Cancellation algorithms applied to recorded audio data

An investigation into the effectiveness of cancellation algorithms applied to recorded audio data

Presented by:
Karan Joseph Abraham

Prepared for:
Dr W. P. F. Schonken Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical and Computer Engineering.

October 24, 2024

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed and has been cited and referenced. Any section taken from an internet source has been referenced to that source.
3. This report is my own work and is in my own words (except where I have attributed it to others).
4. I have not paid a third party to complete my work on my behalf. My use of artificial intelligence software has been limited to generating some of my plots of my data, and generating latex code for my tables.
5. I have not allowed and will not allow anyone to copy my work with the intention of passing it off as his or her own work.
6. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Word count: 4722

Name: Karan Abraham

Signature: ______________________

Date: 24/10/2024

Terms of Reference

The Endler Hall at Stellenbosch University's Music Conservatorium is commonly used to record music concerts and other performances. A problem that is often encountered is noise and interference from outside the venue, such as traffic or pedestrians. This project will investigate the effectiveness of several cancellation algorithms commonly used in Passive Radar systems, such as ECA and CGLS, when applied to recorded audio data.

Acknowledgments

I would like to thank the following people for their help:

My brother, Dr Chris Abraham, for always being willing to help me out at any hour not just in this project, but throughout my entire degree.

My supervisor, Dr Francois Schonken, for making himself available constantly throughout my project and advising me on issues I brought up in our weekly meetings.

Dr Stephen Paine, for stepping in when Dr Schonken was on leave.

Professor Fred Nicolls, for inspiring me to do a digital signal processing project.

My friend, Bonga Njamela, who did most of the heavy-lifting in our projects and who I learned most of my Latex from.

My other lecturers, friends and classmates: this degree was far more enjoyable with them around particularly after doing half the degree in quarantine.

Abstract

The aim of this study is to investigate the effectiveness of cancellation algorithms, commonly used in Passive Radar, applied to recorded audio data. The algorithms used in particular are ECA, Gradient Descent and CGLS. The purpose of the study is for the algorithms to be applied to concert recordings taken at Endler Hall at Stellenbosch University. The algorithms were tested on different types of audio. The audio sounds were synthetically overlayed to simulate recording. The results show that all the algorithms were effective, which was measured using correlation of signals. ECA was found to be the most efficient, executing in the fastest time in all tests. This study gives a foundation for the algorithms to be applied to real recordings.

Contents

Chapter 1

Introduction

1.1 Background to the study

The Endler Hall at Stellenbosch University's Music Conservatorium is commonly used to record music concerts and other performances. However, it is situated near a busy street and surrounded by noise from students. Therefore, a problem that is often encountered is noise from outside the venue, such as traffic or pedestrians, can be heard in the concert recordings. High quality productions are recorded there; therefore, they cannot have noise in the recordings. One solution to this problem is to apply a cancellation algorithm to the recording. There has been much research on cancellation algorithms applied to Passive Radar, such as ECA and CGLS. These algorithms could be applied to audio recordings too. However, an additional recording would be required that records the noise alone. This could be achieved by placing microphones outside the Endler Hall during recording as shown in figure 1.1.

1.2 Objectives of this study

1.2.1 Purpose of the study

Passive Radar cancellation techniques can be applied to any kind of digital data. Therefore, they are suitable for audio noise cancellation. Determining the effectiveness of the algorithms on audio, tells us whether they can be used as a solution for eliminating noise

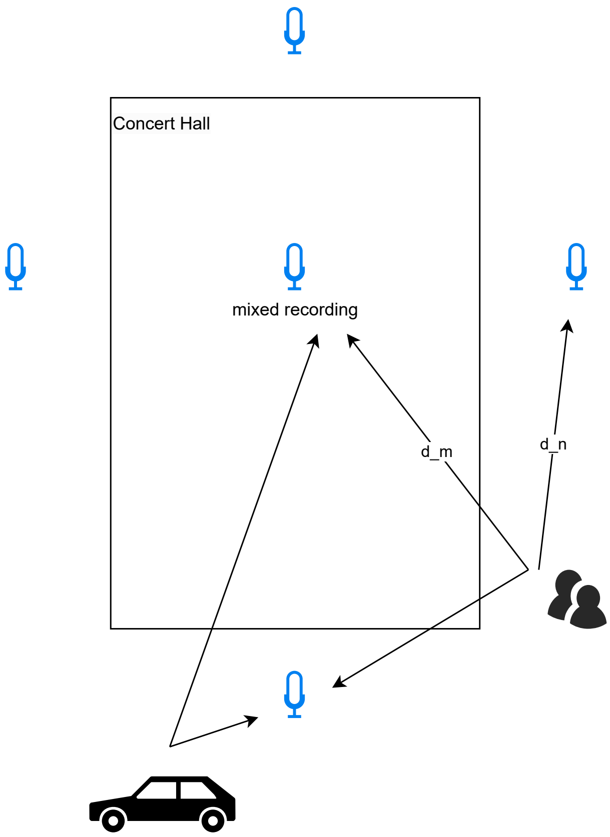

Figure 1.1: Plan of setup at the concert hall with one microphone inside to record the concert and 4 microphones outside to record just the noise outside. d_m is the distance between the noise source and the "mixed" microphone. d_n is the distance between the noise source and one of the "noise" microphones.

in the concert recordings at Endler Hall.

1.2.2 Problems to be investigated

The system needs to be modeled as the the Passive Radar algorithms. The noise and mixed recordings will pick up the unwanted noises at different times depending on the distance of the noise sources from the microphones. This is depicted in figure 1.1.

Another problem to be investigated is how algorithms effectiveness can be measured.

1.3 Scope and Limitations

The concert recording containing both the concert music and the unwanted noise is synthetically created, not actually recorded. The algorithms are tested with generated mathematical signal audio, as well as music audio to simulate the concert and noise signals. This project assumes that all significant noise sources are stationary; therefore, Doppler effect is not considered.

1.4 Plan of development

This study begins by investigating and comparing the cancellation algorithms commonly used in Passive Radar in chapter 2. Thereafter, in chapter 3, the system is modeled to the algorithms and its mathematical theory is described. Chapter 4 describes how the algorithms are applied to the audio data. The algorithms are finally tested on the data in chapter 5.

Chapter 2

Literature Review

Literature in this chapter covers experiments done on radar data to cancel noise. There is an extensive amount of experiments done on cancellation techniques in the Passive Radar field. In contrast, limited literature was found on audio cancellation. The most common algorithms in Passive Radar are LMS (Least Mean Squares) based. Using the LMS algorithm, the suppressed signal is described in equation (2.1) as done by Russel [1].

$$s = m - Nx \tag{2.1}$$

m is the mixed signal. N is a matrix where each column is a shifted replica of the noise signal. x is a vector consisting of each weight for each signal in N. The algorithms reviewed in this section are used to determine x that cancels the noise the most effectively. Additionally, the algorithm should be efficient, so that it can work for long recordings.

2.1 Extensive Cancellation Algorithm

The Extensive Cancellation Algorithm (ECA) algorithm was first proposed by Colone et. al in 2008 [2], when investigating disturbance removal in radar. It implements the LMS algorithm. It is the simplest algorithm as it solves for x analytically, by setting the derivative of $f(x)$ to zero [1]. This involves multiplying b to the inverse of A. The result of x is shown in equation (2.2).

$$x = A^{-1}b \tag{2.2}$$

where $A = N^T N$ and $b = N^T m$.

2.1.1 Extensive Cancellation Algorithm in Batches

In order to to achieve better efficiency, Colone [2] introduced the Extensive Cancellation Algorithm in Batches (ECA-B). In [2], the ECA-B performed target detection better than the ECA. An additional advantage of this algorithm is that it can be effectively parallelized, as shown by Hicks [3] in his investigation into efficient processing of radar data. The fact that is can be parllelized is particularly important when dealing with large quantities of data in real-time.

2.1.2 Extensive Cancellation Algorithm by Carrier

The Extensive Cancellation Algorithm by Carrier (ECA-C) builds on the ECA. This algorithm is designed by Zhao et. al. as noted in [3]. It uses properties of OFDM signals and Fourier Theory [3]. These properties allow the algorithm to be parallelized more effectively than ECA-B. Hence, it outperforms the ECA-B in [3] by recording a faster execution time.

2.2 Iterative Algorithms

The iteraritve algorithms, Gradient Descent and CGLS were investigated. Iterative methods are suited for use with sparse matrices [4]. In this case, if N is sparse, meaning it contains many zeros, then an iterative algorithm would be more efficient. A disadvantage of the iterative algorithms is that it has limited parallelism, since it an iterative operation. This means that speed-up potential is limited [3].

2.2.1 Gradient Descent Algorithm

Gradient Descent uses the gradient to iteratively update x to find the minimum point of $f(x)$. It updates x using equation (2.3).

$$x_{i+1} = x_i + \alpha_i r_i \tag{2.3}$$

where α is the step size and r is the residual (negative gradient).

