Multi-Microphone Speech Enhancement

Chang Ge (5727820)
Nadine van Dam (4912527)

1 Introduction

Over the years the amount of usage of mobile speech processing applications has increased. The processing is done in cell phones (especially hands-free calls), hearing aids, and speech recognition [3]. There can be a lot of noise in those applications. The noise is there even while an accurate interpretation of the speech can be necessary. Therefore, it is important to find the underlying signal without noise. When a single microphone is used, the signal quality can already be enhanced by the right estimator, but when multiple microphones are used, the quality of the signal is improved even more. This is because spatial filtering can then be applied [3]. With spatial filtering, it should be taken into account that multiple microphones are at different places. Therefore, the signal does not necessarily need to be present at the same moment as there can be a time shift.
Moreover, it is important to notice that a speech signal is most of the time non-stationary. To tackle this problem, one could use small (overlapping) windows such that the signal can be seen as almost stationary [6].
Lastly, the processing in the frequency domain is more efficient than in the time domain [6]. Therefore, the data will be converted into the frequency domain after the windowing.
After calculating the estimator in the frequency domain it will be converted back to the time domain. The entire workflow can be seen in Figure 1.
This report is structured as follows. Chapter 2 is started with an explanation of the signal model. Then the Cramer Rao Lower Bound (CRLB) will be determined. Afterward, two estimators will be discussed. The first one will assume that the signal is deterministic while in the second case, the signal is taken as a random signal. In Chapter 3 the CRLB will be compared to the variance of the two estimators with different amounts of microphones. Moreover, the time domain results of the estimators are plotted. Lastly, in Chapter 4, the results will be analyzed and some recommendations will be given.

2 Methodology

2.1 Model

In this section, the signal model of the multi-microphones system will be elaborated. Before determining the exact model it is important to have some knowledge about the provided data. The provided data exist of a signal measured on 16 different microphones. The signals recorded by the multiple microphones can be represented through Equation 1, where $m \in \{1, ..., M\}$ denotes the index of microphones, $y_m(n)$, $s_m(n)$, $w_m(n)$ denote the recorded noisy signal, target clean signal, and noise at sample time n respectfully.

$$y_m(n) = s_m(n) + w_m(n) \tag{1}$$

The noise signal $w_m(n)$ is assumed to be White Gaussian Noise (WGN) and is stationary over time. Each noise signals from different microphones are independent, but not identical. This means that each microphone receives noise with a different variance.
As mentioned before speech signals are non-stationary. To be able to assume stationarity it is decided to divide into short-time frames of 20 ms with 50% overlap and windowed with Hanning windows. The Hanning window is used due to its useful properties such as having almost no leakage while preserving most properties in the frequency domain and high amplitude accuracy [5]. Then, the Fourier Transform is taken from the framed signal. The signal recorded by the m_{th} microphone in frequency domain can be modeled by Equation 2, where $l \in \{1, ..., L\}$ denotes the index of time frame and $k \in \{1, ..., K\}$ denotes the index of frequency bin.

$$Y_m(l,k) = S_m(l,k) + W_m(l,k) \tag{2}$$

Taking into consideration the propagation from the signal source to the locations of multiple microphones, the acoustic transfer function (ATF) is defined as $A_m(l,k)$. The signal received by multiple phones at l_{th} time frame and k_{th} frequency bin can be modeled by Equation 5.2.

$$\boldsymbol{Y(l,k)} = \boldsymbol{A(l,k)}S(l,k) + \boldsymbol{W(l,k)} \tag{3}$$

Here, $\boldsymbol{Y(l,k)} = [Y_1(l,k), Y_2(l,k), ..., Y_M(l,k)]^T$,
$\boldsymbol{A(l,k)} = [A_1(l,k), A_2(l,k), ..., A_M(l,k)]^T$ and,
$\boldsymbol{W(l,k)} = [W_1(l,k), W_2(l,k), ..., W_M(l,k)]^T$. Note that, this section is closely related to the knowledge provided by the Signal Processing Systems group [6].

