Delft University of Technology

EE4715 Array processing

Speech Enhancement

Authors:
Weijia Yi (5511127) & Xuan Gao (5232481)

April 28, 2023

1 Objective

This project aims to design and build a multi-microphone speech enhancement/beamforming system for far-end noise reduction. It contains following requirements:

- Generate signals according to the signal model discussed in class using the given audio files and impulse responses.
- Build a multi-microphone speech enhancement system for noise reduction.
- Perform an evaluation of the speech enhancement system.

There are five speech signals, containing two clean signals and three different types of noise signals (babble noise, artificial non-stationary noise, and stationary speech shaped noise). The impulse responses have five sets and each set contains the impulse responses from a source location to the four microphone locations. All datasets are provided from array processing course: Project 2: Speech enhancement. The references of the implementation algorithm are course slides [1]. The evaluation method is referred to [2].

2 Signal model

The received signals by a single microphone can be modeled as

$$x[n] = \sum_{i=1}^{d} (h_i * s)[n] + n[n].$$

The received speech signals with noise are considered as the sum of additive noise signals n and the convolution of the input clean signal s and the impulse responses h of different channels. Take one channel as the target signal (in practical the early reflection), while signals from other channels are all considered as interference, then the model can be rewritten as

$$x[n] = \underbrace{\sum_{i=1}^{d} (h_{target} * s_i)[n]}_{\text{target}} + \underbrace{\sum_{i=1}^{d} (h_{inter} * s_i)[n]}_{\text{interference}} + n[n]. \tag{1}$$

The additive noise utilized in this project includes artificial non-stationary noise, babble noise and speech-shaped noise. Since the artificial noise signal is not perceivable, we amplify it to make it detectable. The noisy speech signals are showed in Figure 1.

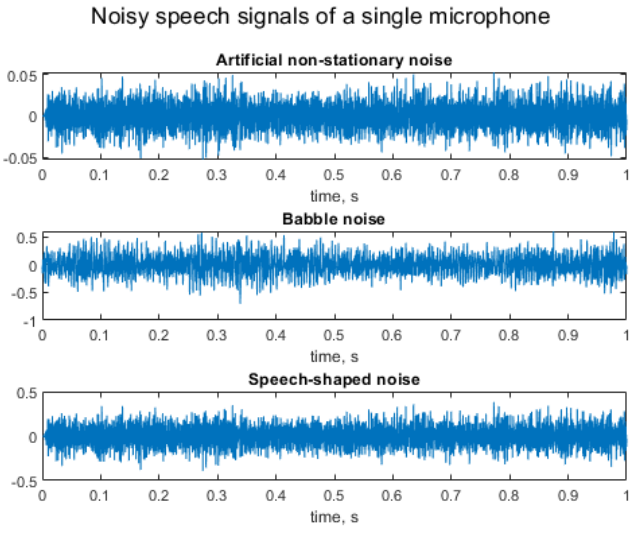

Figure 1: Noisy speech signals of a single microphone with 3 types of additive noise

3 Multi-microphone speech enhancement system implement

3.1 STEP 1: short-time stationary Fourier transform (STFT)

As speech signals change over time, the spectral and temporal characteristics of the speech waveform are time-varying. Therefore, only short segments of speech waveform can be assumed to have similar acoustic properties. This is what it means by "short-term stationary". Speech signals are typically assumed stationary over 20-30ms.
The processing is done in STFT domain, which we do fast Fourier transform (FFT) on short windowed time frames. All time frames obey short time wide-sense stationary (WSS) assumption.
For M microphones, speech signals with noise and noise signals are stacked vector notation as Equation 2. In our project, we use hamming window of the size 320. The number of frames is 6718 and the size of each frame is 320.

$$\begin{aligned}\mathbf{x}(k,l) &= \underbrace{\mathbf{a}_1(k,l)s_1(k,l)}_{\text{target}} + \underbrace{\sum_{i=2}^{d}\mathbf{a}_i(k,l)s_i(k,l) + \mathbf{n}'(k,l)}_{\text{interference+noise}} \\ &= \mathbf{a}(k,l)s(k,l) + \mathbf{n}(k,l)\end{aligned} \tag{2}$$