2.2.2 Conjugate Gradient Least Squares Algorithm

The Conjugate Gradient Least Squares (CGLS) algorithm builds on the Gradient Descent Algorithm, but uses conjugate directions d to update x iteratively using equation (2.4).

$$x_{i+1} = x_i + \alpha_i d_i \tag{2.4}$$

This results in x reaching the solution in fewer iteration than Gradient Descent. This is shown in a study by Tong [5] , where the algorithm reaches a satisfactory solution in just two iterations.

2.3 Audio applications

Bernardi et. al. [6] considers three algorithms applied to audio signals. These are BNLMS (Block Normalized LMS), PEM-FDAF (Prediction Error Method based Frequency Domain Adaptive Filter), and PEM-FDKF (PEM based Frequency Domain Kalman Filter). They find that BNLMS is the simplest but PEM-FDKF has the best performance. They also found that the subjective scores correlated highly with the objective scores. This shows that listening to the recordings is a good indicator of the effectiveness of cancellation.

Chapter 3

Modelling and Theory

This section details the modelling of the system mathematically and describes the full mathematical theory using the model.

3.1 Least Mean Squares

The LMS algorithm is an adaptive filter that was first developed by Widrow and Hoff in 1960 [7] [1].

Russel [1] describes the algorithm using two time-delayed signals of the desired signal to create an estimate of the desired signal as shown in figure 3.1 .

In order to reduce the error, the weights of the system need to be adjusted. In this context, the input to system are the time shifted replicas of the noise signal, contained in each column of matrix N (3.1).

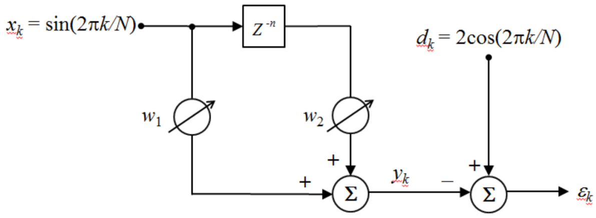

Figure 3.1: Example system used by Russel

$$N = \begin{bmatrix} n[k] & n[k-1] & \cdots & n[0] \\ n[k+1] & n[k] & \cdots & n[1] \\ \vdots & \vdots & \ddots & \vdots \\ n[k+L-1] & n[k+L-3] & \cdots & n[k] \end{bmatrix} \tag{3.1}$$

where k is the number of delayed noise signals. This is also referred to as the number of bins. L is the length of the signals. The length n is $k + L$. The weight of each signal as an element in vector x. The system; therefore, produces an estimate of the noise signal contained in the mixed signal, denoted $\hat{n}$. The system is modeled mathematically in equation (3.2).

$$\hat{n} = Nx \tag{3.2}$$

In order to remove the noise in the mixed signal, the $\hat{n}$ needs to be subtracted from the mixed signal, which creates the suppressed signal (3.3).

$$\begin{aligned} s &= m - \hat{n} \\ &= m - Nx \end{aligned} \tag{3.3}$$

In order to minimize s, the mean square of the suppressed signal $\frac{1}{L}\sum_{i=1}^{L} s_i^2$. However, in this study (as is done in other studies) the mean square is not normalized. This means the cost function $f(x) = \sum_{i=1}^{n} s_i^2$. This is also known as the signal energy. Substituting $s = m - Nx$ into $f(x)$ results in

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} s_i^2 \\ &= x^T N^T N x - 2m^T N x + m^T m \\ &= x^T A x - 2b^T x + m^T m \end{aligned} \tag{3.4}$$

$A = N^T N$ the cross-correlation matrix of the possible delayed noise signals, and $b = N^T m$ the the cross-correlation matrix of the mixed signal with the possible delayed noise signals.

In order to minimize the cost function, its derivative should be set to 0. The derivative is $f'(x) = 2Ax - 2b$. However, as done by Russel [1], the factor of 2 is removed which results in a simpler derivative, $f'(x) = Ax - b$; therefore, the equation to be solved is (3.5).

$$Ax - b = 0 \tag{3.5}$$

x is solved using one of the algorithms described below. The full algorithm is illustrated in algorithm 1.

Algorithm 1 Suppress function

function SUPPRESS(n, m, min_delay, num_bins)
  $suppressed \leftarrow m[min\_delay : end]$
  $beg \leftarrow m[1 : end]$ ▷ This part is unaffected by noise
  $N \leftarrow$ zero matrix of dimentions ($length(m) \times num\_bins$)
  for $i \leftarrow 1$ to num_bins **do**
    Populate i'th column with noise delayed by i samples
  end for
  $A \leftarrow N^T N$ ▷ Matrix of correlations between noise variations
  $b \leftarrow N^T m$ ▷ Vector of correlations between mixed and noise variations
  Determine x using one of the methods used in the cancellation algorithms
  $suppressed \leftarrow m - Nx$
  $suppressed \leftarrow [beg; suppressed]$
end function

The ECA algorithm implements the LMS algorithm and solves for x (2.2) analytically by rearranging equation 3.5

$$x = A^{-1}b \tag{2.2 revisited}$$

3.2 Iterative Methods

Steepest Descent or CGLS determine x interatively instead.

3.2.1 Gradient Descent

This algorithm updates x using the gradient and a step size.

$$x_{i+1} = x_i + \alpha_i r_i \tag{3.6}$$

where α is the step size and r is the residual. The residual is the negative of the gradient (3.7).

$$\begin{aligned} r &= -f'(x) \\ &= b - Ax \end{aligned} \tag{3.7}$$

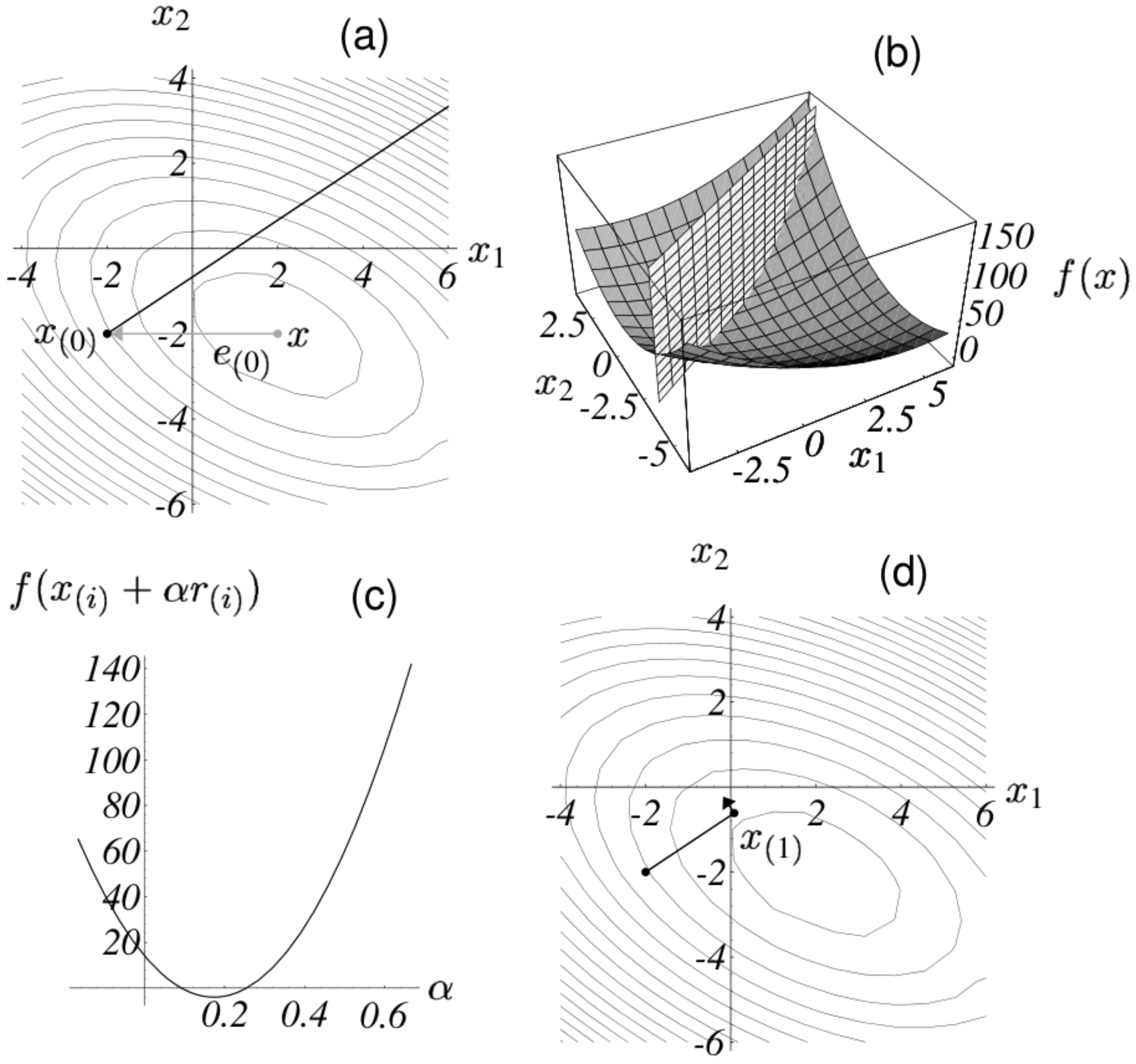

Figure 3.2: Plots describing method of determining step size [4].

