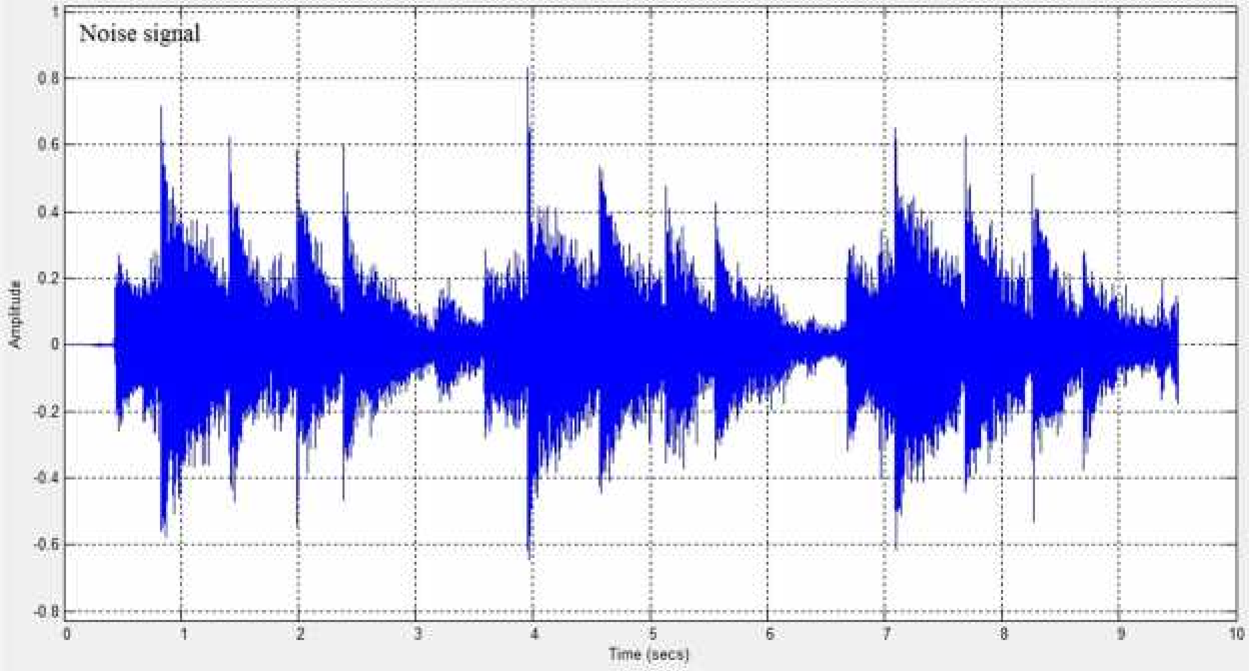
Figure 4.8 LMS Filter Function Block Parameters

The LMS filter parameters like the filter length, step size and the initial value of filter weight are changed to have the desired filtered output voice as shown in the above interface photo.

After running this system in matlab simulink the wanted filtered voice has isolated from any background noises as it is aimed to.

As shown in the following figure 4.9 the first image represents the voice or the speech signal, the second image represents the background noise wanted to get rid of, the third image is the sum of the two voices the speech or voice signal and the background noise signal. The last image shows the output of the LMS filter from the error port after separate the background noise from the voice and outputs the cleared voice signal. It is clear from the third image that the LMS filter successfully cancel and remove the background noise after minimizing the difference between the speech or the voice signal and the background signal with suitable step size and filter length (tap size) parameters and as it appears in the third image the output voice has a high amplitude more than the entered voice signal.


Voice signal
Amplitude
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10
Time (secs)


Noise signal
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8
9
10
Time (secs)

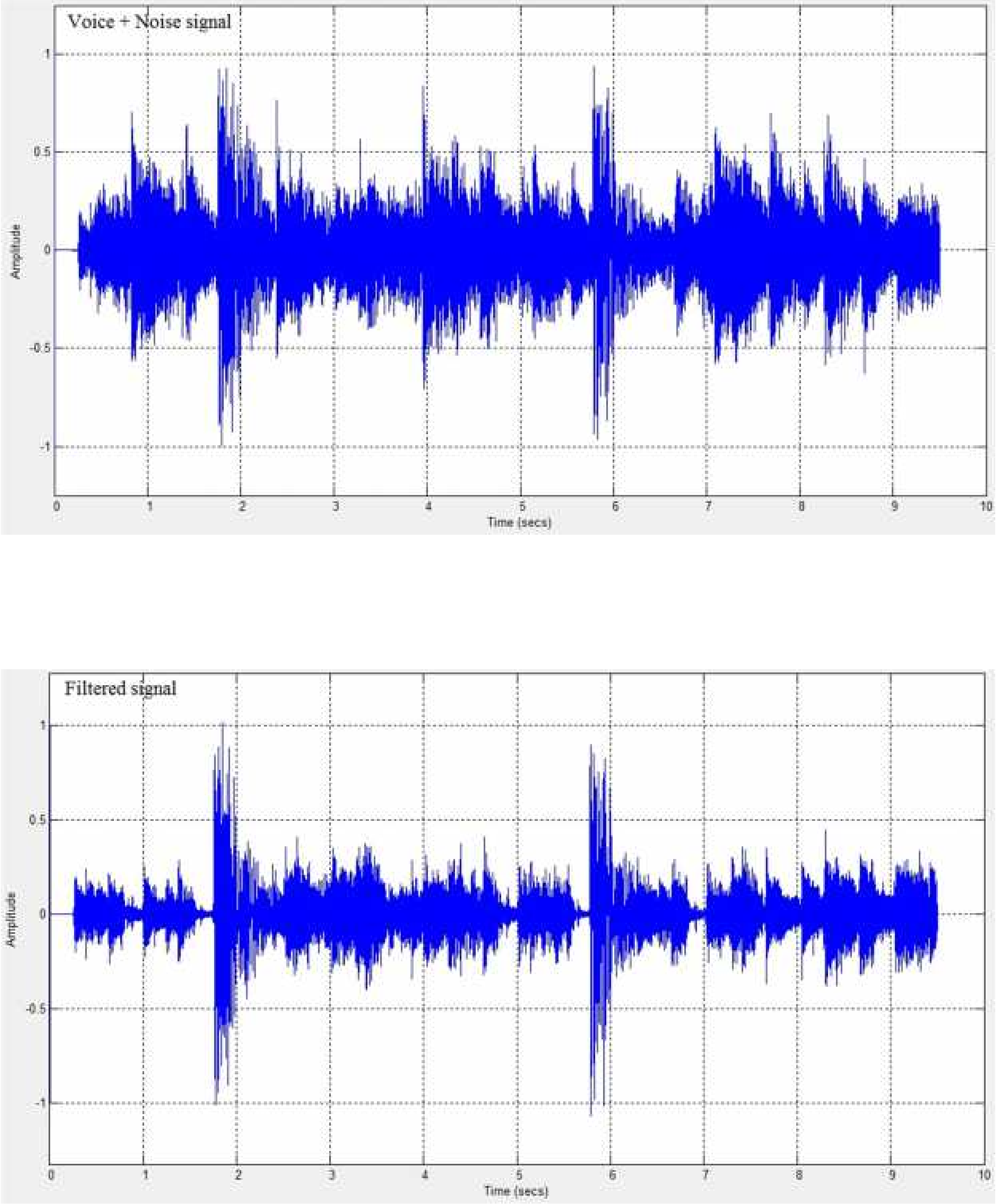


Figure 4.9 Results of the designed Voice Isolation System

From previous pictures we see that there is absolutely no different between the voice signal and the filtered signal, the error is very low, so we will subtract the filtered signal from the voice signal to see and hear what the different between them, the following figure show how the subtract is happened.