3.2 STEP 2: Estimation of $\mathbf{R_s}$ using GEVD

Firstly, we estimate $\mathbf{R_x}$ and $\mathbf{R_n}$, then we use generalized eigenvalue decomposition (GEVD) to estimate $\mathbf{R_s}$. The autocorrelation sequence r_x is calculated as $r_x(k,l) = E\{x(k)x^*(l)\}$. Therefore,

$$r_x = [r_x(-3), r_x(-2), r_x(-1), r_x(0), r_x(1), r_x(2), r_x(3)]$$

The autocorrelation matrix $\mathbf{R_x}$ is a Hermitian and Toeplitz matrix and shown in Equation 3.

$$\mathbf{R}_x = E\left\{\mathbf{x}\mathbf{x}^H\right\} = \begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & r_x(3) \\ r_x(1) & r_x(0) & r_x(1) & r_x(2) \\ r_x(2) & r_x(1) & r_x(0) & r_x(1) \\ r_x(3) & r_x(2) & r_x(1) & r_x(0) \end{bmatrix} \tag{3}$$

The autocorrelation sequence is $r_n = [r_n(-3), r_n(-2), r_n(-1), r_n(0), r_n(1), r_n(2), r_n(3)]$. The autocorrelation matrix $\mathbf{R_n}$ is a Hermitian and Toeplitz matrix and shown in Equation 4.

$$\mathbf{R}_n = E\left\{\mathbf{n}\mathbf{n}^H\right\} = \begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & r_n(3) \\ r_n(1) & r_n(0) & r_n(1) & r_n(2) \\ r_n(2) & r_n(1) & r_n(0) & r_n(1) \\ r_n(3) & r_n(2) & r_n(1) & r_n(0) \end{bmatrix} \tag{4}$$

Next, we apply GEVD to $\mathbf{R_x}$ and $\mathbf{R_n}$. As $\mathbf{R_n} = \mathbf{U}^{-H}\Lambda_\mathbf{n}\mathbf{U}^{-1}$ and $\mathbf{R_x} = \mathbf{U}^{-H}\Lambda_\mathbf{x}\mathbf{U}^{-1}$, therefore,

$$\mathbf{U}^H\mathbf{R_x}\mathbf{U} = \text{diag}\,(a_1, \ldots, a_n) \quad \text{and} \quad \mathbf{U}^H\mathbf{R_n}\mathbf{U} = \text{diag}\,(b_1, \ldots, b_n) \tag{5}$$

Hence, we have $\mathbf{R_n}\mathbf{U} = \mathbf{U}^{-H}\mathbf{\Lambda}_{R_n}$ so that

$$\mathbf{R_x}\mathbf{U} = \mathbf{U}^{-H}\mathbf{\Lambda}_{\mathbf{R_x}} = \mathbf{U}^{-H}\mathbf{\Lambda}_{\mathbf{R_n}}\mathbf{\Lambda}_{\mathbf{R_n}}^{-1}\mathbf{\Lambda}_{R_x} = \mathbf{R_n}\mathbf{U}\mathbf{\Lambda}_{\boldsymbol{nx}} \tag{6}$$

$$\mathbf{R_n^{-1}R_x} = \mathbf{U}\mathbf{\Lambda}_{\boldsymbol{nx}}\mathbf{U^{-1}} \tag{7}$$

Hence, the generalised eigenvalues and eigenvectors of $\mathbf{R_x}$ and $\mathbf{R_n}$ are the eigenvalues and eigenvectors of

the matrix $\mathbf{R_n^{-1}R_x}$.
Since Equation 8, we subtract 1 from the diagonal values of $\boldsymbol{\Lambda_{nx}}$ to get $\boldsymbol{\Lambda_{ns}}$.

$$\mathbf{R_n^{-1}R_x} = \mathbf{R_n^{-1}}\left(\mathbf{R_n} + \mathbf{R_s}\right) = \mathbf{I} + \mathbf{R_n^{-1}R_s} = \mathbf{U}\boldsymbol{\Lambda_{nx}}\mathbf{U^{-1}} = \mathbf{U}\left(\boldsymbol{\Lambda_{ns}} + \mathbf{I}\right)\mathbf{U^{-1}} \tag{8}$$

After we get $\boldsymbol{\Lambda_{ns}}$, $\mathbf{R_s}$ is calculated by Equation 9:

$$\mathbf{R_n^{-1}R_s} = \mathbf{U}\boldsymbol{\Lambda_{ns}}\mathbf{U^{-1}}$$

$$\mathbf{R_s} = \mathbf{R_n}\mathbf{U}\boldsymbol{\Lambda_{ns}}\mathbf{U^{-1}} \tag{9}$$

