Stochastic Signal Processing

Lesson 7+8: Experimental Report 2

Frequency Estimation of Sin Signal

&

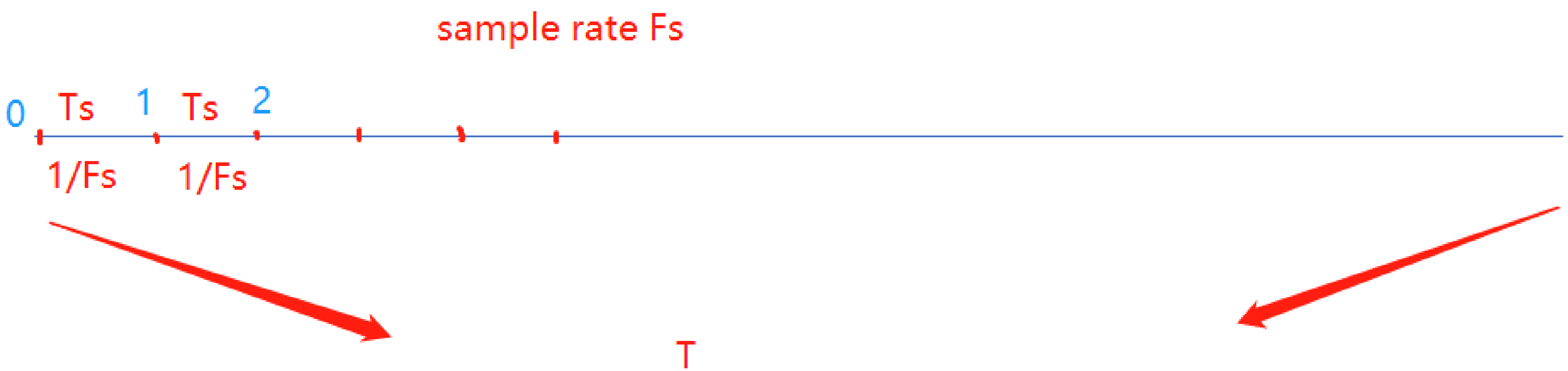
Time Domain Beamforming and Localization

+

DOA estimation

- Introduction

For continuous-time signals, it can be simulated by discrete sequences with sufficiently small sampling intervals. For example, to generate a continuous signal with a start time zero and a duration of T seconds, the sampling interval can be set to T_S (or sample rate $F_S = 1/T_S$), then sequences length is $N = \frac{T}{T_S} + 1$.