Determining step size

This subsection details how the first step size is determined. Subsequent step sizes are determined the same way. To determine the step size, a line search is conducted. A line search searches along a line for a value of α that minimizes f along the line shown in figure 3.2 (a). This line is in the direction of the residual.

$f(x_1(\alpha))$ is shown in figure 3.2 (c). The minimum point of $f(x_1(\alpha))$ is calculated by equating its derivative to zero.

$$\frac{df(x_1)}{d\alpha_0} = 0 \tag{3.8}$$

It can be shown that the step size needs to be such that r_1 is orthogonal to r_0.

First, the derivative is written in terms of r_0 and r_1

$$\begin{aligned} \frac{df(x_1)}{d\alpha_0} &= \frac{df(x_1)}{dx_1}^T \frac{dx_1}{d\alpha_0} \\ &= f'(x_1)\frac{dx_1}{d\alpha_0} \\ &= -r_1^T r_0 \end{aligned} \tag{3.9}$$

Then expression 3.9 is substituted into equation 3.8

$$\begin{aligned} \frac{df(x_1)}{d\alpha_0} &= 0 \\ -r_1^T r_0 &= 0 \\ r_1^T r_0 &= 0 \end{aligned} \tag{3.10}$$

r_1 can be written in terms of α

$$\begin{aligned} x_1 &= x_0 + \alpha d_0 \\ A(x_1) + b &= A(x_0 + \alpha d_0) + b \\ r_1 &= r_0 + \alpha A d_0 \end{aligned} \tag{3.11}$$

Now, by substitution and rearranging equation 3.10, α can be determined

$$\begin{aligned} r_0^T r_1 &= 0 \\ r_0^T(r_0 + \alpha_0 A r_0) &= 0 \\ \alpha_0 &= \frac{r_0^T r_0}{r_0^T A r_0} \end{aligned} \tag{3.12}$$

3.2.2 CGLS

The CGLS algorithm updates x using search directions instead of just the gradient.

$$x_{i+1} = x_i + \alpha_i d_i$$

where d is the search direction.

Determining step size

To determine the step size α, equation 3.8 is used. Note that here $\frac{dx_1}{d\alpha_0} = d_0$.

$$\begin{aligned} \frac{df(x_1)}{d\alpha_0} &= \frac{df(x_1)}{dx_1}^T \frac{dx_1}{d\alpha_0} \\ &= -r_1^T d_0 \\ &= -(r_0 + \alpha A d_0) d_0 \end{aligned} \tag{3.13}$$

By equating this to zero and rearranging, α can be determined

$$\alpha = \frac{d_0^T r_0}{d_0^T A d_0}$$

There are n search directions, $d_0, d_1, ..., d_n$, that are A-orthogonal or conjugate to each other. Two vectors d_i and d_j are A-orthogonal if $d_i^T A d_j = 0$.

The first search direction $d_0 = r_0$; thereafter d_i is computed as

$$d_1 = r_1 + \beta_1 d_0$$

where

$$\beta_1 = \frac{r_1^T r_1}{r_0^T r_0}$$

The derivation for this is described in technical report by Shewchuk [4].

Algorithm 2 illustrates how x is calculated in the CGLS algorithm.

3.3 Computational considerations

The part of the run time that takes longest is lines 7 to 9 in algorithm 1 . This is known as the hot spot. Therefore, it is important to consider what effects the number of computations in this section of code. These lines compute matrix multiplication.

Algorithm 2 Calculating x in CGLS Algorithm

Input: A, b
Output: x
num_bins $\leftarrow$ length(b)
$x \leftarrow$ zero vector of length num_bins
$r \leftarrow b - A \cdot x$
$d \leftarrow r$
for $i = 1$ to num_bins **do**
  $\alpha \leftarrow \frac{d^\top r}{d^\top Ad}$
  $x \leftarrow x + \alpha \cdot r$
  ▷ Next iteration:
  $r\_prev \leftarrow r$
  $r \leftarrow r - \alpha \cdot A \cdot r$
  $\beta \leftarrow \frac{r^\top r}{r\_prev^\top r\_prev}$
  $d \leftarrow r + \beta \cdot d$
end for

3.3.1 Matrix Multiplication Theory

Let there be two matrices A and B where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

and the resulting matrix C

$$C = A \times B = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

Matrix multiplication's number of computations can be determined using number of elements of C multiplied by the number of computations per element.

Each element c_{ij} in matrix C is computed as the dot product of the i-th row of A and the j-th column of B. Let the i-th row of matrix A be represented as a row vector

$$\vec{a}_i = \begin{pmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{pmatrix}$$

Similarly, let the j-th column of matrix B be represented as a column vector

$$\vec{b}_j = \begin{pmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{pmatrix}$$

The element c_{ij} in the result matrix C is the dot product of the row vector $\vec{a}_i$ from matrix A and the column vector $\vec{b}_j$ from matrix B

$$\begin{aligned} c_{ij} &= \vec{a}_i \cdot \vec{b}_j \\ &= a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} \end{aligned} \tag{3.14}$$

There are n terms and 2 computations per term; therefore, the number of computations per element is equal to $2n$. This is multiplied by the number of elements in C to give a total number of computations.

$$\text{Total computations} = (m \times p) \times n \times 2 \tag{3.15}$$

3.3.2 Total Computations in Hot Spot

Using equation (3.15), the total computations can be calculated for lines 7 to 9 in algorithm 1, using ECA. Line 7 has $k^2 \times L \times 2$ computations. Line 8 has $k \times L \times 2$ computations. line 9 is $x \leftarrow A^{-1}b$ in the ECA, which has $k \times k \times 2$. Therefore, the sum total is $2Lk^2 + 2Lk + 2k^2 = (2L + 2)k^2 + 2Lk$. Therefore, computational time would increase by $O(n^2)$ as a function of k. This means increasing the number of bins would increase execution time quadratically.

Chapter 4

Methodology

This section describes the implementation of the algorithms. The code and sound files can be found in the project repository. The link to the repository can be found in Appendix D. As shown in figure 4.1 , there are three top-level blocks of the project implementation: `Load params`, `Suppress`, and `Verify`. `Load params` loads the pure, noise and mixed signals. The pure signal is only for verification. `Load params` also sets the delay range that will be used in `Supress`. `Supress` suppresses the noise in the mixed signal using one of the cancellation algorithms detailed in section 3 . `Verify` takes the suppressed signal and compares it to the pure signal.

4.1 Parameters

This subsection details how the parameters of the `suppress` block are loaded.

4.1.1 Creating audio files

The audio signals were created using Audacity or Matlab. Audacity is a digital audio editor. Audacity was used to create the pure, noise, and mixed signals. The mixed signal was They were then read into Matlab using the built-in `audioread` function. This loads the audio file as a matrix with dimensions number of samples by number of channels.