3.3 STEP 3: Beamforming

To solve out multi-channel wiener beamformer and MVDR beamformer, we first derive the expression of signal distortion and residual noise variance by the mean square error of the beamformer output $\mathbf{w}^H\mathbf{x}$ signals and the desired target signal at the reference microphone s_1:

$$\begin{aligned}\mathrm{E}\left|\mathbf{w}^H\mathbf{x} - s_1\right|^2 &= \mathrm{E}\left|\mathbf{w}^H\mathbf{s} + \mathbf{w}^H\mathbf{n} - s_1\right|^2 \\ &= \mathrm{E}\left|\mathbf{w}^H\mathbf{s} - s_1\right|^2 + \mathrm{E}\left|\mathbf{w}^H\mathbf{n}\right|^2\end{aligned}$$

The first term is the signal distortion and the second is the residual noise variance. First begin with the constraint optimization problem to minimize the distortion constrained by reducing noise power

$$\begin{aligned}\text{minimise} \quad & \mathrm{E}\left|\mathbf{w}^H\mathbf{s} - s_1\right|^2 \\ \text{subject to} \quad & \mathrm{E}\left|\mathbf{w}^H\mathbf{n}\right|^2 \leq c,\end{aligned}$$

where $0 \leq c \leq \sigma_{n_1}^2$, $\sigma_{n_1}^2$ is the noise variance at the reference microphone. The solution of this optimization problem is

$$\mathbf{w}^* = \left(\mathbf{R_s} + \mu\mathbf{R_n}\right)^{-1}\mathbf{R_s}\mathbf{e}_1, \tag{10}$$

where μ is the Lagrange multiplier.

- **Multi-Channel Wiener beamformer**
  When $\mu = 1$, the classical multi-channel Wiener filter is given as

  $$\mathbf{w}_{MWF} = \mathbf{R_x^{-1}R_s}\mathbf{e}_1 \tag{11}$$

  where $\mathbf{e_1} = [1, 0, ..., 0]^T$.

- **MVDR beamformer**
  When $\mu = 0$, it leads to

  $$\mathbf{w}_{MVDR} = \frac{\mathbf{R_n^{-1}a}}{\mathbf{a}^H\mathbf{R_n^{-1}a}} \tag{12}$$

  $\mathbf{a}$ is one column of $\mathbf{Q}$ ($\mathbf{Q} = \mathbf{U^{-H}}$) corresponding to the largest eigenvalue.

Therefore, source signal is reconstructed as Equation 13:

$$\mathbf{s} = \mathbf{w}^H\mathbf{x} \tag{13}$$

3.4 STEP 4: Inverse STFT (ISTFT)

The source signal $\mathbf{s}$ is still in Fourier domain. To convert it back to time domain, ISTFT is applied to $\mathbf{s}$.

4 Evaluation

4.1 Speech intelligibility in bits

Speech intelligibility in bits (SIIB) is an estimate of the amount of information shared between a talker and a listener in bits per second [2]. The communication process consisting of a message $\{\mathbf{M}_t\}$, speech signal $\{\mathbf{X}_t\}$, and degraded speech signal $\{\mathbf{Y}_t\}$ is described by a Markov chain as 14. $\{\mathbf{M}_t\} \rightarrow \{\mathbf{X}_t\}$ is the speech production channel and $\{\mathbf{X}_t\} \rightarrow \{\mathbf{Y}_t\}$ is the environmental channel.

$$\{\mathbf{M}_t\} \rightarrow \{\mathbf{X}_t\} \rightarrow \{\mathbf{Y}_t\} \tag{14}$$

The mutual information rate of the environmental channel is:

$$\begin{aligned} I\left(\{\mathbf{X}_t\};\{\mathbf{Y}_t\}\right) &= \lim_{K\to\infty} \frac{1}{K} I\left(\mathbf{X}^K;\mathbf{Y}^K\right) \\ &= \lim_{K\to\infty} \frac{1}{K} I\left(\tilde{\mathbf{X}}^K;\tilde{\mathbf{Y}}^K\right) \\ &= \lim_{K\to\infty} \frac{1}{K} \sum_{j=1}^{KJ} I\left(\tilde{\mathbf{X}}_j^K;\tilde{\mathbf{Y}}_j^K\right) \end{aligned} \tag{15}$$