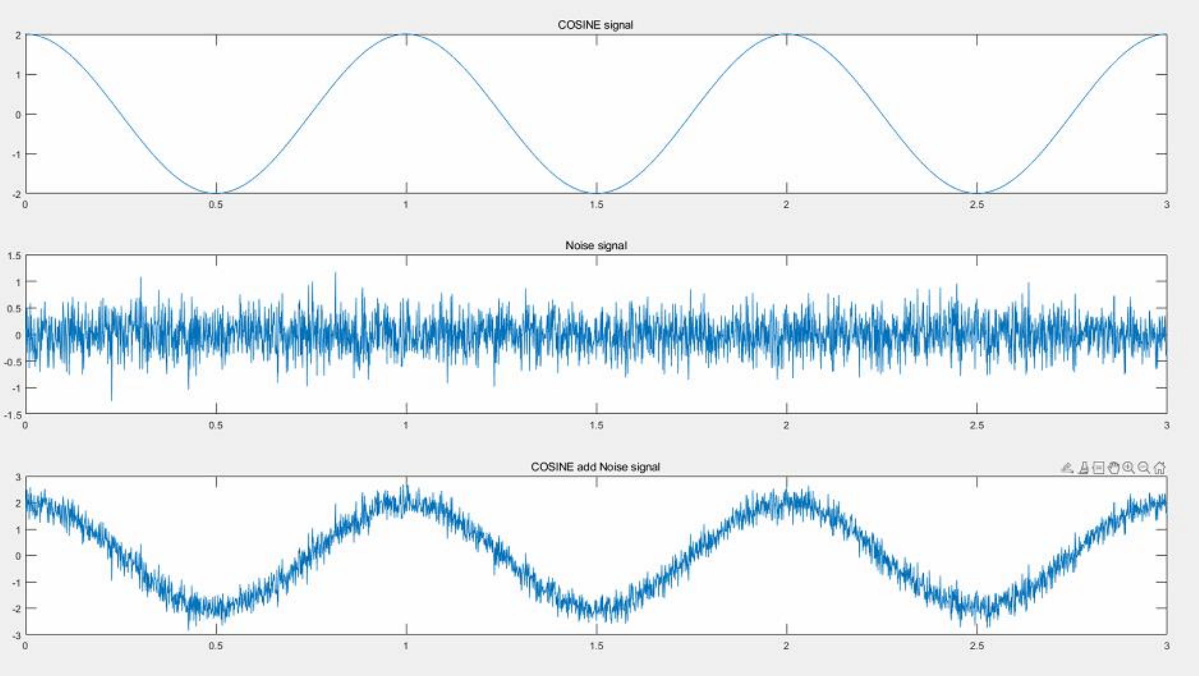
- Practice 1

The simulation generates a sinusoidal signal with frequency 1Hz and Gaussian noise. The Signal amplitude is 2 and noise variance $\sigma^2 = 0.1$. Generate a signal sampled at F_S= 1 kHz for T = 3 seconds. Plot a sinusoidal signal with Gaussian noise.

In order to ensure that the signal is not distorted, the sampling rate should be greater than twice the highest frequency of the signal.

- Practice 1

```
T = 3;              % signal time_len(s)
Fs = 1000;          % sample rate(Hz)
t = 0:1/Fs:T-1/Fs; % sample point
s = 2*cos(2*pi*t); % signal cos
subplot(311);
plot(t,s);
title('\fontname{}COSINE signal');
n = sqrt(0.1)*randn(size(t)) %noise
subplot(312);
plot(t,n);
title('\fontname{}Noise signal');
x=s+n;              % signal cos add noise
subplot(313);
plot(t,x);
title('\fontname{}COSINE add Noise signal');
```



- Review Lesson 6

Cross-Correlation, Cross-Covariance and Jointly WSS

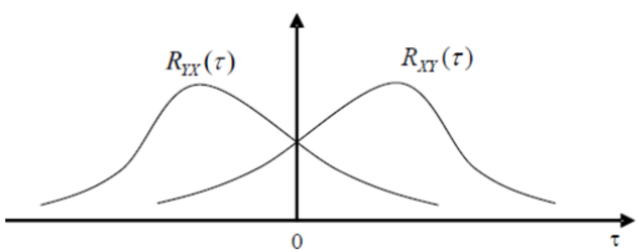
Properties of jointly WSS:

- Property 1: for real valued process

$$R_{XY}(-\tau) = R_{YX}(\tau)$$
$$C_{XY}(-\tau) = C_{YX}(\tau)$$

$$\begin{aligned} R_{XY}(\tau) &= E[X(t)Y(t-\tau)] \\ R_{XY}(-\tau) &= E[X(t)Y(t+\tau)] \end{aligned} \rightarrow R_{XY}(\tau) \overset{\neq R_{YX}(\tau)}{\neq} R_{XY}(-\tau) \quad R_{XY}(\tau) \neq -R_{XY}(-\tau)$$

Topical curve of Cross-Correlation of Jointly WSS processes



25

- Cross-Correlation

Suppose $\{x_n\}_{n=0}^{N-1}$ and $\{y_n\}_{n=0}^{N-1}$ are samples of real random signals X_n and Y_n, respectively. For real valued process. The formula of the cross-correlation function:

$$\hat{R}_{XY}[m] = \begin{cases} \alpha \sum_{n=0}^{N-m-1} x_{n+m} y_n, & 0 \leq m \leq N-1 \\ \hat{R}_{YX}[-m], & -(N-1) \leq m \leq 0 \end{cases}$$

The formula of the autocorrelation function:

$$\hat{R}_X[m] = \alpha \sum_{n=0}^{N-|m|-1} x_{n+|m|} x_n, \quad 0 \leq |m| \leq N-1$$

- xcorr

Cross-correlation   r = xcorr(___,scaleopt)

scaleopt	归一化系数α
none	1
biased	1/N
unbiased	1/(N-|m|)
coeff	$1/\sqrt{\hat{R}_X(0)\hat{R}_Y(0)}$

`scaleopt` — Normalization option

`'none'` (default) | `'biased'` | `'unbiased'` | `'normalized'` | `'coeff'`

Normalization option, specified as one of the following.