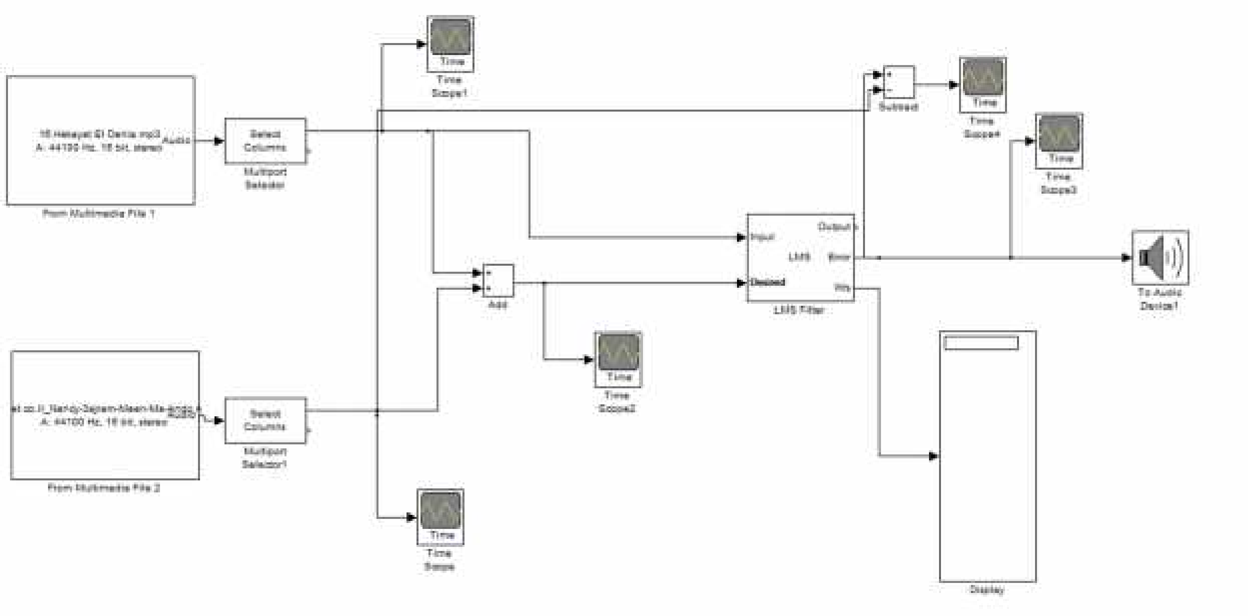
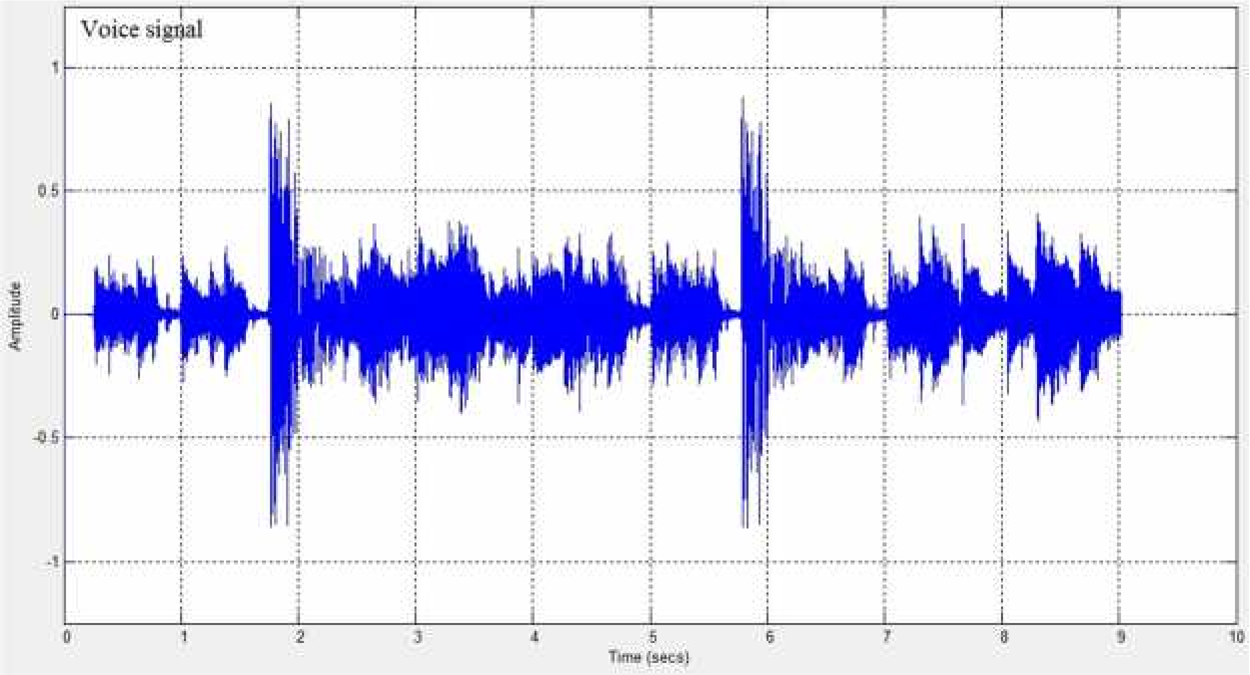


Figure 4.10 Difference signal

The following pictures showing to us the voice signal, filtered signal and the difference between them, we see the difference is a complete signal but if we hear the difference signal we will find it nothing.