2.2 Classical Methods

The classical method can be used when the signal without noise is assumed to be deterministic. In this case, the signal is assumed to be stationary at a time instance l in a frequency bin k. Therefore, $S(l,k)$ can be seen as a constant value. Hence, the signal model can be seen as a linear Gaussian model.
For a linear Gaussian model, the MVUE, MLE, and BLUE are all equal [4]. Moreover, if the MVUE can be found that is the best estimator. Hence the MVUE will be implemented. Using Equation 5.2, the estimator will be equal to 4 (see the derivation Appendix 5.1). In here $\boldsymbol{C_w(l,k)}$ is equal to the covariance matrix of the noise.
To verify the performance of the estimator the variance of the estimator can be compared to the CRLB. The CRLB can be seen in 5 and is derived in Appendix 5.1.

$$\hat{S}(l,k) = \frac{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1} \boldsymbol{Y(l,k)}}{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1} \boldsymbol{A(l,k)}} \tag{4}$$

$$var(\hat{S}(l,k)) \geq \frac{1}{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1} \boldsymbol{A(l,k)}} \tag{5}$$

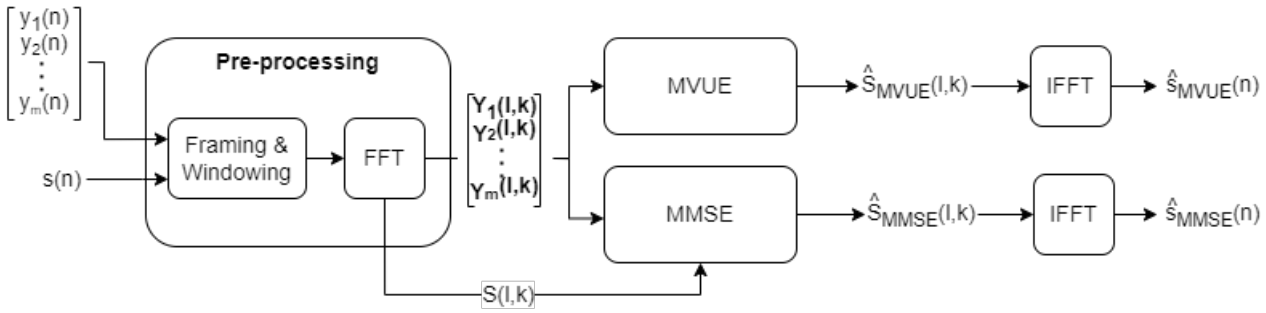

Figure 1: Workflow of the project (based on information from [6]).

2.3 Bayesian Methods

Here the sound signal without noise is no longer assumed to be deterministic but random, and the Bayesian method can be used to estimate the clean signal $S(l,k)$. According to [2] and Figure 4, the clean signal $S(l,k)$ is assumed to be Gaussian distributed, $S(l,k) \sim \mathcal{N}(\mu_s, C_s)$. As the noise signal $W(l,k)$ is assumed to be Gaussian white noise, the signal model in Equation 5.2 is equal to a Bayesian linear model. Thus, the minimum mean square estimator (MMSE), the linear minimum mean square estimator (LMMSE), and the MAP estimator are identical [4] and can be used to estimate S(l,k). The MMSE estimator is shown in Equation 5.2 and is derived in Appendix 5.2.

$$\hat{S}(l,k) = \mu_s + \boldsymbol{C_s A(l,k)}^T (\boldsymbol{A(l,k) C_s A(l,k)}^T + \boldsymbol{C_w})^{-1} (\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}\mu_s) \tag{6}$$

2.4 Performance metric

As mentioned before, the variance of the estimator will show the performance of the model. Hence, it is important to know how the variance can be calculated. The variance can be calculated with Equation 7 [6]. In this equation $S(l,k)$ is the clean data signal.

$$var(\hat{S}(l,k)) = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} |\hat{S}(l,k) - S(l,k)|^2 \tag{7}$$

3 Results

The data provided contains one clean audio signal and noisy audio signals recorded at $M = 16$ microphones. The sampling frequency f_s) is 16 kHz. In the pre-processing step, each audio signal is then divided into 20 ms short time frames containing 320 samples with 50% overlap and windowed by Hanning window. Then the data are transformed into the frequency domain through FFT.