- `'none'` — Raw, unscaled cross-correlation. `'none'` is the only valid option when x and y have different lengths.
- `'biased'` — Biased estimate of the cross-correlation:

$$\hat{R}_{xy,\text{biased}}(m) = \frac{1}{N}\hat{R}_{xy}(m).$$

- `'unbiased'` — Unbiased estimate of the cross-correlation:

$$\hat{R}_{xy,\text{unbiased}}(m) = \frac{1}{N-|m|}\hat{R}_{xy}(m).$$

- `'normalized'` or `'coeff'` — Normalizes the sequence so that the autocorrelations at zero lag equal 1:

$$\hat{R}_{xy,\text{coeff}}(m) = \frac{1}{\sqrt{\hat{R}_{xx}(0)\hat{R}_{yy}(0)}}\hat{R}_{xy}(m).$$

- Practice 2

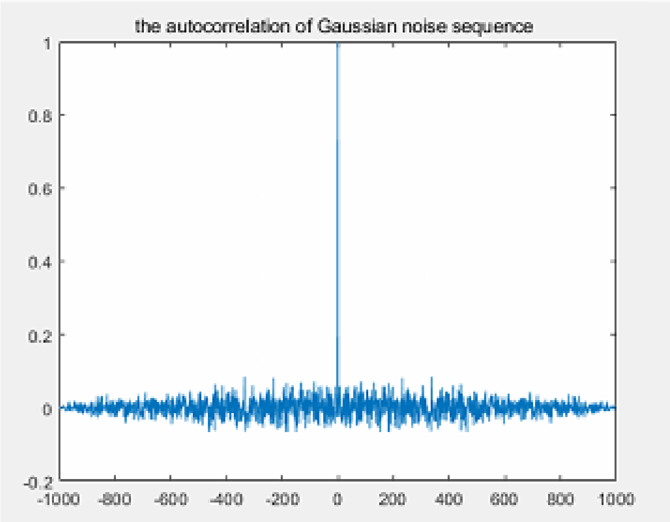
Simulate Gaussian noise sequence with mean 0 and variance 5, and estimate its mean, variance and autocorrelation

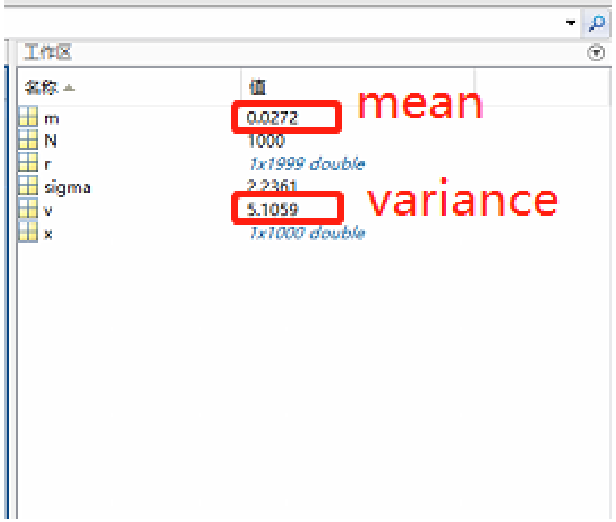
```
N = 1000;                 % sequence_len
sigma = sqrt(5);                         % std
x = randn(1,N)*sigma;          % Simulate Gaussian sequence with
m = mean(x);
v = var(x);
r = xcorr(x,'coeff');     % Normalizes the sequence so that the
                                                        autocorrelations at zero lag equal 1
plot(1-N:N-1,r)
title('\fontname{}the  autocorrelation of Gaussian noise sequence ')
```