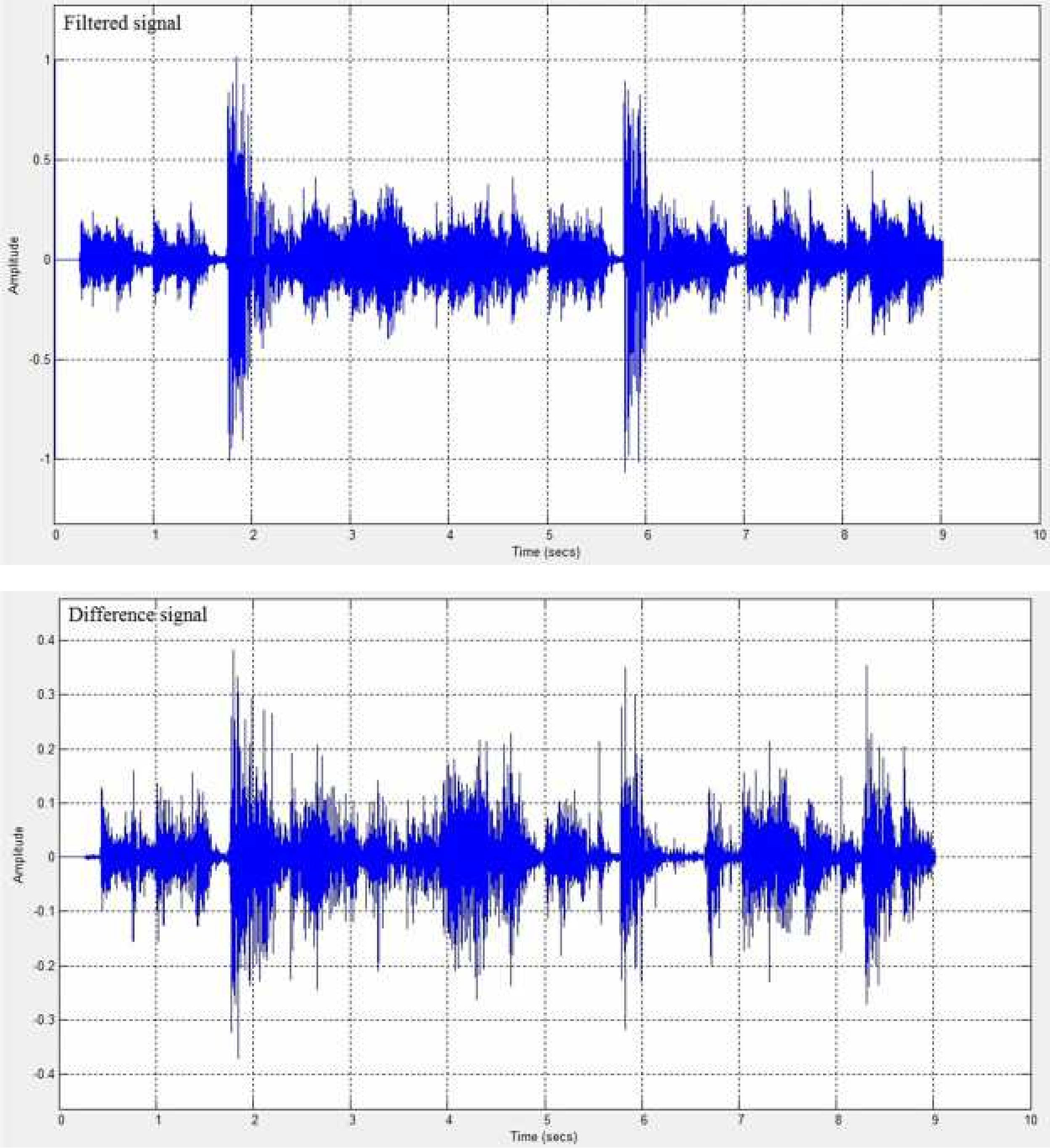


Figure 4.11 Results of the designed difference signal

4.3 Downloading on DSP KIT

The system is to be downloaded on DSP KIT must be different and have some changes rather than the system that implemented using matlab simulink. The system must be supported by an analog to digital (ADC) converter to convert the voice to digital signal that the DSP KIT works with it, and digital to analog (DAC) converter to convert the output of the DSP KIT again to analog signal. The following figure 4.10 shows the system downloaded into the DSP KIT.

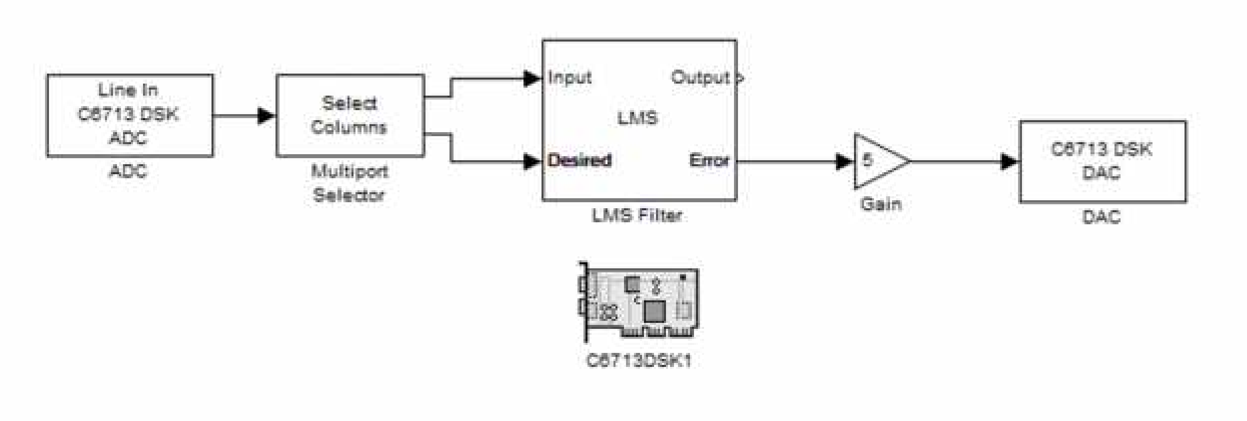


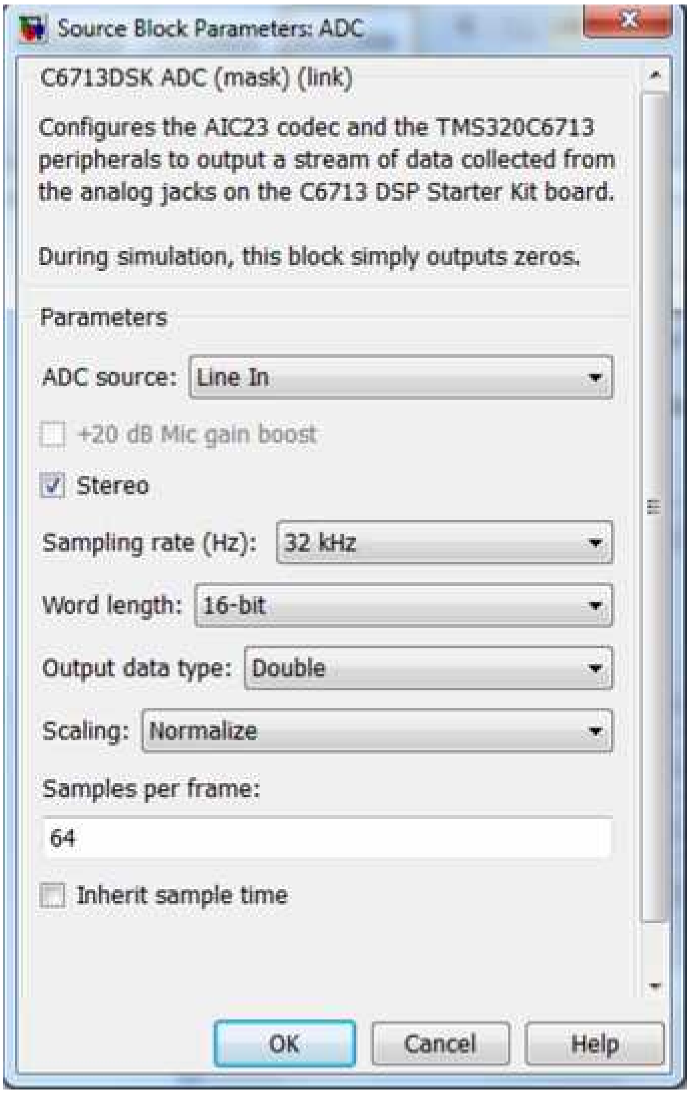
Figure 4.12 Voice Isolation System

4.3.1 Line In C6713 DSK ADC

Use the C6713 DSK ADC (analog-to-digital converter) block to capture and digitize analog signals from external sources, such as signal generators, frequency generators or audio devices. Placing an C6713 DSK ADC block in our Simulink block diagram lets us use the audio coder-decoder module (codec) on the C6713 DSK to convert an analog input signal to a digital signal for the digital signal processor.