The first second of noisy audio signals is assumed to contain noise only and is used to estimate noise. Since the noise signal are independent, C_w is a $m \times m$ (m is the number of microphones used to estimate the signal) diagonal matrix $C_{w_{ii}} = \sigma^2_{W_{m(k)}}$, where $\sigma^2_{W_{m(k)}}$ is the variance of the noise signal recorded at the m_{th} microphone at k_{th} frequency bin.

Furthermore, it is given that the signals are measured in the far field of the signal source. Hence, the clean signal at each microphone is started almost at the same point. Therefore, the ATF is equal to $\boldsymbol{A(l,k)} = 1_m, \forall l,k$. So it is equal to a vector of ones with a length equal to the number of microphones used.

3.1 Classical Methods

The estimator for the classical method (Equation 4) and the CRLB (Equation 5) are implemented in Matlab (see Appendix 6). In Figure 2, the variance of both the CRLB and MVUE with different amounts of microphones are shown. From the figure, it can be seen that the variance of the MVUE is almost equal to the CRLB. This is expected as for a Gaussian linear model the MVUE reaches the CRLB. Moreover, it can be seen that using more microphones quickly decreases the variance.

To find the percentage of the difference between the CRLB and the variance of the MVUE, Equation 8 can be used. In Figure 3, the error can be seen. The largest error is$\epsilon = 1.2\%$, while most of the errors are less than $\epsilon = 0.4\%$. So the error is indeed quite small. From the graph, two remarkable things can be noticed. Firstly, the error is not equal to $\epsilon = 0\%$. This is because the assumption that the signal will be deterministic is not fully correct. Secondly, it can be seen that the estimator is sometimes below the CRLB. This shouldn't be possible when the signal would have been deterministic in WGN. Moreover, this suggests that there might be better estimators when the signal is not assumed to be deterministic.

$$\epsilon = \frac{var(\hat{S}(l,k)) - CRLB(l,k)}{CRLB(l,k)} * 100 \tag{8}$$

3.2 Bayesian Methods

The provided clean signal is used as prior information. Taking the first frequency band as an example, Figure 4 shows the histogram of the first frequency band of frequency domain clean speech signal. According to the distribution of the frequency domain signal, it is assumed to be Gaussian distributed, $S(l,k) \sim \mathcal{N}(\mu_s, C_s)$. The mean μ_s and covariance C_s of $S(l,k)$ are calculated for each frequency band. Then the Bayesian estimator in Equation 5.2 is implemented (see Appendix 6). The empirical variance using various numbers of microphones is calculated and compared with CRLB obtained in 2.2. The results are shown in Figure 5.

As shown in Figure 5, the empirical variance between estimated signal $\hat{S}(l,k)$ and clean signal $S(l,k)$ is smaller than CRLB. This is because the use of prior information improves the performance of the estimator. Also, as more and more

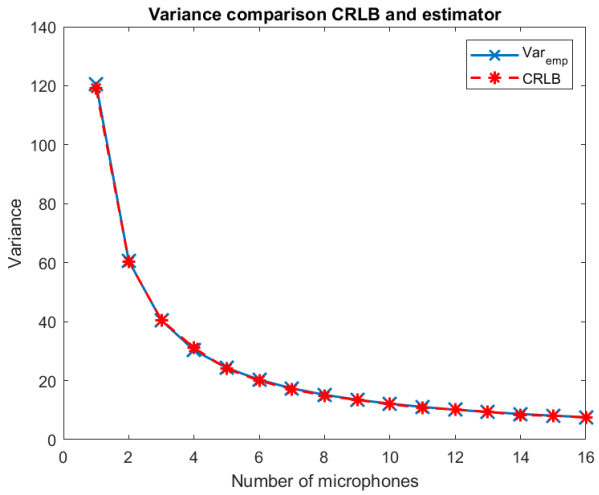

Figure 2: The CRLB and the variance of the MVU with different amounts of microphones used.