- Practice 2

For real valued Stationary Stochastic Processes:
$R_X(0) \geq |R_X(\tau)|$

From the figure, we can see that 0-point is maximum and autocorrelation is an even function. Except for 0, other real values are basically stable around zero. It is called White Gaussian Noise





- Experimental Report 2
- Basic 1 (10ponits)：A stochastic signal

$$X(t) = sin(2\pi f_1 t) + 2cos(2\pi f_2 t) + N(t)$$

where f_1=60Hz, f_2=150Hz, $N(t)$ is white Gaussian Noise with mean 0 and variance and $\sigma^2 = 0.1$. Generate a signal sampled at 1 kHz for 4 seconds. Plot the signal, and the autocorrelation $R_X(\tau)$, and Cross-Correlation $R_{XN}(\tau)$.

Hint : see practice 1, 2.

- **Requirement**：

1. Please provide your code, they must be runnable (no error, warning accepted), otherwise, 0 point. All figures should be come with titles.
2. Your code should be run on Matlab, and give the figures on your report.

- Basic knowledge of Basic 2

Characteristics of Autocorrelation of Stationary Stochastic Processes

Periodic Signal / Processes: $X(t) = X(t+T)$

- If $X(t) = X(t+T)$, and it is Stationary, then:

$$R_X(\tau) = E[X(t+\tau)X(t)] = E[X(t+T+\tau)X(t)] = R_X(\tau+T)$$

If the stochastic process has a periodic component, the autocorrelation function also has a periodic component.

- Experimental Report 2

- Basic 2 (40 points): construct a sinusoidal signal with Gaussian noise, such as $X(t) = cos(2\pi f t + \theta) + N(t)$, where $N(t)$ is white Gaussian noise with mean 0 and variance σ^2= 0.1, and you can choose other parameters (for example, frequency f, phase θ, sampling rate, sampling time, etc.).
  - Use characteristics of autocorrelation to estimate your signal frequency.

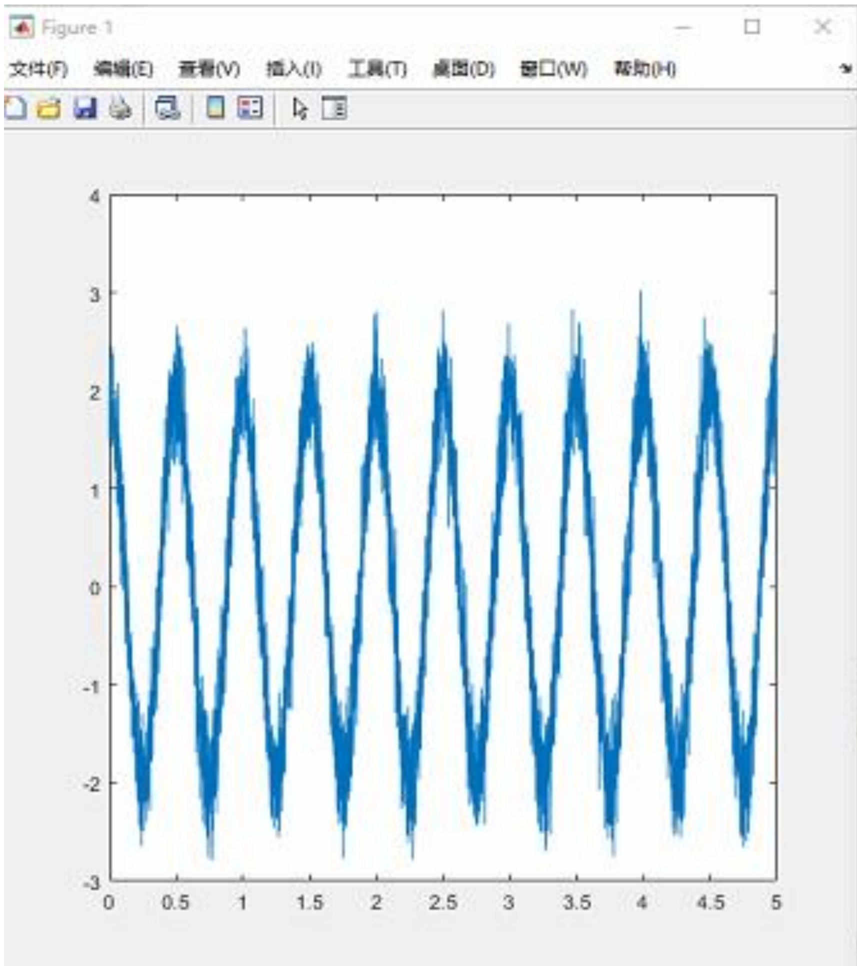
  (Hint: $R_X(\tau) = R_X(\tau + T)$)
  - For σ^2=0.5, 1, 5 (you can try more), plot the figure of Autocorrelation for one run.
  - Test the accuracy of **your estimation of signal frequency** using autocorrelation for **100 independent runs**, show the results using table(s) or figure(s), and give analysis.
- Requirement

