[[1]](#footnote-1)

Digital Channel Sounding and Noise Filtering

McRee Brian, He Xiang and Okodaso Mamie

*Abstract*—This report presents our study of digital channel sounding and noise filtering. We were able to estimate the amplitude frequency-response of a digital channel between a transmitter and a receiver in the frequency range given. We designed a digital filter using MATLAB program that reduces added noise in the wireless transmission of a telephony speech signal which was applied to the speech signal in the given frequency range.

# INTRODUCTION

A

digital wireless communication channel can be modeled as a discrete-time LTI system with additive noise. The aim of this project was to apply advanced MATLAB functions to design and implement the channel for the transmitted and received signals. The channel; model in which x(n) is the transmitted signal, h(n) is the channel’s unit sample response, N(n) is the noise, and y(n) is the received signal. The effects of additive noise could be reduced through the use of noise filtering.

# THEORY

## Determining the Noise

Due to the random nature of noise, multiple samples of audio are needed. Following the Law of Large Numbers, the noise spectrum will converge to an expected value for each frequency after a sufficient amount of samples are taken. Procedure wise, a long recording of noise was taken. The long sample was divided into standard lengths and averaged. This produces an average noise spectrum that can be used to reduce noise.

## Channel Sounding

A given channel will attenuate certain frequencies more than others. This can be modeled by producing each audio frequency on one end of the channel, and measuring the magnitude at the other end of the channel. The frequencies can be swept by use of the “Chirp” function in MATLAB. Noise can corrupt measurements on the receiving end, so multiple samples must be taken and averaged, much like the noise spectrum. After averaging and ensuring convergence, a band from 200Hz to 14kHz of the chirp was created to match the system.

## Filtering

The voice band has been defined to be from 300 to 3.3kHz. A brick wall type bandpass filter can eliminate all unwanted frequencies outside of the voice band. The effect of the channel on the transmitted signal needs to be reversed by equalization. A direct inversion of the channel transfer function fails to equalize the signal due to extremely high gains from zeros on the transfer function.

## Flowcharts

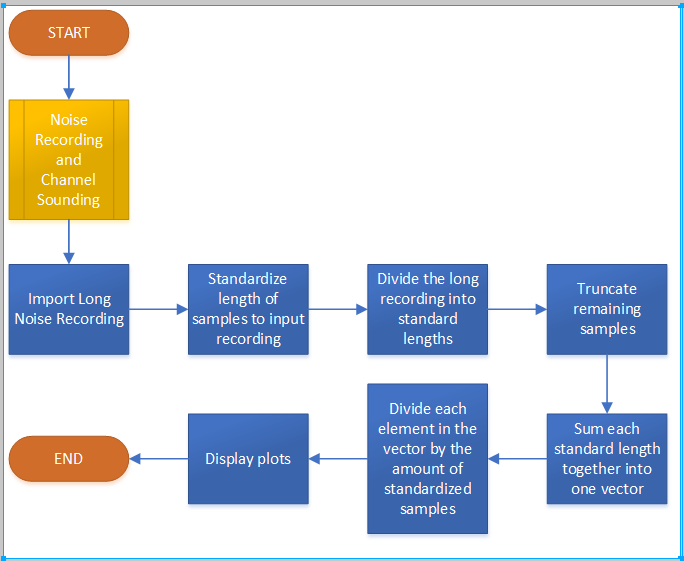


Fig. 1. Process of getting the noise recording and channel sounding.

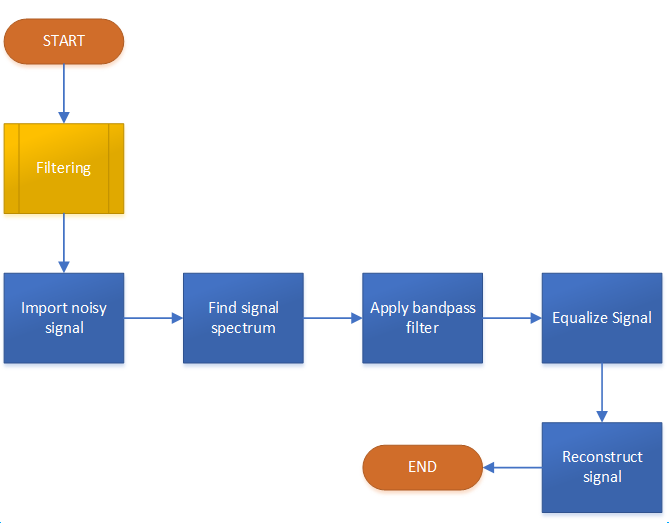


Fig. 2. Process of filtering the sound.

# MATH

## Equations

*System model:*

x(n) \* h(n) + N(n) = y(n) (1)

*Frequency domain equation:*

X(ω) H(ω) + N(ω) = Y(ω) (2)

Remove noise and equalize:

X(ω) = (Y(ω) - N(ω)) /(H(ω)) (3)

# Conclusion

Ideal “Brick wall” types of digital filters effectively remove undesired frequencies that corrupt signal. This only applies to filters in software as they are not realizable in a LTI system as the system order would approach infinity. Despite this ideality, noise that exists within the voice band will not be removed, thus a still imperfect signal. In the future, more robust methods on noise removal, such as the noise power density spectrum, need to be implemented. Equalization can become quite complex due to inversion of the transfer function. Other methods of equalization needs to be implemented. Finally, simple mathematical models that describe a system do not always yield valid results when applying them to a more complex real world system.

1. [↑](#footnote-ref-1)