The information rate between $\{\mathbf{M}_t\}$ and $\{\mathbf{X}_t\}$ is:

$$\begin{aligned} I\left(\{\mathbf{M}_t\};\{\mathbf{X}_t\}\right) &= \lim_{K\to\infty} \frac{1}{K} \sum_{j=1}^{KJ} I\left(\tilde{\mathbf{M}}_j^K;\tilde{\mathbf{X}}_j^K\right) \\ &= \lim_{K\to\infty} -\frac{1}{K} \sum_{j=1}^{KJ} \frac{1}{2} \log_2\left(1-r_j^2\right) \end{aligned} \tag{16}$$

The mutual information rate of the speech production channel is:

$$I\left(\{\mathbf{M}_t\};\{\mathbf{Y}_t\}\right) \leq \min\left(I\left(\{\mathbf{M}_t\};\{\mathbf{X}_t\}\right), I\left(\{\mathbf{X}_t\};\{\mathbf{Y}_t\}\right)\right) \tag{17}$$

The proposed intelligibility metric combines 15, 16, and 17 to estimate the amount of information shared between $\{\mathbf{M}_t\}$ and $\{\mathbf{Y}_t\}$ in bits per second as shown in Equation 18.

$$\text{SIIB} = \frac{F}{K} \sum_{j=1}^{KJ} \min\left(-\frac{1}{2}\log_2\left(1-r_j^2\right), I\left(\tilde{\mathbf{X}}_j^K;\tilde{\mathbf{Y}}_j^K\right)\right) \tag{18}$$

SIIB has high performance. Given a clean acoustic speech signal and a distorted acoustic speech signal, SIIB estimates the amount of information shared between two signals. In our project, we input the clean acoustic signal and the reconstructed acoustic signal to output the SIIB value.

4.2 Results and analysis

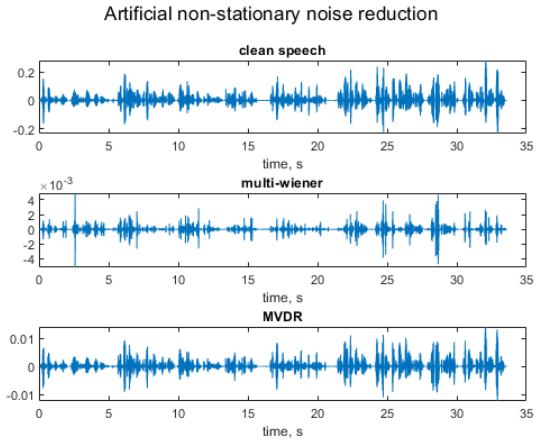

Figure 2: The comparison of artificial non-stationary noise reduction performance

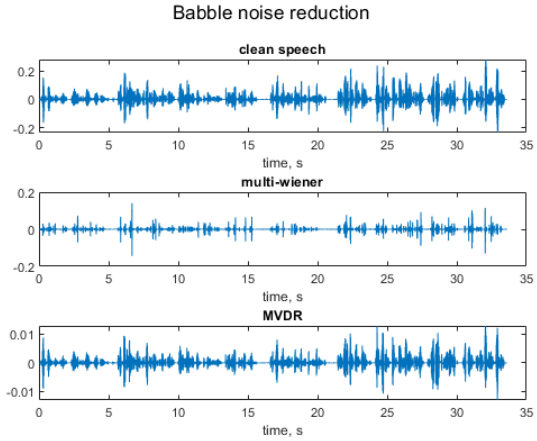

Figure 3: The comparison of babble noise reduction performance

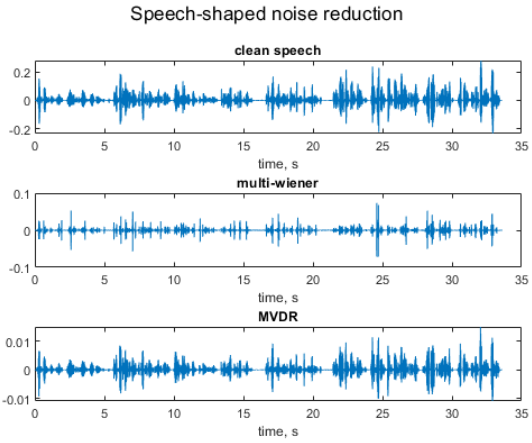

Figure 4: The comparison of speech-shaped noise reduction performance

Noise types	artificial	babble	speech-shaped
MVDR	100.5258	101.3844	101.2689
Multi-Wiener	19.0688	23.3159	29.2890