Line In
C6713 DSK
ADC
ADC

Figure 4.13 Analog to Digital Converter


Source Block Parameters: ADC

C6713DSK ADC (mask) (link)

Configures the AIC23 codec and the TMS320C6713 peripherals to output a stream of data collected from the analog jacks on the C6713 DSP Starter Kit board.

During simulation, this block simply outputs zeros.

Parameters

ADC source: Line In

□ +20 dB Mic gain boost

☑ Stereo

Sampling rate (Hz): 32 kHz

Word length: 16-bit

Output data type: Double

Scaling: Normalize

Samples per frame: 64

□ Inherit sample time

Figure 4.14 ADC Source Block Parameters

You can select one of three input sources from the ADC source list:

- Line In - the codec accepts input from the line in connector (LINE IN) on the board's mounting bracket.
- Mic In - the codec accepts input from the microphone connector (MIC IN) on the board mounting bracket.

The block uses frame-based processing of inputs, buffering the input data into frames at the specified samples per frame rate. In Simulink software, the block puts monaural data into an N-element column vector. Stereo data input forms an N-by-2 matrix with N data values and two stereo channels (left and right).

When the samples per frame setting is more than one, each frame of data is either the N-element vector (monaural input) or N-by-2 matrix (stereo input). For monaural input, the elements in each frame form the column vector of input audio data. In the stereo format, the frame is the matrix of audio data represented by the matrix rows and columns, the rows are the audio data samples and the columns are the left and right audio channels.

When you select Mic for ADC source, you can select the +20 dB Mic gain boost check box to add 20 dB to the microphone input signal before the codec digitizes the signal.

4.3.2 Gain

The Gain block multiplies the input by a constant value (gain). The input and the gain can each be a scalar, vector, or matrix.

You specify the value of the gain in the Gain parameter. The Multiplication parameter lets you specify element-wise or matrix multiplication. For matrix multiplication, this parameter also lets you indicate the order of the multiplicands.

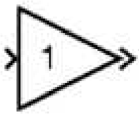
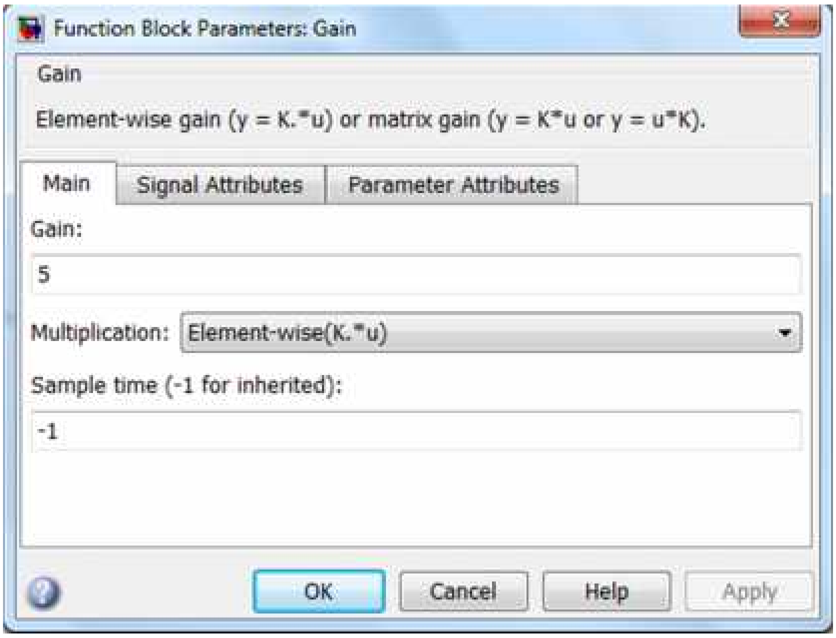


Figure 4.15 Gain



Function Block Parameters: Gain

Gain

Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).

Gain: 5

Multiplication: Element-wise(K.*u)

Sample time (-1 for inherited): -1

Figure 4.16 Gain Function Block Parameters

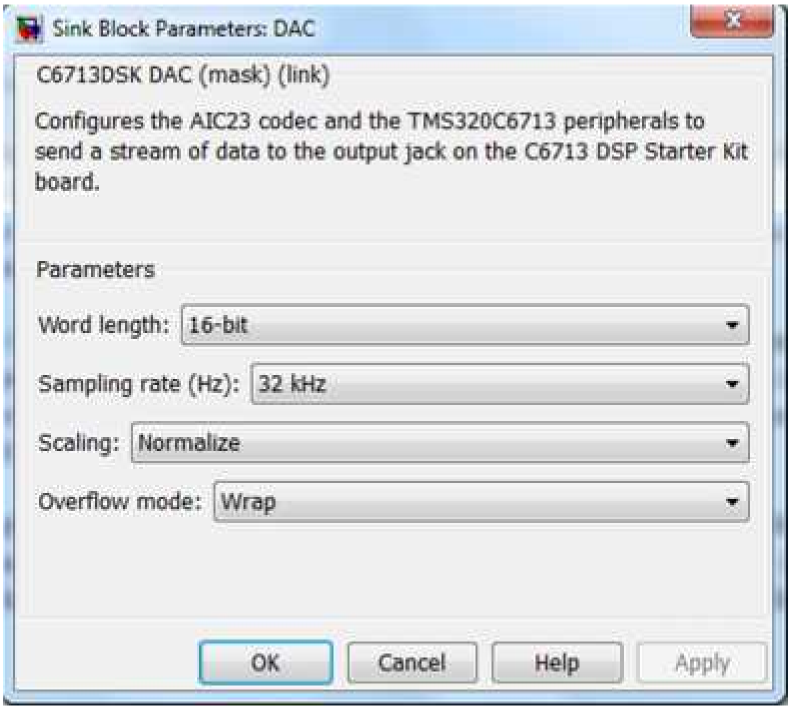
4.3.3 C6713 DSK DAC

Adding the C6713 DSK DAC (digital-to-analog converter) block to your Simulink model lets you connect an analog signal to the analog output jack on the C6713 DSK. When you add the C6713 DSK DAC block, the digital signal received by the codec is converted to an analog signal and sent to the output jack.

The input on the C6713DSK DAC block takes [Nx1] and [Nx2] signals. The AIC23 audio codec on the C6713DSK board always outputs stereo samples, even though it accepts both mono [Nx1] and stereo [Nx2] signals. If the input is a mono signal with dimension [Nx1], the block outputs the same signal on both the left and right channels. If the input is a stereo signal with dimension [Nx2], each of the N samples are output separately through the left and right channels.

C6713 DSK
DAC
DAC

Figure 4.17 Digital to Analog Converter



C6713DSK DAC (mask) (link)

Configures the AIC23 codec and the TMS320C6713 peripherals to send a stream of data to the output jack on the C6713 DSP Starter Kit board.