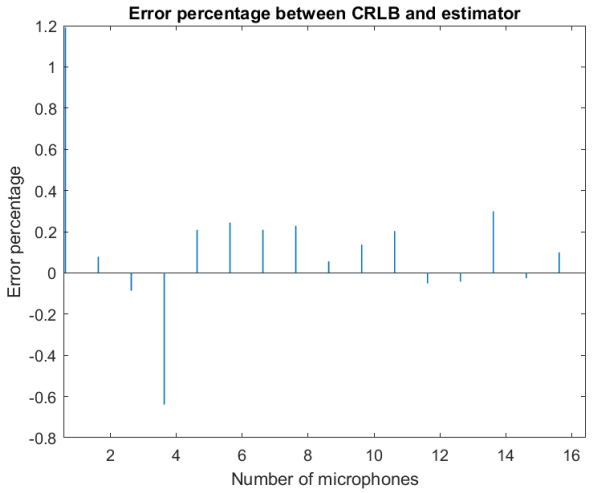

Figure 3: Error in percentage between the CRLB and the variance of the MVUE with different amount of microphones used.

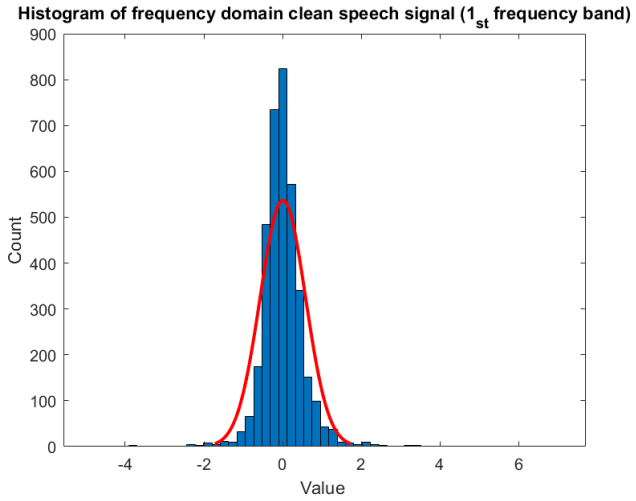

Figure 4: Histogram of the first frequency band of clean signal in the frequency domain.

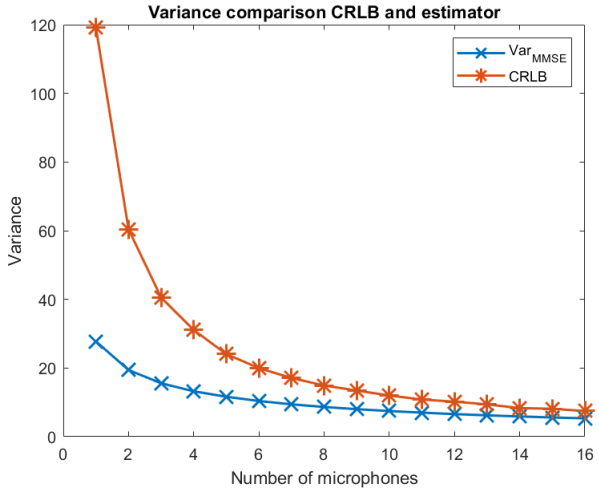

Figure 5: The CRLB and the variance of the LMMSE with different amounts of microphones used.

microphones are used to estimate $\hat{S}(l,k)$, the empirical variance is getting closer to CRLB. This is because by increasing the number of data points, the estimated result becomes less dependent on prior information[4].

3.3 Time domain

When the estimators are converted back to the time domain, then a comparison can be made with the clean and noisy speech signal. It is decided to focus on the MVUE and MMSE for both 1 microphone and 16 microphones. The results are shown in Figure 6. It can be seen that the MVUE with 1 microphone still has much noise and therefore it is closer to the noisy signal than to the clean signal. When the MVUE with 16 microphones is used, then the estimator is close to the clean signal. The MMSE estimator with 1 microphone relies a lot on the prior information. This can be noticed as the signal is smoothed out a bit too much. Hence, some parts of the clean signal are almost missed. Lastly, the MMSE with 16 microphones is again closer to the clean speech signal than the MMSE with 1 microphone. This is since the estimator with 16 microphones relies more on the pdf of the clean signal than the 1 microphone estimator does. The quality of the MVUE with 16 microphones and MMSE with 16 microphones seems similar. The MVUE with 1 microphone performs a lot worse than the MMSE with 1 microphone (as expected from the variance in the frequency domain).