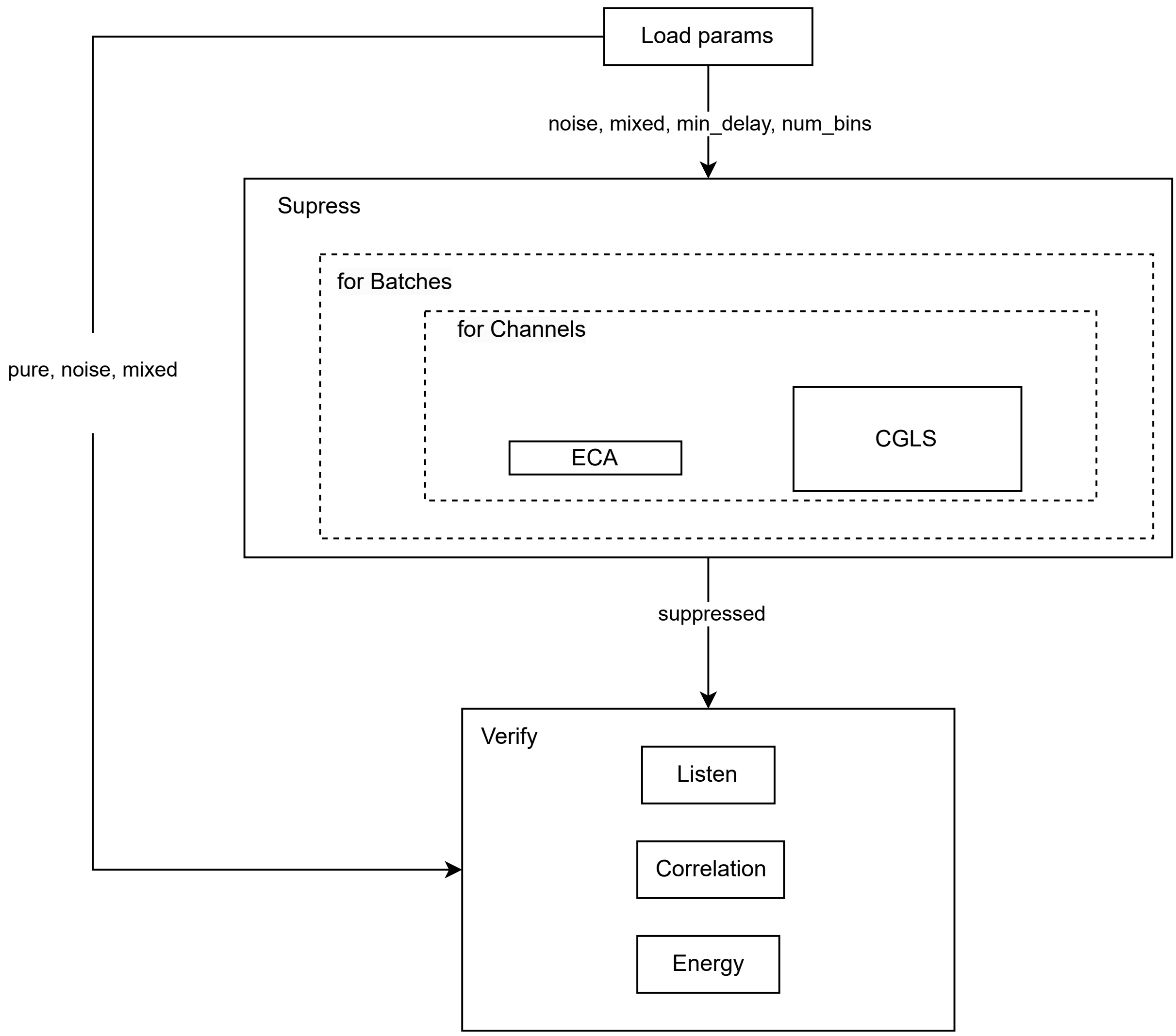

Figure 4.1: Block diagram of implementation

4.1.2 Delay Range

Every possible time delayed noise needs to be correlated against the mixed signal; therefore, the range of the possible time delays needs to be determined. This is based on the range of differences in distance of the noise sources from the microphones as depicted in figure 1.1 . In this study, delay ranges were experimentally chosen based on reasonable execution times.

In a setting where the actual delay is unknown a reasonable range of Δd, differences in distances, needs to be determined. Given minimum and maximum Δd, and the speed of sound being 343 m/s, the minimum and maximum sample delay can be determined.

$$t_{delay} = \frac{\Delta d}{343} \tag{4.1}$$

$$\begin{aligned} n_{delay} &= t_{delay} f_s \\ &= \frac{\Delta d}{343} f_s \end{aligned} \tag{4.2}$$

where f_s is the sampling frequency of the audio recording.

4.1.3 Batches

Music concert recordings can be quite long, meaning that the size of the data could be quite large. This would increase execution time to cancel any noise. Therefore, the algorithm was applied to batches of the audio data. This would ensure at least part of the mixed signal would be suppressed, and potentially improve execution time.

4.2 Suppression

This subsection covers the suppression of the noise in the mixed signal. It considers all noise The beginning part (from index 1 to `min_delay`) of the mixed audio does not need to be processed because the noise starts being recorded at the same time as the mixed audio, therefore the earliest possible time the noise can feature in `mixed` is at `t = min_delay`. This is illustrated in figure 4.2 .

N is initialized as a matrix of zeros. Its number of rows is the length of the mixed audio.

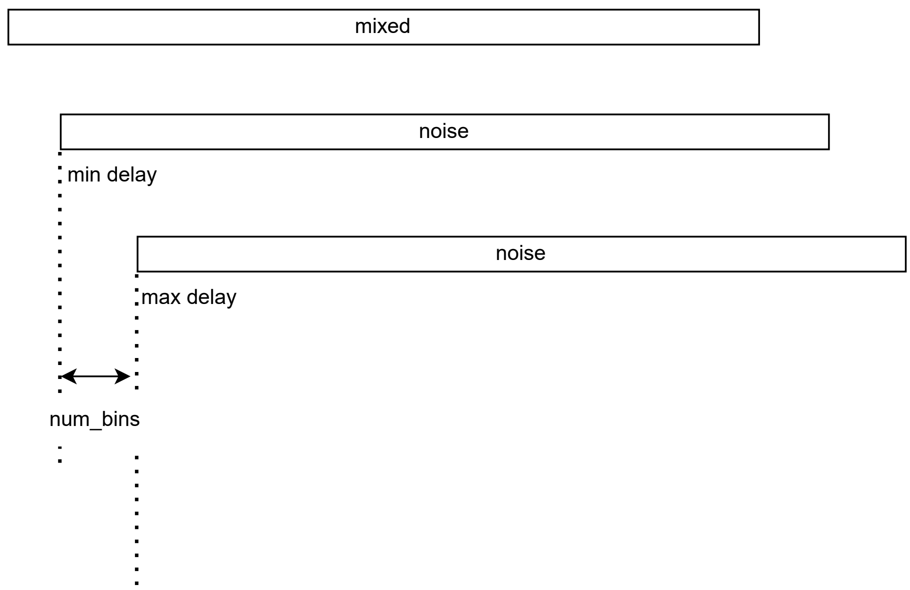

Figure 4.2: Delay range of noise in mixed signal

Its number of columns is `num_bins`. Each column of N is populated with the noise audio shifted by iterative amounts.

x is calculated using one of the algorithms. It can be seen in algorithm 1 that once `R` and `p` are calculated, only one line is needed to calculate the weights for the ECA algorithm.

Gradient Descent and CGLS requires more code to calculate x as shown in algorithm 2. CGLS was implemented using a `for` loop which iterated k times, where k is the number of bins. However, it was found that it gets close enough to the solution far earlier than the k-th iteration. Therefore, the algorithm was adjusted to break the loop if the gradient was very close to zero, which would be evaluated as zero by Matlab. Gradient Descent does not converge in a set number of iterations, but in order to prevent the program from possibly timing out (e.g. in the case of subsection 5.2.1), the algorithm was limited to a set number of iterations, should it take long to converge to the solution. The number of iterations was set to $2 \times k$.

4.2.1 Stereo

The cancellation algorithms are to be applied on music audio, which has two channels. Therefore, the algorithms need to be iterated on each channel. The audio is loaded as a matrix with 2 columns (for each channel). Its number of rows is the length of the audio recording.

4.3 Verification

This subsection details how the verify block checks if the implementation meets specifications. Several verification methods were used: hearing, correlation, and signal energy.

4.3.1 Hearing

`supressed` is written to a `.wav` file using Matlab's `audiowrite` function. This audio can be listened to and compared to unprocessed audio. This is a good way to evaluate the effectiveness of the cancellation as shown by Bernardi [6] . However, an objective measure is required to quantify the suppression. Two metrics were used: correlation and energy.

4.3.2 Correlation

Matlab's `corrcoef` function was used to find the correlation between `suppressed` and `pure`, denoted `corr`. This is compared to the correlation of `mixed` with `pure`, denoted as `pre_corr`. Correlation shows how similarly two signals vary. It is calculated using

$$r = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum\left(x_i - \bar{x}\right)^2 \sum\left(y_i - \bar{y}\right)^2}} \tag{4.3}$$

However, the mean of audio signals is basically zero; therefore subtracting from the mean is unnecessary. Matlab's corrcoef function computes this calculation. This can show us how similar the suppressed signal is to the pure signal, which tells how effective the noise cancellation was.

4.3.3 Energy

The signal energy of `pure`, `noise`, `mixed`, and `suppressed` were calculated, denoted as E_p, E_n, E_m, and E_s respectively. From there we can see how much energy was removed from `mixed`, and how much energy `suppressed` has compared to `pure`. The formula for signal energy is

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

For stereo signals, the two channels' energies were added to give the energy of the stereo signal.

4.3.4 Spectrum

The total energy does not tell us what part of the signal has been removed. Therefore, the energy suppression is also visualized using frequency spectrum plots. This was done by taking the ESD of the signal, which is calculated using

$$|\mathcal{F}\{x[n]\}|^2$$

This shows which frequencies in the mixed signal were attenuated, hence making it clear which signal (pure or noise) was removed. The ESD was plotted in dB because the range of the plot is so large. Plotting it in dB makes it more readable.

4.4 Tests

The system was tested on different types of audio and using different parameters.

Hearing, correlation, and energy validations were applied to all algorithms. Spectrum, varying batch length, and varying number of bins were only applied to the ECA.