Table 1: Comparison of noise reduction effects of two types of beamformers by SIIB

There are two beamformers Multi-Wiener and MVDR, and three types of noise. Figure 2 to Figure 4 shows that Multi-Wiener beamformer causes more distortion on signal while MVDR beamformer may contain more noise but less distortion. Table 1 shows the comparison of noise reduction effects by SIIB. The bigger the value is, the more amount of information shared will be. In general, MVDR beamformer performs much better than Multi-Wiener beamformer in all types of noise. The value from MVDR beamformer is 4.37 times in average greater than the value gained from Multi-Wiener beamformer. For MVDR beamformer, it shows similar performance regardless of the noise type, as the value difference is less than 1 among all values. While for Multi-Wiener beamformer, it performs better in speech-shaped noise reduction and shows the least performance in artificial-nonstationary noise reduction. The SIIB value difference between the highest and smallest value of Multi-Wiener beamformer is 10.22. Therefore, reducing noise by MVDR beamformer leads to more understandable speech signals.

A Appendix - Create dataset

```
clear all

amp_arti = 8.8967;
filefolder = 'data\';
filenames = {'clean_speech.wav'; 'clean_speech_2.wav'; 'babble_noise.wav';...
    'aritificial_nonstat_noise.wav';'speech_shaped_noise.wav'};
datanames = {'clean_1'; 'clean_2'; 'babble';...
    'artificial_nonstat'; 'speech_shaped'};
responses = {'h_target';'h_inter1';'h_inter2';'h_inter3';'h_inter4'};
for k = 1:length(filenames)
    filepath = strcat(filefolder,filenames{k});
    importfile(filepath)
    assignin('base', datanames{k}, data);
    clear data filepath
end

load("data\impulse_responses.mat")

h_tar=eval(char(responses(1)));
for M=1:4
    for source=2:5
        h_inter=eval(char(responses(source)));
        s_2(M,:)=conv(h_target(M,:),clean_2);
        s2(M,:)=s_2(M,size(s_2,2)-length(clean_2)+1:size(s_2,2));
        inter_2(M,:,source)=conv(h_inter(M,:),clean_2);
        inter2(M,:,source)=inter_2(M,size(inter_2,2)-length(clean_2)+1:size(inter_2,2),source);
        clear h_inter
    end
    clear source
    inter2=sum(inter2,3);
    noisy_arti_2(M,:)=(artificial_nonstat(1:length(s2(M,:))).*amp_arti)'+inter2(M,:);
    noisy_babble_2(M,:)=(babble(1:length(s2(M,:))))'+inter2(M,:);
    noisy_spee_2(M,:)=(speech_shaped(1:length(s2(M,:))))'+inter2(M,:);
    x_arti_2(M,:)=noisy_arti_2(M,:)+s2(M,:);
    x_babble_2(M,:)=noisy_babble_2(M,:)+s2(M,:);
    x_spee_2(M,:)=noisy_spee_2(M,:)+s2(M,:);
end
clear M

save('data.mat', 'clean_2', 'x_babble_2',...
    'x_arti_2','x_spee_2','fs',...
    'noisy_arti_2','noisy_babble_2','noisy_spee_2')

function importfile(fileToRead1)
%IMPORTFILE(FILETOREAD1)
%  Imports data from the specified file
%  FILETOREAD1:  file to read

newData1 = importdata(fileToRead1);

% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
    assignin('base', vars{i}, newData1.(vars{i}));
end
end
```

B Appendix - Speech enhancement system implement

```
clear all
load("data.mat")