4 Conclusion

After being transformed into the frequency domain, the multi-microphone audio model can be regarded as a linear Gaussian model. When the target signal $S(l,k)$ at k_{th} frequency bin and l_{th} time frame is assumed to be deterministic, the MLE and BLUE are equivalent to MVUE and so these can be used to estimate $S(\hat{l},k)$. When the variance of the estimator is compared to the variance calculated by the CRLB then this is almost equal. The error can be justified as the clean signal is not deterministic. Furthermore, from the variance of this estimator, it can be seen that the variance with a few microphones

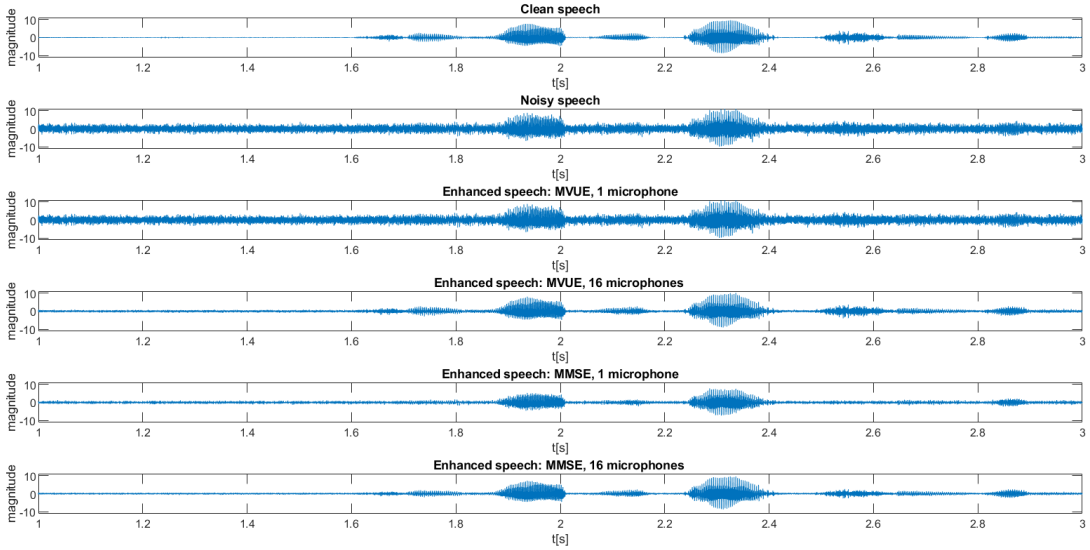

Figure 6: The clean signal, noisy signal, MVUE, and MMSE for both 1 and 16 microphones in the time domain.

is a lot larger than when more microphones are used. This is as expected as with more signals spatial filtering can be done. When the target signal $S(l,k)$ is assumed to be the random signal, the provided clean audio signal is assumed to be Gaussian distributed and is used to calculate prior information. The multi-microphone audio model can be regarded as a Bayesian linear model. LMMSE and MMSE estimator is identical in this case. The variance of this estimator is especially with fewer microphones a lot lower than the CRLB. When more microphones are used, then more points will be used to determine the signal. Therefore, the estimator will less depend on the prior information and more on the pdf. Hence, when more microphones are used it will become closer to the estimators of the deterministic case (and so to the CRLB).

In the time domain it can be seen that the performance of the MVUE with 1 microphone is not sufficient. This conclusion can be taken as it is very close to the noisy signal instead of to the clean signal. The MMSE with 1 microphone seems to smooth out the signal. Therefore, some parts of the clean signal are missed. Still, this estimator would be preferred above the MVUE estimator. When more microphones are used both estimators converge to the clean signal. With 16 microphones the performance is almost equal.

In conclusion, when not many microphones are used it is better to use the MMSE. On the other hand, when a lot of microphones are used there isn't a large difference anymore.

In later research, the Bayesian estimator could be improved by taking the Laplacian distribution instead of the Gaussian distribution as signal pdf. Moreover, the noise is assumed to be stationary over time but this is most of the time not true.