4.4.1 White Noise Cancellation

This test was performed on simple audio generated in Audacity. Energy spectrum analysis cannot be seen as easily on music audio. Therefore, the implementation was additionally tested on a audio that is more distinct in the frequency domain. A tone of 440 Hz was used as the pure signal. White noise was used as the noise signal.

4.4.2 Music Cancellation

Stereo music audio was used for this test to more closely resemble the actual data used. Also, an additional noise source is introduced. Both noise signals are attenuated by 4 times. However, this information is only known because mixed signal was made synthetically. Therefore, (to work on real data) the original noise signal, not attenuated, was used. Since it is less obvious what the pure and noise ESDs look like, they were plotted as well.

4.4.3 Varying number of bins

Additionally, the number of bins and batch length was varied. Number of bins was varied as this system could be applied in a different context with a different range of noise delays. It was shown in subsection 3.3.2 that computational time would increase by $O(n^2)$ as a function of k. Therefore, it is expected that execution time increase as the number of bins is increased.

4.4.4 Varying batch length

Batch length was varied to see if batching the the audio data would improve accuracy and efficiency. This is determined by measuring the correlation and the execution time.

Chapter 5

Results and Discussion

This chapter details the results of experimenting the cancellation algorithms on white noise and music. Additionally, the ECA-B is tested on music, and the number of bins are varied.

5.1 Test 1 - White Noise Cancellation

The cancellation algorithms were first applied to mathematical signals. A 440 Hz tone was generated in Audacity to act as the pure signal. White noise was generated to act as the noise signal. The white noise signal was delayed and attenuated then added to the tone signal. This would act as the mixed signal.

```
For this test:
    Delays considered were 400 to 900.
    Audio length = 400 000 (approximately 10 seconds)
    No batching was done.
    mixed corr = 0.78 .
```

Hearing:

It can be heard in suppressed signal that the noise is fainter in comparison to the pure signal.

5.1.1 Correlation

Method	Correlation	Execution Time (s)
ECA	0.983	4.87
Grad Desc	0.983	5.52
CGLS	0.983	5.34

Table 5.1: Correlation and Execution Time by Method (3 significant figures)

Table 5.1 , shows that all algorithms improved correlation from 0.78 to 0.98 . ECA was the fastest to execute. Figure C.1 visually shows the improvement, which can be found in Appendix C. Figure C.2 compares the execution times visually, which can also be found in Appendix C.

5.1.2 Energy

Quantity	Value (kJ)
E_p	141
E_n	6
E_m	186

Table 5.2: Energy values in kJ for E_p, E_n, and E_m. It must be noted that E_n refers to the energy of the noise (which has been attenuated) in the mixed signal.

Table 5.2 shows that $E_m = 186kJ$. All the algorithm make $E_s = 116kJ$. Therefore, the cancellation took the energy from $186kJ$ to $116kJ$. This is less than E_p , which means that some of the E_p in the mixed signal was removed as well.

5.1.3 Spectrum

Figure 5.1 shows that the spike (from the pure signal) is preserved while the band (from the noise) is attenuated. The bottom plot shows more clearly the noise attenuation.

5.2 Test 2 - Music Cancellation

Pure was a stereo music audio. It was overlayed by two noise music audios in Matlab. All the audio lengths were set to 400 000 (10 seconds). The actual delays are 500 and 800.

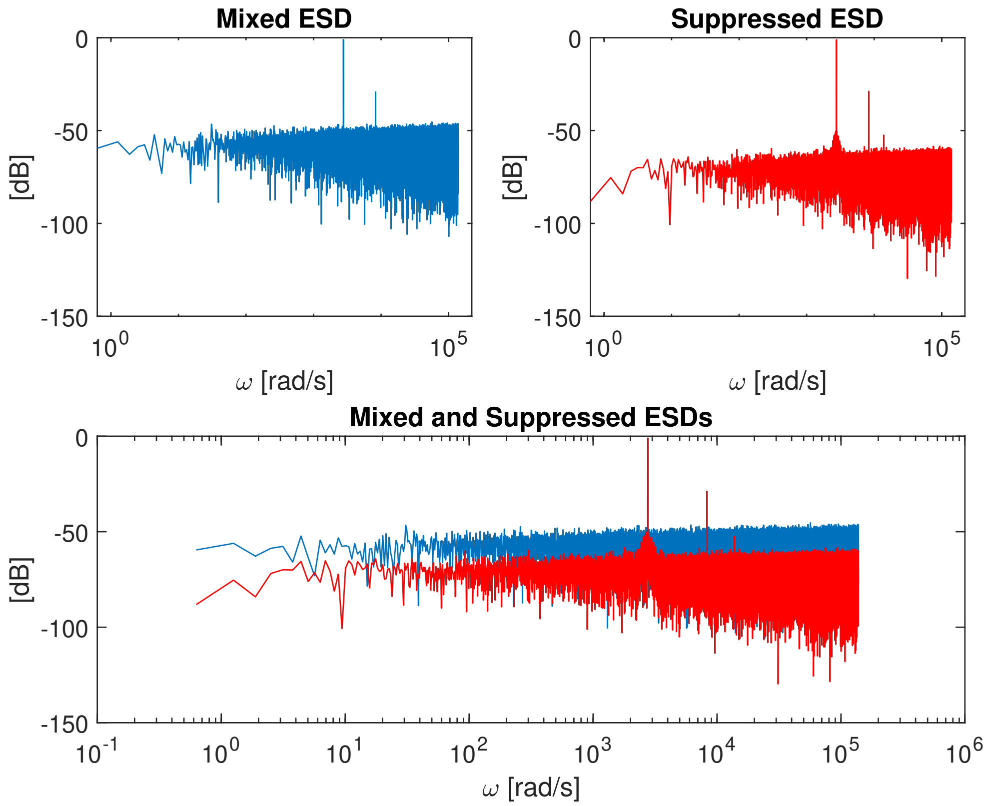

Figure 5.1: One-sided ESD plots of mixed and suppressed signal. The bottom plot shows the top plots overlayed with each other. This shows which frequencies were attenuated.

`Min_delay` is 100. Therefore, at least 700 bins are required to cover all the actual delays in this experiment. The correlation between pure and mixed was `pre_corr = 0.8779`. Using audio length of 400 000 without batching and bin range from 400 to 900 (`num_bins` = 500), all algorithms were compared.

Hearing:

When playing the audio files, the noise signal sounds distinctively softer. The audio files can be found on the project repository in the `sound files\stereo` folder.

5.2.1 Correlation

Method	Correlation	Execution Time (s)
ECA	0.9610	8.28
Grad Desc	0.9607	9.13
CGLS	0.9610	8.70

Table 5.3: Comparison of algorithms

Table 5.3 shows that all algorithms improved the correlation from 0.88 to 0.96 . ECA and CGLS performed slightly better than Gradient Descent, which had not converged to the solution within $2 \times k$ iterations. ECA is the fastest because the A matrix is dense (not sparse which is more suited for iterative algorithms). Gradient Descent and CGLS iterated without breaking out the loop. Therefore, CGLS iterated half as many times as Gradient Descent, hence why it was faster. Figure C.3 in Appendix C shows that all algorithms have a marked improvement in correlation. Figure C.4 compares the execution times visually.

5.2.2 Energy

Quantity	**Value (kJ)**
E_p	18.2
E_n	5.5
E_m	23.5

Table 5.4: Energy values in kJ for E_p, E_n, and E_m, where E_n refers to the energy of the noise (which has been attenuated) in the mixed signal.

Table 5.4 shows that $E_m = 23.5kJ$. The ECA and CGLS algorithm give $E_s = 19.39kJ$. Therefore, the cancellation reduced the energy from $23.5kJ$ to $19.6kJ$. This is just above E_p. Grad desc has more $E_s = 19.40$ because the solution did not converge, hence not achieving maximum cancellation that ECA and CGLS did.

5.2.3 Spectrum

Figure C.5 shows each signal's ESD individually, which can be found in Appendix C. It can be seen if figure 5.2 that energy in frequencies between 10^2 and 10^3 (energy that came from the noise) was attenuated in the the suppressed plot compared to the mixed plot.