Parameters

Word length: 16-bit

Sampling rate (Hz): 32 kHz

Scaling: Normalize

Overflow mode: Wrap

Figure 4.18 DAC Block Parameters

Sets the DAC to interpret the input data word length. Without this setting, the DAC cannot convert the digital data to analog correctly. The value defaults to 16 bits, with options of 20, 24, and 32 bits. Select the word length to match the ADC setting.

5

Chapter Five

Results and Future Works

5.1 Introduction

5.2 Results of the Practical Implementation

5.3 Future Works

5.1 Introduction

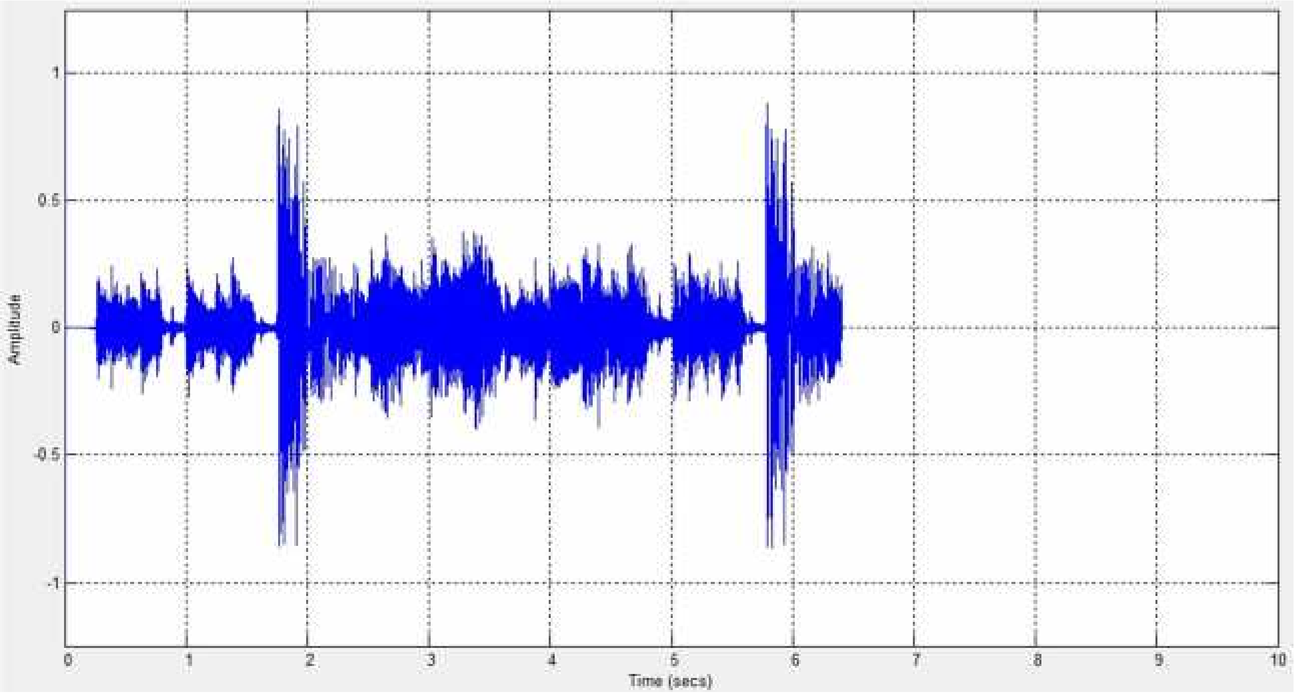
This final chapter explains the obtained measurements and results of the design implementation, and gives some recommendations and the future works of the project .

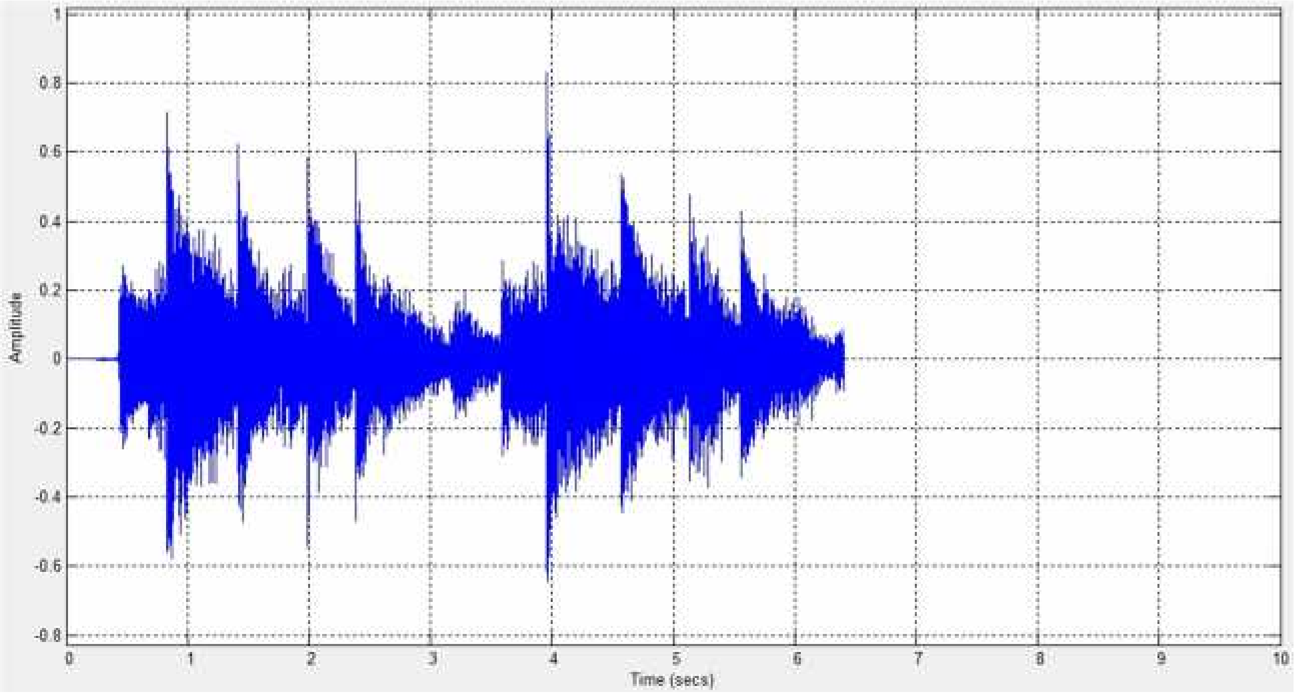
5.2 Results of the Practical Implementation