```

```
frame_num=6718;
frame_size=320;

%% STFT
win = hamming(frame_size,'periodic');%16kHz,0.02s

% noise n, speech signals with noise x

for M=1:4
    noisy_1(M,:,:)=stft(noisy_arti_2(M,:),fs,'Window',win);
    noisy_2(M,:,:)=stft(noisy_babble_2(M,:),fs,'Window',win);
    noisy_3(M,:,:)=stft(noisy_spee_2(M,:),fs,'Window',win);
    x_1(M,:,:)=stft(x_arti_2(M,:),fs,'Window',win);
    x_2(M,:,:)=stft(x_babble_2(M,:),fs,'Window',win);
    x_3(M,:,:)=stft(x_spee_2(M,:),fs,'Window',win);
end

%% s estimation
[s_arti_wie,s_arti_mvdr]=beamformer(x_1,noisy_1,frame_num,frame_size);
[s_babble_wie,s_babble_mvdr]=beamformer(x_2,noisy_2,frame_num,frame_size);
[s_spee_wie,s_spee_mvdr]=beamformer(x_3,noisy_3,frame_num,frame_size);

%% ISTFT

esti_arti_1=istft(s_arti_wie,fs,'Window',win);
esti_arti_2=istft(s_arti_mvdr,fs,'Window',win);
esti_babble_1=istft(s_babble_wie,fs,'Window',win);
esti_babble_2=istft(s_babble_mvdr,fs,'Window',win);
esti_spee_1=istft(s_spee_wie,fs,'Window',win);
esti_spee_2=istft(s_spee_mvdr,fs,'Window',win);

save('output.mat', 'esti_arti_1', 'esti_arti_2',...
    'esti_babble_1', 'esti_babble_2','esti_spee_1', 'esti_spee_2')

%% beamformer func
function [s_wie,s_mvdr]=beamformer(x,n,frame_num,frame_size)
for k=1:frame_num
    for l=1:frame_size
% Calculate Rn,Rx
        rx_1=xcorr(x(:,l,k));
        rx_1=rx_1(4:7);
        Rx_1=toeplitz(rx_1);
        clear rx_1

        rn_1=xcorr(n(:,l,k));
        rn_1=rn_1(4:7);
        Rn_1=toeplitz(rn_1);
        clear rn_1

% GEVD Rs
    % A=Rx_1; B=Rn_1;
        [U,Diag,Q]=eig(Rx_1,Rn_1);
        Diag=Diag-eye(size(Diag,1));
        Rs_1=Rn_1*U*Diag/U;

% Calculate Wiener
        e1=zeros(size(Rs_1,2),1);
        e1(1)=1;
        w1_wie=Rx_1\Rs_1*e1;
% Calculate MVDR
        [¬,index_row]=max(abs(diag(Diag)));
        a=Q(:,index_row(1));
        w1_mvdr=Rn_1\a/(a'/Rn_1*a);

% solve s
        s_1_wie=w1_wie'*x(:,l,k);
        s_1_mvdr=w1_mvdr'*x(:,l,k);
        s_wie(l,k)=s_1_wie;
        s_mvdr(l,k)=s_1_mvdr;
    end
```

```
end
end
```

C Appendix - Evaluation

```
load output.mat
load data.mat

%% visualization
len=length(esti_spee_1);
clean=clean_2(1:len);
figure(1)
subplot(3,1,1); plot((1:length(clean))/fs,clean); title('clean speech')
xlabel('time, s')
subplot(3,1,2); plot((1:len)/fs,10*esti_arti_1); title('multi-wiener')
xlabel('time, s')
subplot(3,1,3); plot((1:len)/fs,10*esti_arti_2); title('MVDR')
xlabel('time, s')

figure(2)
subplot(3,1,1); plot((1:length(clean))/fs,clean); title('clean speech')
xlabel('time, s')
subplot(3,1,2); plot((1:len)/fs,10*esti_babble_1); title('multi-wiener')
xlabel('time, s')
subplot(3,1,3); plot((1:len)/fs,10*esti_babble_2); title('MVDR')
xlabel('time, s')

figure(3)
subplot(3,1,1); plot((1:length(clean))/fs,clean); title('clean speech')
xlabel('time, s')
subplot(3,1,2); plot((1:len)/fs,esti_spee_1); title('multi-wiener')
xlabel('time, s')
subplot(3,1,3); plot((1:len)/fs,esti_spee_2); title('MVDR')
xlabel('time, s')

%% SIIB_Gauss
siib_a_w = SIIB_Gauss(clean,esti_arti_1,fs);
siib_a_m = SIIB_Gauss(clean,esti_arti_2,fs);
siib_b_w = SIIB_Gauss(clean,esti_babble_1,fs);
siib_b_m = SIIB_Gauss(clean,esti_babble_2,fs);
siib_s_w = SIIB_Gauss(clean,esti_spee_1,fs);
siib_s_m = SIIB_Gauss(clean,esti_spee_2,fs);
```

References

[1] R. C. Hendriks, *Microphone Array Processing.* TU Delft, June 2022.

[2] S. Van Kuyk, W. B. Kleijn, and R. C. Hendriks, "An instrumental intelligibility metric based on information theory," *IEEE Signal Processing Letters*, vol. 25, no. 1, pp. 115–119, 2017.