5.3 Compare Results of Tests

Test 1 improved correlation from 0.78 to 0.98 . Test 2 improved the correlation from 0.88 to 0.96 . Test 1 had more effective cancellation. This shows that the system is more effective on math signals than music recordings. This makes sense as the math signals are more orthogonal to each other than the music signals. This means they are

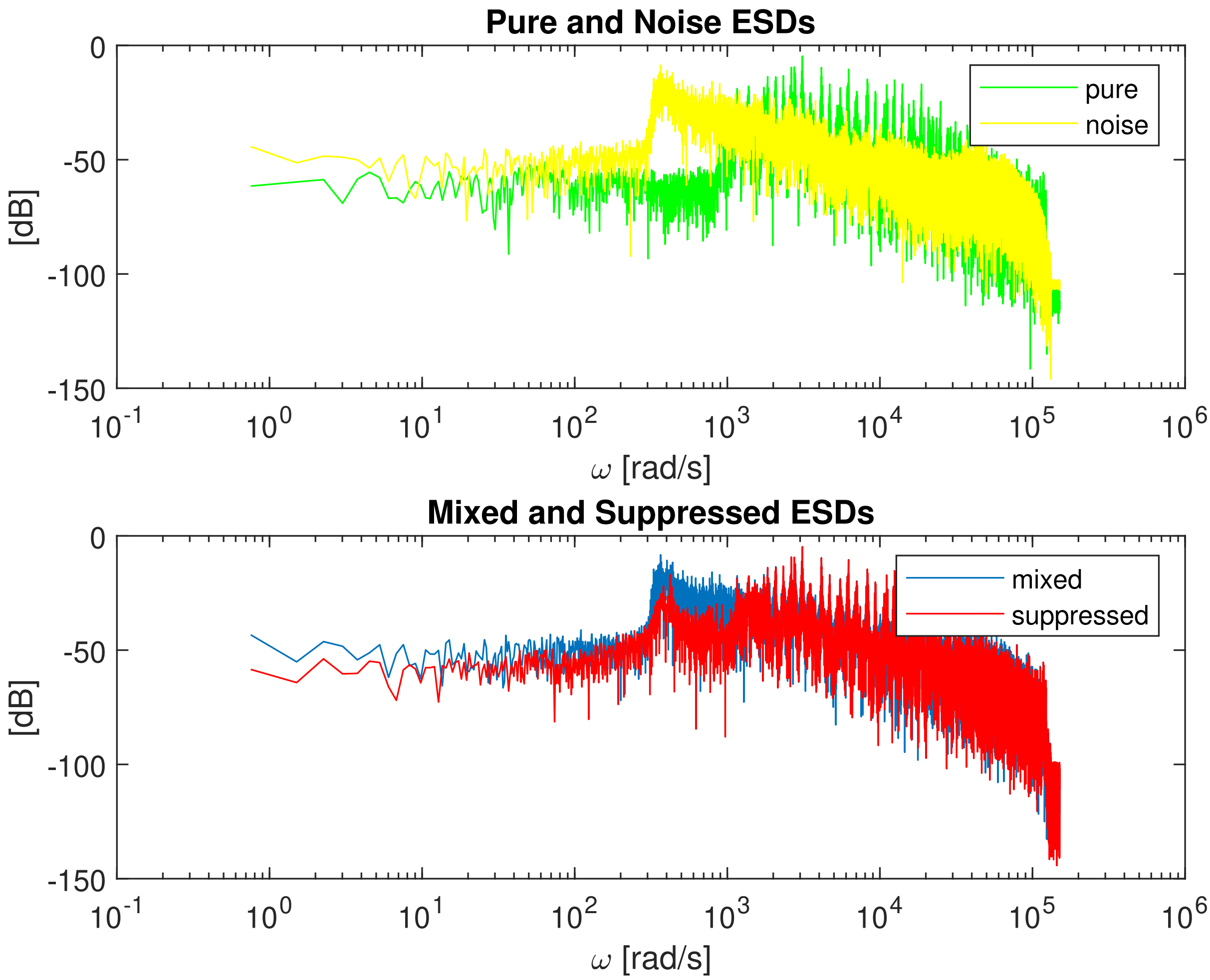

Figure 5.2: Signal ESDs compared.

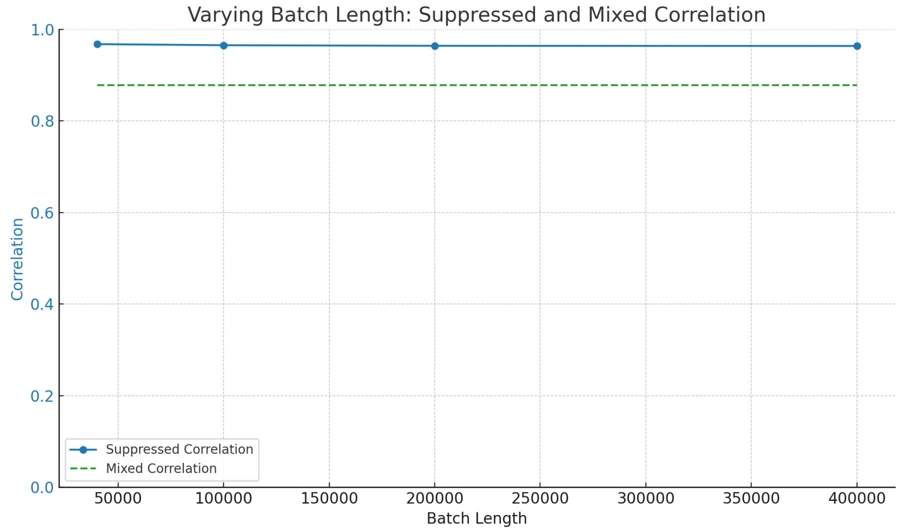

Figure 5.3: Correlations with varying batch lengths

less correlated, hence the correct delay is more heavily weighted in x. This leads to more accurate cancellation.

5.4 Varying Batch Length

The following experiments, varying batch length and number of bins, were performed using the ECA algorithm. This is because ECA was shown to be the most effective algorithm in table 5.3 . Delay range was from 100 to 1100. Batch length was varied from 40 000 to 400 000. Since the audio length is 400 000, this is the same as no batching. Figure 5.3 shows that using any batch length produces good improvement in

Batch Length	Number of Bins	Correlation	Execution Time (s)
40 000	1 000	0.968	18.43
100 000	1 000	0.965	17.22
200 000	1 000	0.964	19.92
400 000	1 000	0.964	19.97

Table 5.5: Results with varying Batch Length

correlation. Table 5.5 shows that more batching (smaller batch sizes) produced slightly better correlation.

Figure 5.4 shows that the best batch length is around 100000. Table 5.5 shows that using a batch length of 100000, would result in an execution time 17.2 seconds compared 20.0 seconds without batching. Therefore, using 4 batches may be slightly more ideal than

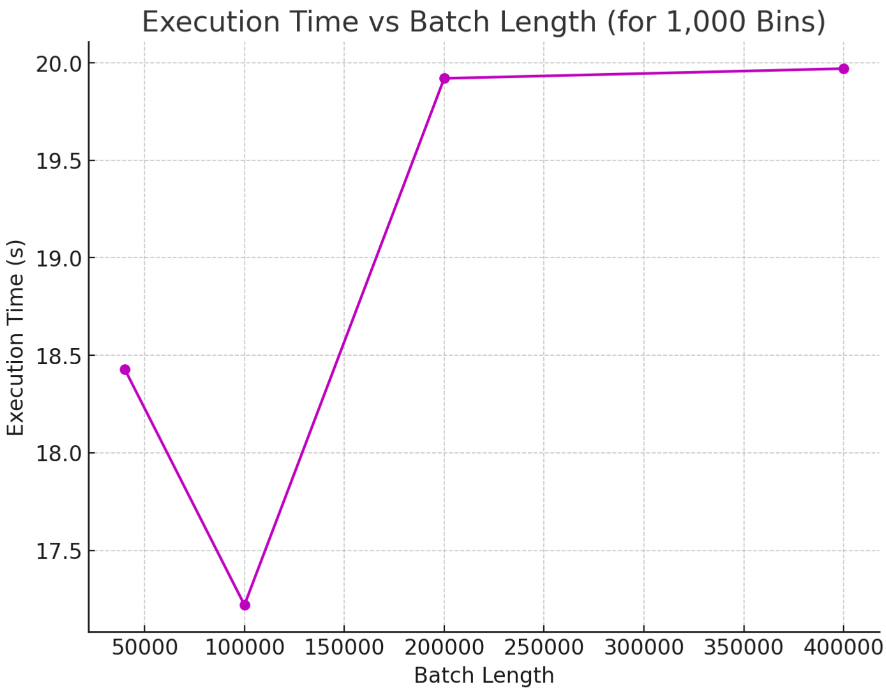

Figure 5.4: Execution time with varying batch lengths

just 1.