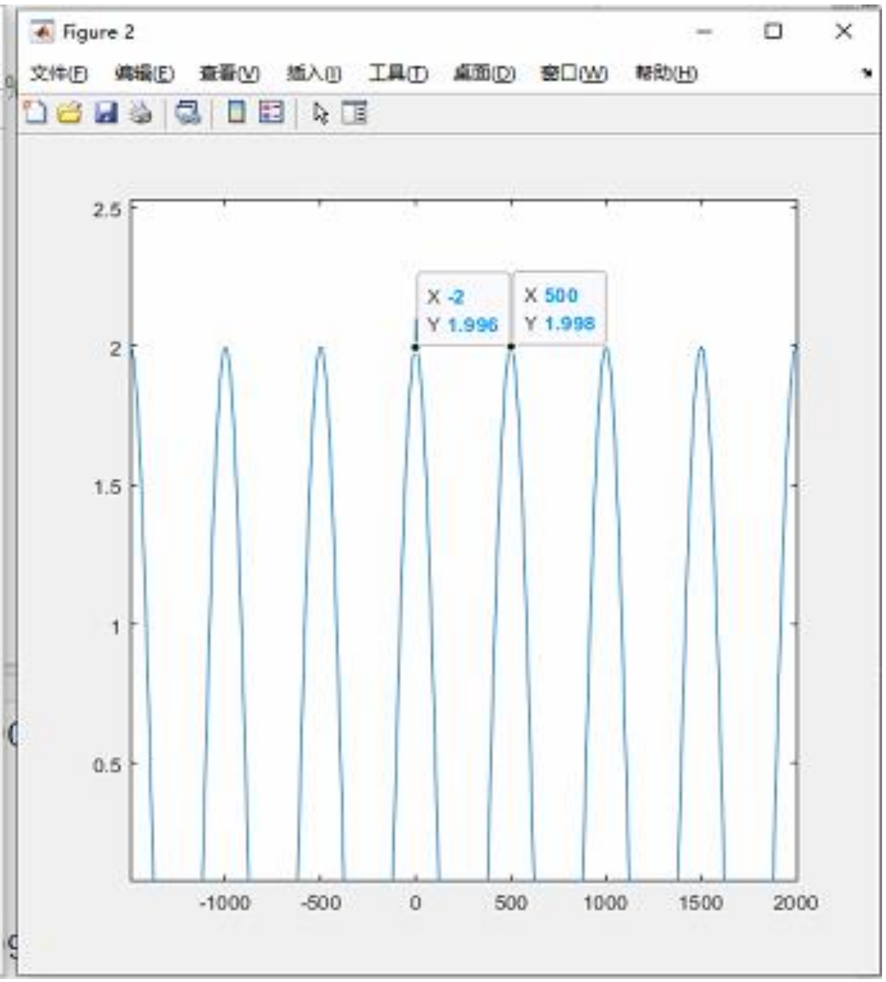
1. Your code must be runnable (no error, warning accepted), otherwise, 0 point. All figures should be come with titles.
2. Explain why it is possible to estimate the **signal frequency** through **autocorrelation**.
3. Output the figures, and give analysis.