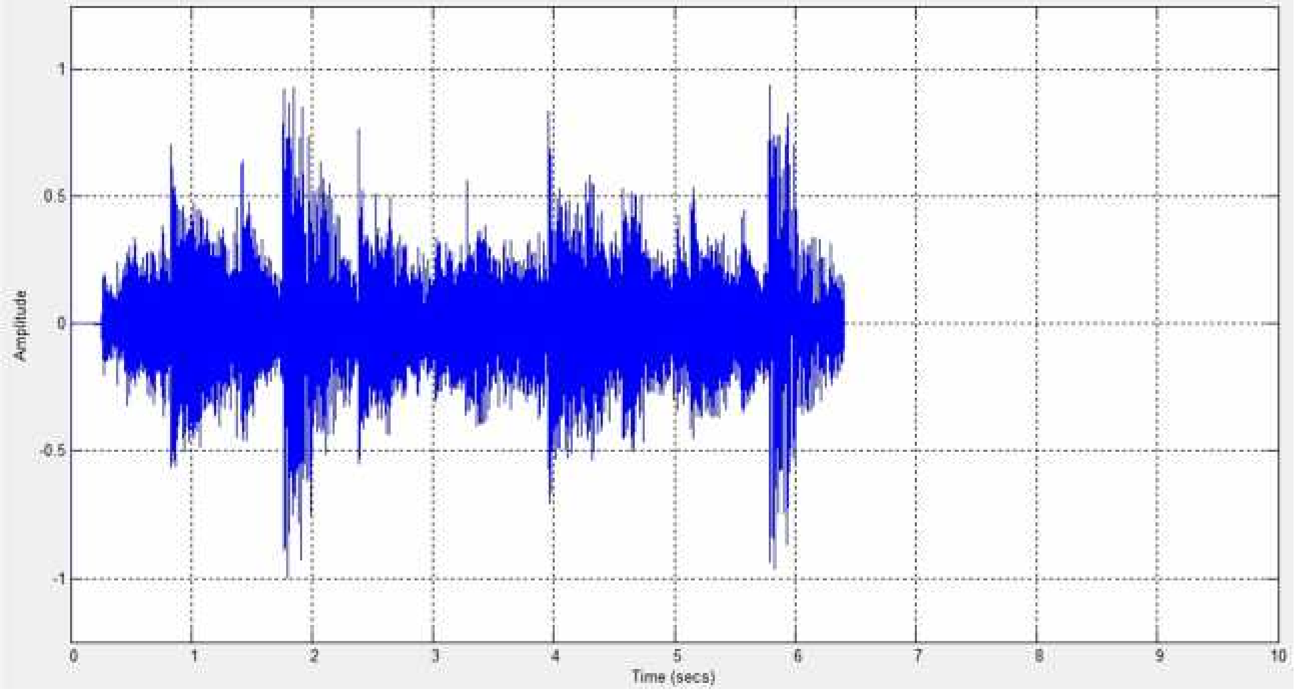
It was able to implement our proposed system of voice isolation on matlab simulink environment using many blocks from the library of this simulator after support it of the important parameters of each used block .

The main purpose of our project has achieved which is the isolation of the voice signal from the background signals, and we test the system in two technical way. The first way is by supporting the design of a voice signal on a microphone and voice and background signals on the other microphone , and it successfully cancels the background signals and outputs the voice signal, in the second way the same voice signal is supported to the two microphones and the LMS filter successfully cancels them and there was no output on the headphone.

When the step size changed to a small value less than 0.1 there will be a little echo in the output so to get rid of it is better to set the value to 0.1, the next figure show to us what is the different.




Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8
9
10
Time (secs)


Amplitude
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10
Time (secs)