5.5 Varying Number of Bins

For this test, batch length was set to 100 000 because it performed best in the previous test. The actual delays are 500 and 800. `min_delay` was set to 100. The number of bins was varied from 500 to 2000. Figure 5.5 shows that using any number of bins produces

Batch Length	Number of Bins	Correlation	Execution Time (s)
100 000	500	0.954	7.21
100 000	1 000	0.965	16.74
100 000	1 500	0.964	30.78
100 000	2 000	0.963	48.26

Table 5.6: Results with varying Number of Bins

good improvement in correlation. This makes sense as actual delays were known in this controlled setting, and at least one of those delays was covered in each execution. Table 5.6 shows that the first run had a lower correlation. This is because it only covered up to a delay of 600 (not covering one of the actual delays: 800). It was expected that execution time would increase with increase in number of bins because it means increase in number of columns of N, hence more computations. However, figure 5.6 shows that time increases

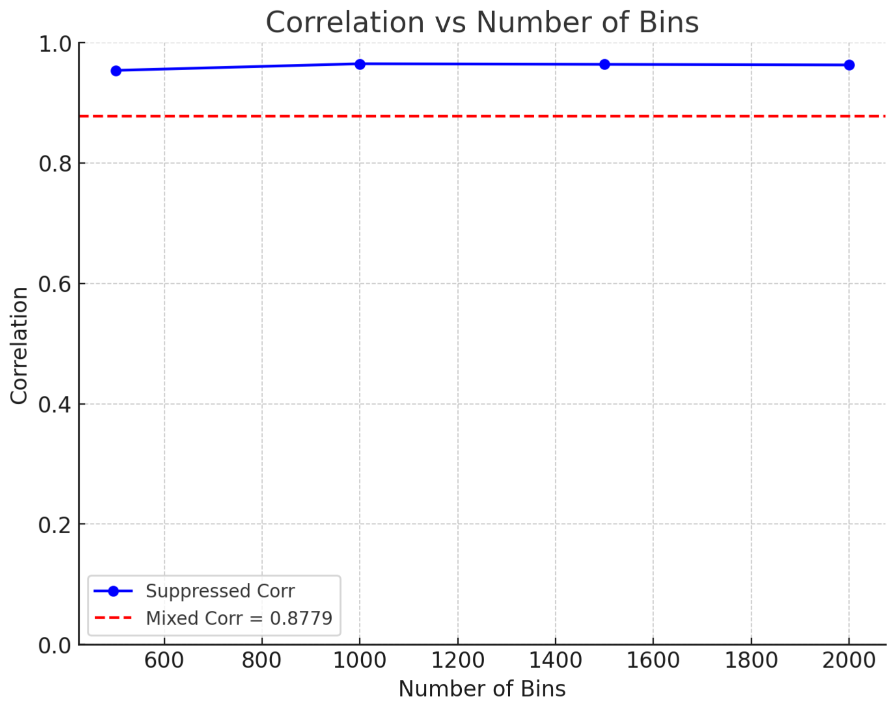

Figure 5.5: Correlations with varying number of bins

greater than $O(n)$. This is because the number of computations is dependent on number of bins by $O(n^2)$ as discussed in subsection 3.3.2.

Figure 5.6: Execution time with varying number of bins

Chapter 6

Conclusions

The goal of this study was to investigate the effectiveness of ECA, Gradient Descent and CGLS on audio data. The algorithms were tested on two different types of data: white noise, and music. This was done using several metrics: correlation, energy, and spectral plots. It was found that all the algorithms were effective, which was measured using correlation of signals. ECA was found to be the most efficient, executing in the fastest time in all tests. ECA-B was found to be slightly more efficient than ECA without batching. The comparison of algorithms resulted as in other literature. The most important discovery from this study is that passive radar algorithms are effective on audio cancellation. However, the cancellation could be improved by using more complex algorithms such as ECA-C used by Hicks [3] or the PEM-FDKF algorithm used by Bernardi [6].

It was also confirmed that increasing the number of bins, increases the execution time quadratically. This is an important consideration when moving to real recordings because in reality the actual delays are not known. In practice, the range of delays could be much larger than those used in this study, which is another reason a more complex algorithm should be investigated.

6.1 Future Work

Firstly, the algorithms need to be applied to longer audio that is approximately as long as a concert performance. Also, a larger number of bins needs to be experimented with. This would increase computational time significantly; therefore, ECA-C would need to be implemented. If ECA-C is not efficient enough a PEM-based algorithm would need to be implemented.

From there, the algorithms can be tested on real recordings. For this experiment two sources of sound (by a person or a speaker) and two microphones would be required. The microphones would record for the same time and positioned such that one can hear both sounds (emulating the "mixed" microphone) and one that hear just on sound (emulating the "noise" microphone). From there the algorithms can be applied to the mixed recording using the noise recording. If this works, then finally data should be collected at Endler Hall and the algorithms should be applied to that data.

Bibliography

[1] B. Russel, "The least-mean-square (lms) algorithm and its geophysical applications," CREWES, Tech. Rep., 2017.

[2] F. Colone, D. W. O'Hagan, P. Lombardo, and C. J. Baker, "A multistage processing algorithm for disturbance removal and target detection in passive bistatic radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 698–722, 2009.

[3] M. N. Hicks, "Direct signal cancellation for dvb-t2 passive radar," Bachelor's Thesis, University of Cape Town, 2018.

[4] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie-Mellon University. Department of Computer Science Pittsburgh, Tech. Rep., 1994.

[5] C. A. Tong, "A scalable real-time processing chain for radar exploiting illuminators of opportunity," Ph.D. dissertation, University of Cape Town, 2014.

[6] G. Bernardi, T. van Waterschoot, J. Wouters, and M. Moonen, "Subjective and objective sound-quality evaluation of adaptive feedback cancellation algorithms," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 1010–1024, 2018.

[7] B. Widrow and M. Hoff, "Adaptive switching circuits," in *IRE WESCON Conv. Rec.*, 1960, pp. 96–104.

Appendix A

Graduate Attributes (GAs)

GA	Requirement	Justification and section in the report
1	Problem-solving	I modified the ECA, Gradient Descent and CGLS algorithms used in passive radar to recorded audio data in section 3.1. These algorithms use Least Mean Squares (LMS). I figured out how to apply the LMS algorithm to this project. These algorithms have been implemented to calculate the weights of a range of delays to the noise data.
4	Investigations, experiments and data analysis	I investigated three algorithms on two types of audio data in section 4.4. I experimented with those algorithms on audio with 400000 samples. I used 1000 bins in the Musc Test in section 5.2. I then varied the number of bins to see how is affected the execution time. I also varied batch length to see how that affected the results.
5	Use of engineering tools	I used Audacity to manufacture audio data with noise by superimposing an audio track on another with a delay. I am using MATLAB to implement the cancellation algorithms. I used Overleaf and Latex to write the report.
6	Professional and technical communication (Long report)	I wrote this professional technical report.
8	Individual work	I did all the project on my own.
9	Independent learning ability	I have searched the internet for literature describing the cancellation algorithms. I have taken knowledge learned from thesis papers and lecture slides and applied the mathematics to this project as shown in chapter 3. At the beginning of this project, I was expecting to find one source of literature to simply describe one algorithm which I could apply. However, I have discovered that bits from multiple sources of literature need to be consolidated in order to understand and apply an algorithm to the project at hand.

Appendix B

AI Usage

Examples of prompts given to chatGPT for generating plots and latex tables are shown here.

```
\textbf{Batch Length}   \textbf{Number of Bins}        \textbf{
   Correlation}    \textbf{Execution Time (s)}
100 000          500               0.954            7.21
100 000          1 000             0.965            16.74
100 000          1 500             0.964            30.78
100 000          2 000             0.963            48.26

plot the corrs as suppressed_corr and plot mixed_corr = .8779 .
   ylims must be 0:1
```

```
tabulate:
'ECA'     0.983087613058433          4.87299800000000
'Grad Desc'        0.983087613103330          5.51655040000000
'CGLS'    0.983087613058433          5.34118300000000
```