[1]

References

[1] Christopher & Dana Reeve Foundation. Causes of paralysis, 2022.

[2] S. Gazor and Wei Zhang. Speech probability distribution. *IEEE Signal Processing Letters*, 10(7):204–207, 2003.

[3] Richard C Hendriks and Timo Gerkmann. Noise correlation matrix estimation for multi-microphone speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):223–233, 2011.

[4] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Fundamentals of Statistical Signal Processing. Prentice-Hall, 1993.

[5] LDS-group. Understanding fft windows, 2014.

[6] Signal Processing Systems group. Et4386 estimation and detection assignment multi-microphone speech enhancement, 2022.

5 Derivations

In this appendix, the derivations for the estimator formulas are shown. First, the derivation in the deterministic case will be given. Afterward, the estimator for the random case will be derived.

5.1 Classical method

As mentioned in Chapter 2 the MVUE, MLE and BLUE are equal. Therefore, only the MVUE will be derived here. This section ends with the derivation of the CRLB.
The MVUE and MLE derivations start with the pdf. For a linear Gaussian model that is equal to

$$p(\boldsymbol{Y(l,k)}; S(l,k)) = \frac{1}{(2\pi)^{N/2} det(\boldsymbol{C_w(l,k)})^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{Y(l,k)}-\boldsymbol{A(l,k)}S(l,k))^T \boldsymbol{C_w(l,k)}^{-1}(\boldsymbol{Y(l,k)}-\boldsymbol{A(l,k)}S(l,k))}$$

The next step is to take the natural logarithm of the pdf.

$$\ln(p(\boldsymbol{Y(l,k)}; S(l,k))) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(det(\boldsymbol{C_w(l,k)})) - \frac{1}{2}(\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}S(l,k))^T \boldsymbol{C_w(l,k)}^{-1}(\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}S(l,k))$$

To find the score function, the derivative with respect to $S(l,k)$ should be taken.

$$s(\boldsymbol{Y(l,k)}; S(l,k)) = \frac{\partial \ln(p(\boldsymbol{Y(l,k)}; S(l,k)))}{\partial S(l,k)} = \boldsymbol{A(l,k)C_w(l,k)}^{-1}(\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}S(l,k))$$

For the MVUE the fisher information needs to be found, which is the derivative of the score times minus 1.

$$I(S(l,k)) = -\frac{\partial s(\boldsymbol{Y(l,k)}; S(l,k))}{\partial S(l,k)} = --\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A(l,k)} = \boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A(l,k)}$$

The next step is rewriting the score function in the form $s(\boldsymbol{Y(l,k)}; S(l,k)) = I(S(l,k))(g(\boldsymbol{Y(l,k)}) - S(l,k))$.

$$s(\boldsymbol{Y(l,k)}; S(l,k)) = \boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}(\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}S(l,k)) = \boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A(l,k)}(g(\boldsymbol{Y(l,k)}) - S(l,k)$$

So now $g(\boldsymbol{Y(l,k)})$ can be calculated.

$$\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A}g(\boldsymbol{Y(l,k)}) = \boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{Y(l,k)}$$
$$g(\boldsymbol{Y(l,k)}) = \frac{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{Y(l,k)}}{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A(l,k)}}$$

The estimator $S(\hat{l},k)$ will be equal to $g(\boldsymbol{Y(l,k)})$.

The next step will be deriving the CRLB. As the fisher information is already known this is only substituting.

$$var(\hat{S}(l,k)) \geq \frac{1}{I(S(l,k))} = \frac{1}{\boldsymbol{A(l,k)}^T \boldsymbol{C_w(l,k)}^{-1}\boldsymbol{A(l,k)}}$$