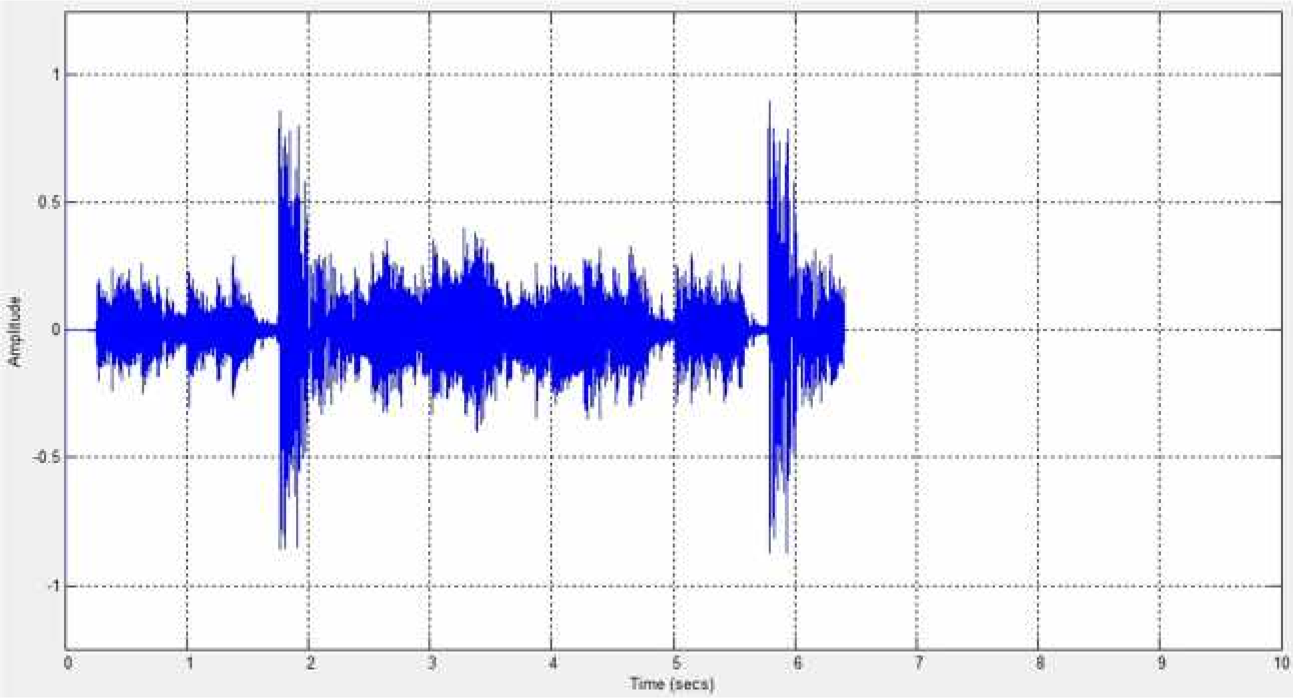
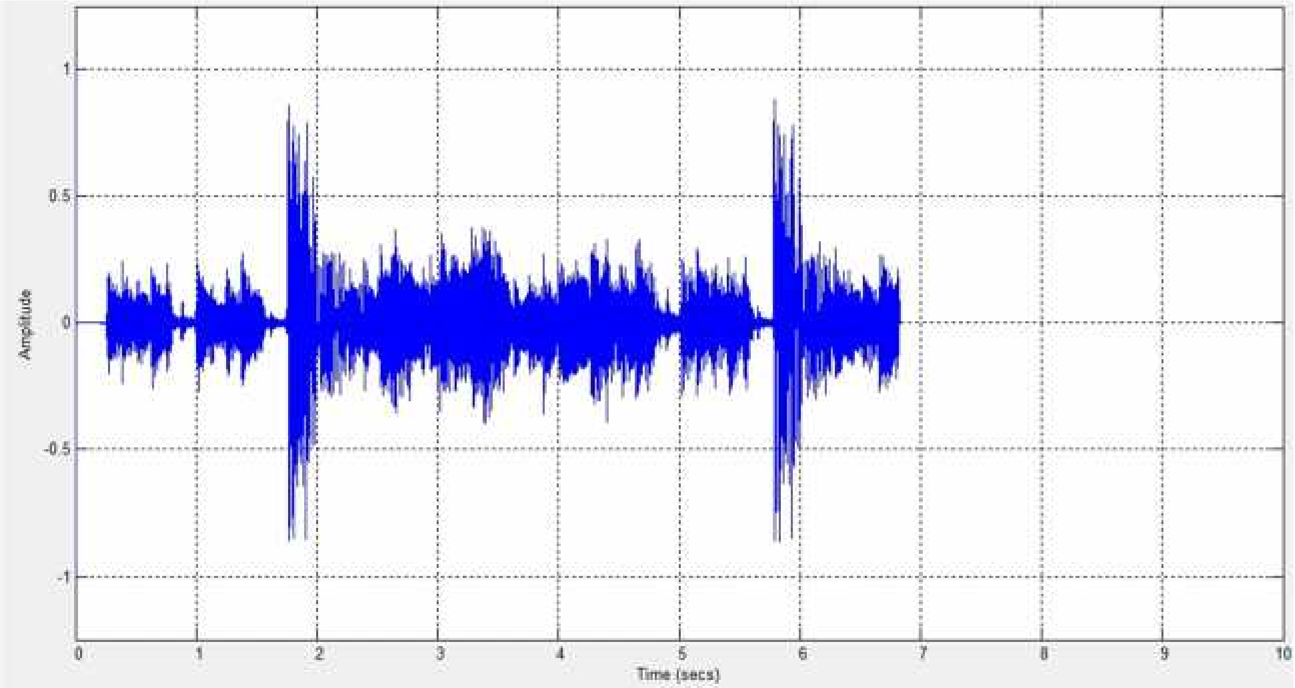
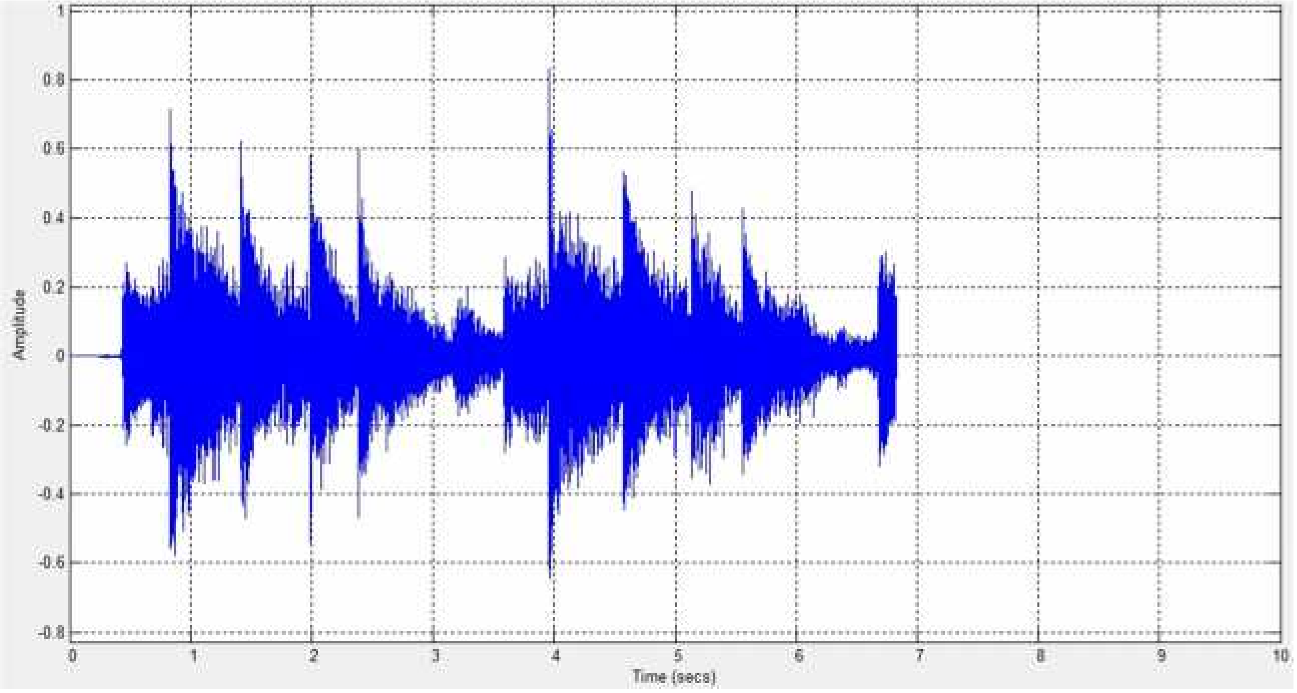
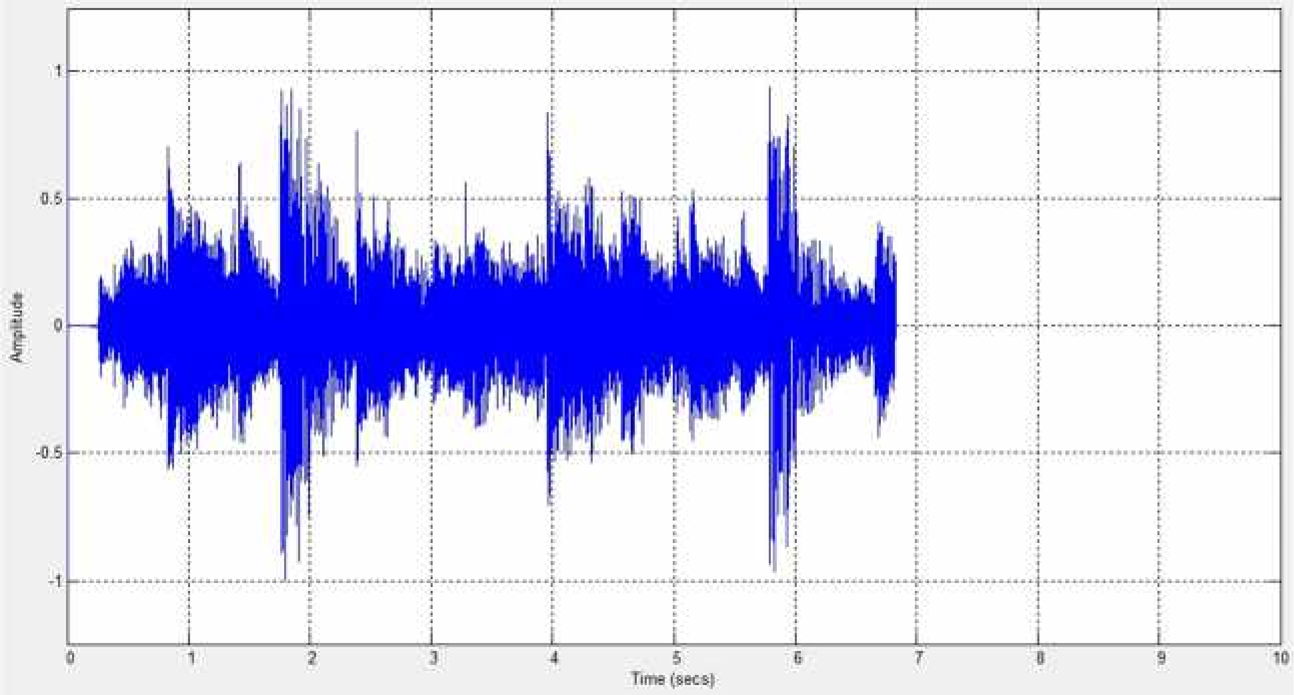


Figure 5.1 The output signal when step size is less than 0.1

When the step size changed to a large value more than 0.1 there will be a little echo in the output and the filter need more time to estimate the noise so to get rid of it is better to set the value to 0.1, the next figure show to us what is the different.




Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8
9
10
Time (secs)


Amplitude
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10
Time (secs)

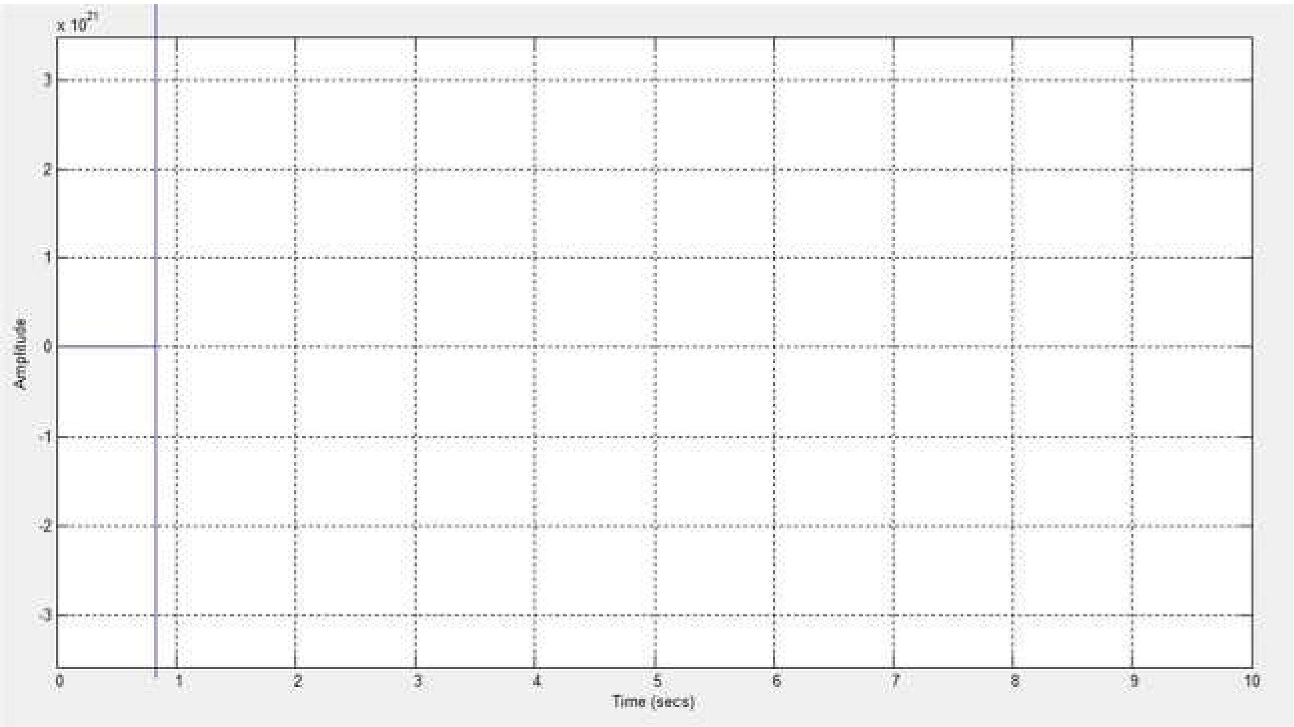


Figure 5.2 The output signal when the step size is larger than 0.1

5.3 Future Works

We were able to represent the final design in software and hardware . The DSP board was able to properly distinguish between the voice and noise signal and significantly reduce the noise. In addition, the wireless portion transmitted a clear signal well beyond our stated specifications. Most importantly, all the components worked together seamlessly to function in a real life situation.

We have a number of recommendations to improve the performance of our design. One of our principle concerns is the fact that the wireless technology that can be used is radio frequency. with radio frequency transmission, an unknown person can intercept the channel to receive the communication without the knowledge of the intended participants. Using a digital protocol with encoding such as Blue Tooth can alleviate such a problem.

Using digital communication can further improve performance by erasing the need for filters in the back end of the receiver. With analog communication, noise was introduced through the wireless system. A system that transmits digitally will not encounter that problem and will instead transmit a clearer signal. However additional work will be needed to produce a wireless protocol and add digital-to-analog converter on the back end of the receiver.

There are several steps that can be taken to improve the packaging of the product for the actual market. First, the DSP board will have to be replaced by a DSP chip. The board contains components such as LEDs and switches that we never used during our design process, and they represent a waste of material and space. Secondly, the microphone array can be dramatically improved by adding more appealing cupping devices and using materials that would eliminate noisy such as sponges.

To make our system more reliable, there will be additional part can be added to our system, which is the wireless components and its explained in the following:

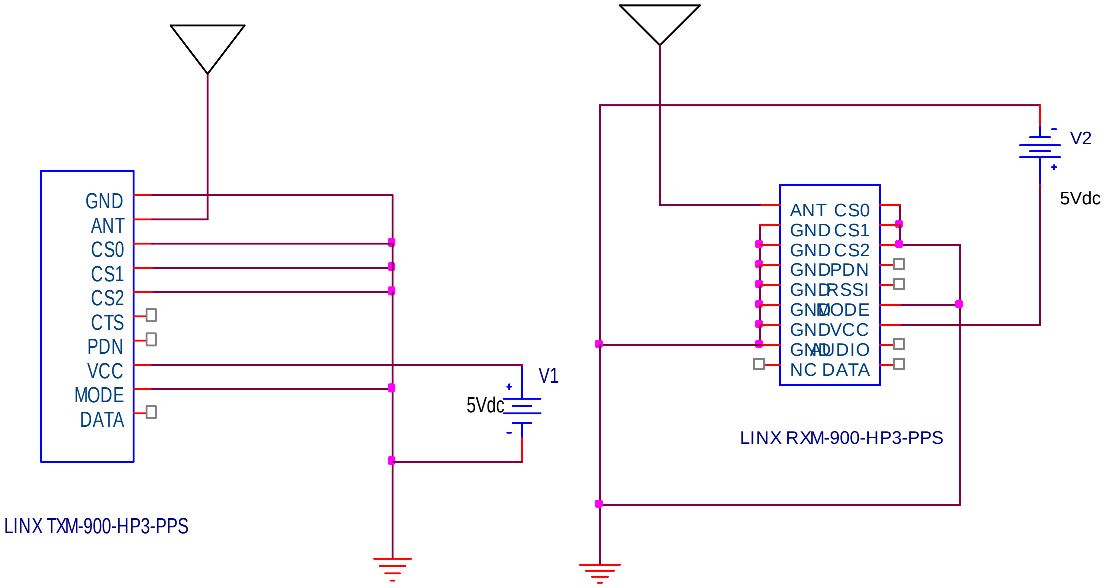


Figure 5.3 LINX TXM-900-HP3-PPS Transmitter and RXM-900-HP3-PPS Receiver

In this project you can use the Radio Frequency (RF) to connect between transmitter and receiver, using RF fit the scope of our project, as all we needed was to transmit a wireless audio signal across the length of a room.