Appendix C

Additional Plots

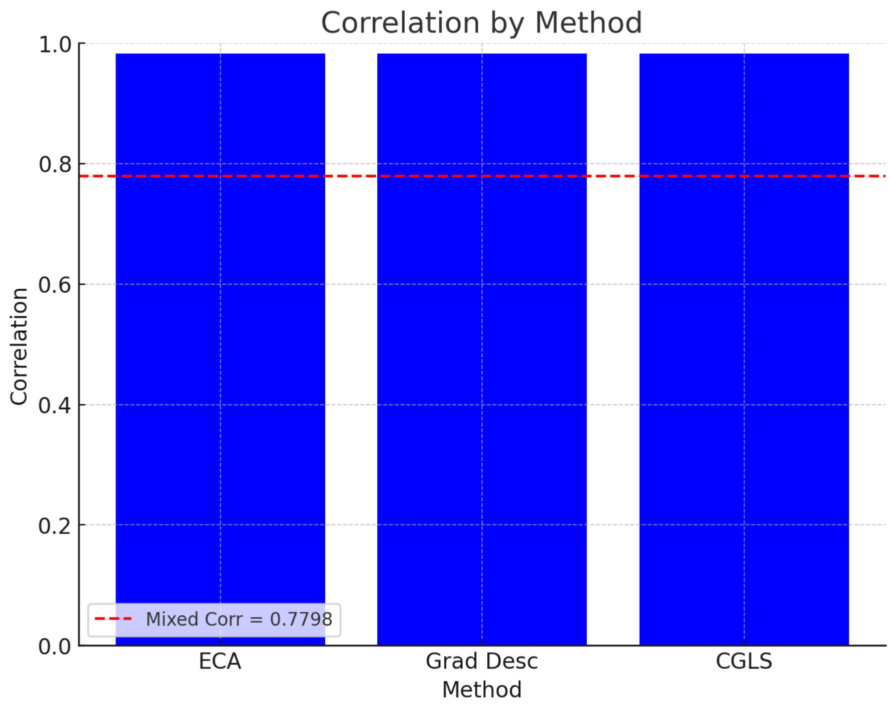

Figure C.1: Test 1 correlation results

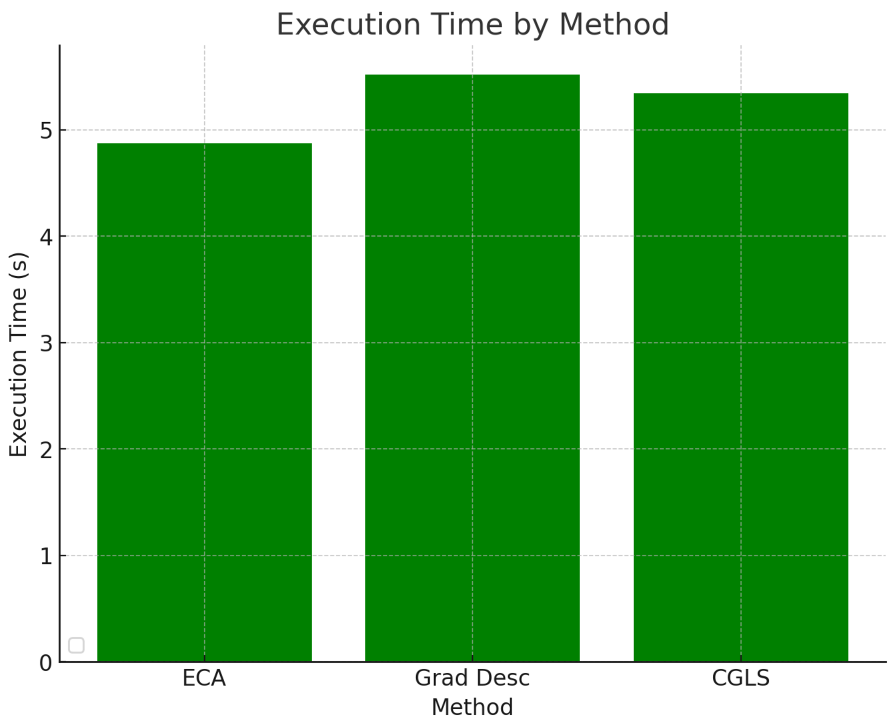

Figure C.2: Test 1 execution times

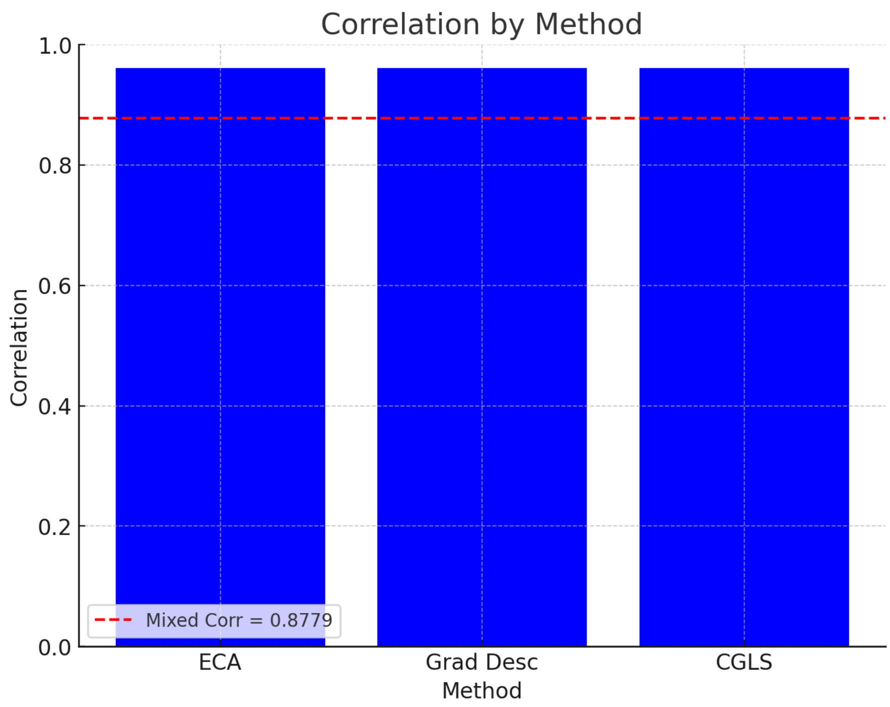

Figure C.3: Test 2 correlation results

Figure C.4: Test 2 execution times

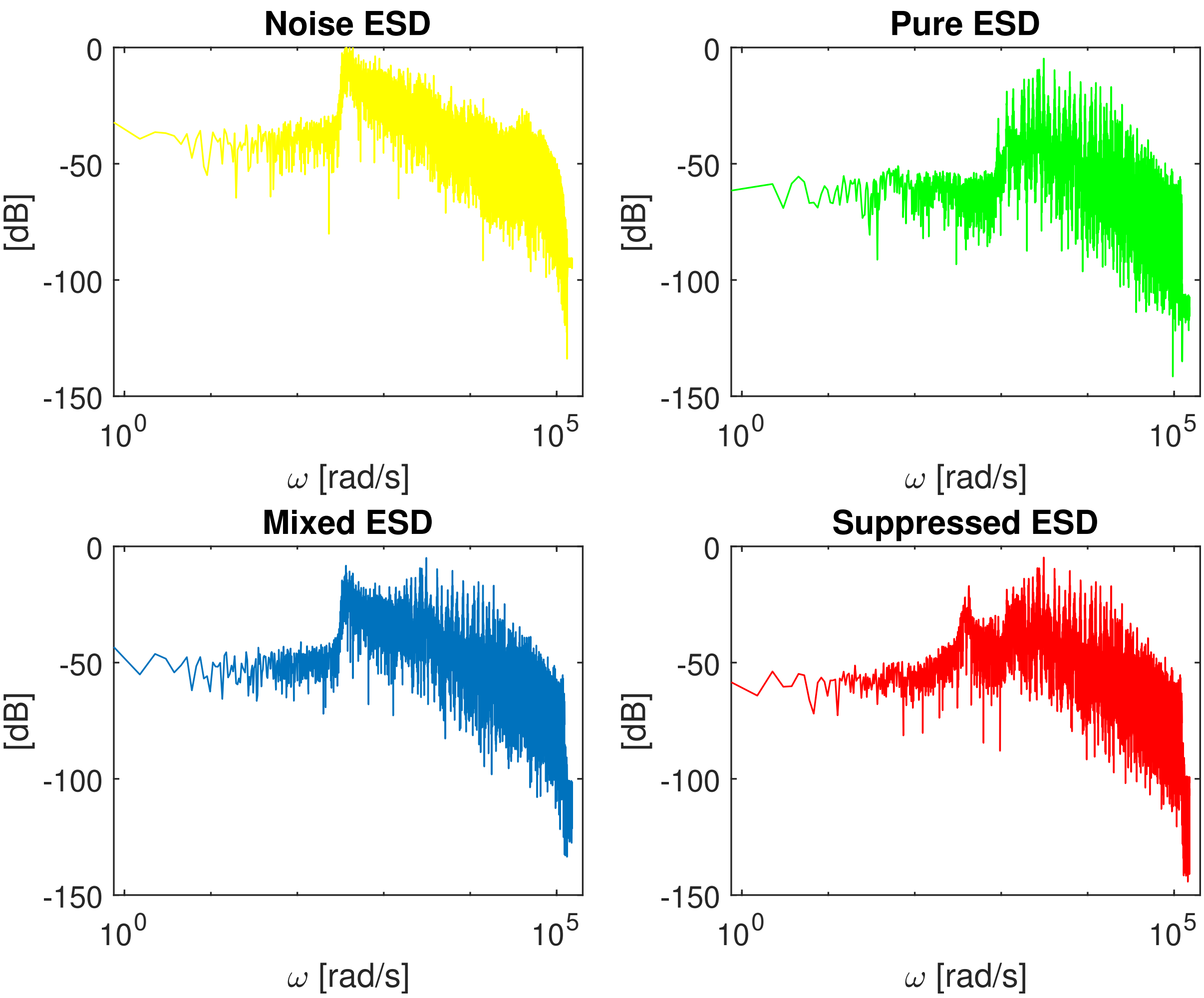

Figure C.5: Each individual signal's ESD in Test 2.

Appendix D

GitHub repository

The project repository containing the Matlab source code and audio files can be found at `https://github.com/karanimaan/Final-Year-Project` .