- Experimental Report 2

- Basic 2 (40 points): construct a sinusoidal signal with Gaussian noise, such as $X(t) = cos(2\pi f t + \theta) + N(t)$, where $N(t)$ is white Gaussian noise with mean 0 and variance σ^2= 0.1, and you can choose other parameters (for example, frequency f, phase θ, sampling rate, sampling time, etc.).
    1. For example, you define the signal frequency f=2Hz, and plot the figure of Autocorrelation. In the figure, count **the number of points *N* between two adjacent top point**. *N+1* is the period for $R_X(\tau)$ and $X(t)$. At last, estimate the frequency $\hat{f}$ (we usually use a $\widehat{\ }$ to represent the parameter estimated by a method) and test whether it is close to the real frequency f.

- *for example,* $S(t) = cos(2\pi f t) + N(t)$*,* $f = 2Hz$*, sampling rate is 1k Hz, which is, one sample takes 0.001s*
- From the figure2, N ≈ 500, T =500*0.001=0.5s is the period

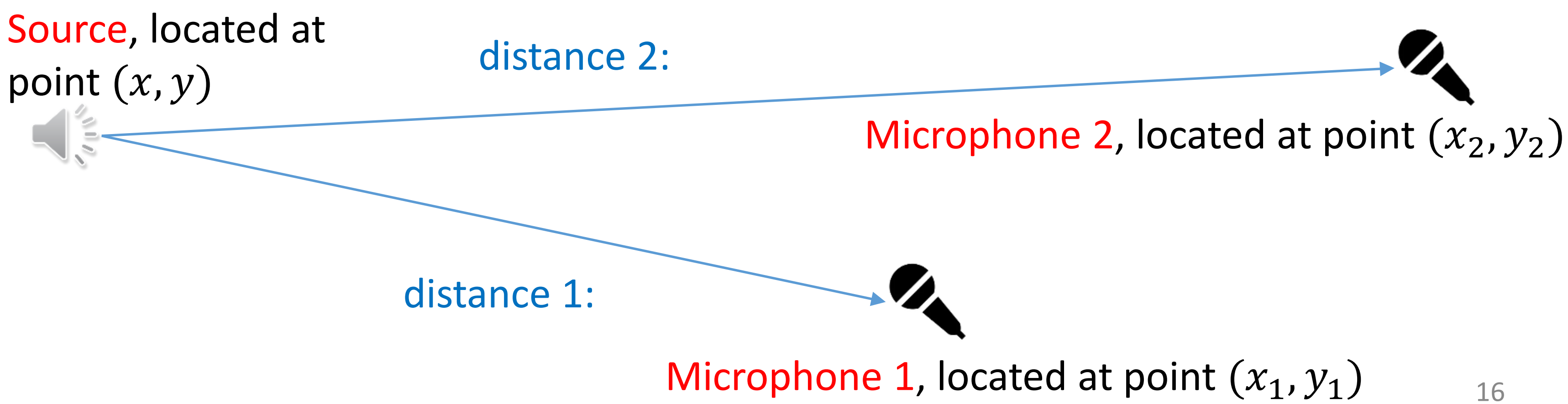




- Experimental Report 2
- Basic 2 (40 points): construct a sinusoidal signal with Gaussian noise, such as $X(t) = cos(2\pi f t + \theta) + N(t)$, where $N(t)$ is white Gaussian noise with mean 0 and variance σ^2= 0.1, and you can choose other parameters (for example, frequency f, phase θ, sampling rate, sampling time, etc.).
    1. For example, you define the signal frequency f=2Hz, and plot the figure of Autocorrelation. In the figure, count **the number of points *N* between two adjacent top point**. *N+1* is the period for $R_X(\tau)$ and $X(t)$. At last, estimate the frequency $\hat{f}$ (we usually use a ^ to represent the parameter estimated by a method) and test whether it is close to the real frequency f.
    2. However, simply **the number of points *N* between two adjacent top point** to estimate the frequency might be very sensitive to noise. **Is there other method?**