5.2 Bayesian method

For the linear Bayesian model,

$$\boldsymbol{Y(l,k)} = \boldsymbol{A(l,k)}S(l,k) + \boldsymbol{W(l,k)}$$

Where $S(l,k)$ target variable, $S(l,k) \sim \mathcal{N}(\mu_s, C_s)$, and $\boldsymbol{W(l,k)}$ is white Gaussian noise,$\boldsymbol{W(l,k)} \sim \mathcal{N}(0, C_w)$. The MMSE, LMMSE, and MAP estimators are identical. The MMSE estimator is,

$$\begin{aligned} S(\hat{l},k) &= E[S(l,k)|\boldsymbol{Y(l,k)}] && (9)\\ &= E[\boldsymbol{Y(l,k)}] + C_{YS}C_{YY}^{-1}(\boldsymbol{Y(l,k)} - E[\boldsymbol{Y(l,k)})] && (10)\end{aligned}$$

Here,

$$\begin{aligned} E[\boldsymbol{Y(l,k)}] &= E[\boldsymbol{A(l,k)}S(l,k) + \boldsymbol{W(l,k)}] \\ &= \boldsymbol{A(l,k)}E[S(l,k)] + E[\boldsymbol{W(l,k)}] \\ &= \mu_s \end{aligned}$$

$$\begin{aligned} C_{YY} &= E[(\boldsymbol{Y(l,k)} - E[\boldsymbol{Y(l,k)}])(\boldsymbol{Y(l,k)} - E[\boldsymbol{Y(l,k)}])^T] \\ &= E[(\boldsymbol{A(l,k)}S(l,k) + W(l,k) - \boldsymbol{A(l,k)}\mu_S)(\boldsymbol{A(l,k)}S(l,k) + \boldsymbol{W(l,k)} - \boldsymbol{A(l,k)}\mu_S)^T] \\ &= \boldsymbol{A(l,k)}E[(S(l,k) - \mu_S)(S(l,k) - \mu_S)^T]\boldsymbol{A(l,k)^T} + E[\boldsymbol{W(l,k)W(l,k)^T} \\ &= \boldsymbol{A(l,k)}C_S\boldsymbol{A(l,k)^T} + \boldsymbol{C_w} \end{aligned}$$

$$\begin{aligned} C_{YS} &= E[(\boldsymbol{Y(l,k)} - E[\boldsymbol{Y(l,k)}])(S(l,k) - E[S(l,k)])^T] \\ &= E[(\boldsymbol{A(l,k)}S(l,k) + W(l,k) - \mu_S)(S(l,k) - \mu_S)] \\ &= E[\boldsymbol{A(l,k)}(S(l,k) - \mu_S)(S(l,k) - \mu_S)^T] \\ &= \boldsymbol{A(l,k)}C_S \end{aligned}$$

Substitute the variables in equation 9, the final formula for the MMSE estimator is as follow,

$$\hat{S}(l,k) = \mu_s + \boldsymbol{C_s}\boldsymbol{A(l,k)}^T(\boldsymbol{A(l,k)}\boldsymbol{C_s}\boldsymbol{A(l,k)}^T + \boldsymbol{C_w})^{-1}(\boldsymbol{Y(l,k)} - \boldsymbol{A(l,k)}\mu_s)$$

6 Matlab

In this section, the Matlab code is added. It starts with the code which is equal to the main body. Then the different functions are added.

```
clear
clc
close all
load('Data.mat');

%% Transfer signals into frequency domain
%20ms frame length, 50% overlap, hann window
fs = 16000;
t_frame = 0.020 ;        %20ms window size
L_frame = t_frame *fs;
hann_win = hanning(L_frame);
L_noise = 1*fs;          %assume 1st second is noise only
noise_audio = Data(1:L_noise,:);
audio = Data(L_noise+1:end,:);
clean_audio = Clean(L_noise+1:end,:);
clean_fft = enframe(clean_audio,L_frame);
%concatenate the fft results
for i = 1:nrmics
    audio_fft_1 = enframe(audio(:,i),L_frame);
    noise_fft_1 = enframe(noise_audio(:,i),L_frame);
    if i == 1
        audio_fft = audio_fft_1;
        noise_fft = noise_fft_1;
    else
        audio_fft = cat(3,audio_fft,audio_fft_1);
        noise_fft = cat(3,noise_fft,noise_fft_1);
    end
end
%% Calculate noise variance
var_est = var_estimate(noise_fft);
%% Determine the estimator and calculate variance
nrmics = 16;
var = zeros(nrmics,1);
for j =1:nrmics
    estimator = mvue(audio_fft,var_est,j);
    [L,K] = size(estimator);
    var(j) = sum(abs(estimator-clean_fft).^2,'all')/(K*L);
end
%% CRLB
crlb_mic = zeros(nrmics,1);
for ii = 1:nrmics
    crlb_fre = crlb(audio_fft,noise_fft,ii);
    crlb_mic(ii) = mean(crlb_fre);

end
%% Graphs variance
mic = 1:16;
figure(1),
plot(mic,var,'-x','LineWidth',1.5,'MarkerSize',12),
hold on
plot(mic,crlb_mic,'--*','LineWidth',1.5,'MarkerSize',8,'Color',[1, 0, 0, 0.5])
title('Variance comparison CRLB and estimator')
xlabel('Number of microphones')
ylabel('Variance')
legend('Var_{emp}','CRLB')
```

```
hold off;
figure(2);
bar(mic,100*(var(:,1)-crlb_mic)/crlb_mic)
title('Error percentage between CRLB and estimator')
xlabel('Number of microphones')
ylabel('Error percentage')
hold off;

function audio_fft = enframe(audio,L_frame)
    %convert signal into frequency domain
    [L_audio,~] = size(audio);
    step = 0.5 * L_frame;
    n_frame = floor((L_audio - (L_frame - step))/step);
    indf = step*(0 : n_frame - 1);
    inds = (1 : L_frame )';
    index = repmat(indf, L_frame, 1) + repmat(inds, 1, n_frame);
    audio_frame = audio(index);
    [L_frame, t] = size(audio_frame);
    audio_fft = zeros(size(audio_frame));
    for i = 1 : t
        audio_fft(:, i) = fft(audio_frame(:, i) .* hanning(L_frame), L_frame);
    end
    audio_fft =  audio_fft';
end

function var_est = var_estimate(noise_only)
    %calculate the variance of the noisy signal part
    [~,K,n_mic] = size(noise_only);
    var_est = zeros(n_mic,K);
    for i=1:n_mic
        for k=1:K
            var_est(i,k) = var(noise_only(:,k,i));
        end
    end
end

function estimator = mvue(signal,var_est, mic_used)
    %implementation of the MVUE
    [L, K, ~] = size(signal);
    s_hat = zeros(L,K);
    A = ones(mic_used,1); %vector with all ones with length of microphones used
    A_T = transpose(A);
    var_freq = zeros(mic_used,K);
    for k = 1:K
        %create the noise covariance matrix
        var_freq(:,k) = var_est(1:mic_used,k);
        C = diag(var_freq(:,k));
        Cinv = inv(C);
        for l = 1:L
            signal_moment = transpose(reshape(signal(l,k,1:mic_used),1,[]));
            s_hat(l,k)=((A_T*Cinv*A)^(-1))*(A_T*Cinv*signal_moment);
        end
    end
    estimator = s_hat;
end

function min_var = crlb(signal,noise_only,mic_used)
    %calculates the CRLB
    [~, signal_2, ~] = size(signal);
    A = ones(mic_used,1);
```

```
    A_T = transpose(A);
    for k = 1:signal_2
        var_freq(k) = var(noise_only(:,k,mic_used));
        C = diag(var_freq(k));
        Cinv = inv(C);
        min_var(k) = 1/(A_T*Cinv*A);
    end
end

function s_estimate = MMSE(clean_fft,audio_fft,noise_fft,n_mic)

mu_s = mean(clean_fft,1);
var_s = var(clean_fft,1);
C_s = var_s;
var_w = squeeze(var(noise_fft,0,1));
A_m = ones(n_mic,1);
[L,K]= size(clean_fft);
s_estimate = zeros(L,K);
for k = 1:K
    C_w = diag(var_w(k,1:n_mic));
    C = inv(A_m.*C_s(k).*(A_m')+C_w);
    for l = 1:L
        Y_lk = squeeze(audio_fft(l,k,1:n_mic));
        s_estimate(l,k)= mu_s(k)+ C_s(k).*(A_m')*C*(Y_lk-A_m.*mu_s(k));
    end
end

end
```