The Microphone Array

- At time 0, power on the microphones 1 and 2, start the source, then the source will send out speech signal
- The distances between the source and microphones 1 and 2 are d_1 and d_2, here we assume $d_1 < d_2$
- Note that the speed at which sound travels through the air is $c = 340m/s$, therefore, at time $t_1 = d_1 /c$ seconds, the speech reach microphone 1 - microphone 1 can receive meaningful signal, but microphone 1 still receive nothing, or noise only.
- At time $t_2 = d_2 /c$ seconds, the speech reach microphone 2
- Note that what microphone 2 received at time $t_2 = d_2 /c$ is what microphone 1 received at time $t_1 = d_1 /c$, the time delay is $t_1 - t_2$



The Microphone Array

- At time 0, power on the microphones 1 and 2, start the source, then the source will send out speech signal
- The distances between the source and microphones 1 and 2 are d_1 and d_2, here we assume $d_1 < d_2$
- Note that the speed at which sound travels through the air is $c = 340m/s$, therefore, at time $t_1 = d_1 /c$ seconds, the speech reach microphone 1 - microphone 1 can receive meaningful signal, but microphone 1 still receive nothing, or noise only.
- At time $t_2 = \ d_2 /c$ seconds, the speech reach microphone 2
- Note that what microphone 2 received at time $t_2 = d_2 /c$ is what microphone 1 received at time $t_1 = \ d_1 /c$, the time delay is $t_1 - t_2$
- For example, if $d_1 = 17$ and $d_2 = 34$, then $t_1 = 0.05$ and $t_2 = 0.1$, thus the time delay is $t_1 - t_2 = -0.05s$, '$-$' means that microphone 1 leads (领先) microphone 2 by 0.05 seconds.
- For example, if $d_1 = 34$ and $d_2 = 17$, then $t_1 = 0.1$ and $t_2 = 0.05$, thus the time delay is $t_1 - t_2 = +0.05s$, '$+$' means that microphone 1 behinds (落后) microphone 2 by 0.05 seconds.
- This is the situation of continuous case, how about discrete case?

The Microphone Array

- At time $t_1 = d_1/c$ & $t_2 = d_2/c$ seconds, the speech reach microphone 1 & 2, the time delay is $t_1 - t_2$
- For example, if $d_1 = 17$ and $d_2 = 34$, then $t_1 = 0.05$ and $t_2 = 0.1$, thus the time delay is $t_1 - t_2 = -0.05s$, '$-$' means that microphone 1 leads (领先) microphone 2 by 0.05 seconds.
- If the microphone take 100 samples in 1 second, 0.05 second means 5 samples.
- For example, the source send out signal [1,2,3,4,5,6,7,8,9,10], 10 samples (the length is 0.1 second)
- Start from time 0, microphone 1 & 2 receive (from time 0 to 0.2 second):
  - Signal 1 from microphone 1: [0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
  - Signal 2 from microphone 2: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10]
- For noise-free case (without noise), any microphone can give a clear and clean speech result.
- However, there are noise!
- It is better to add what received in both microphones together to get a better speech, instead of using only one!
- However, if add them up simply, will get [0,0,0,0,0,1,2,3,4,5,7,9,11,13,15,6,7,8,9,10], there is overlap! Even worse!
- How to solve this problem?

The Microphone Array

- Cross-Correlation!
  - Signal 1 from microphone 1: [0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
  - Signal 2 from microphone 2: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10]
- Calculate the Cross-Correlation of the two signal, and find the maximum at best_lag=-5 (最佳时延)
- It means that, we can delay Signal 1 by 5 samples, (therefore, add 5 zeros in the end of Signal 2), and get:
  - Signal 1 after delay: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
  - Signal 2 with zero padding: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
- Now add them up and get: [0,0,0,0,0,0,0,0,0,0,2,4,6,8,10,12,14,16,18,20,0,0,0,0,0]. Great! We had enlarged the signal!
- This is called time domain beamforming (时域波束形成).
- Note that best_lag=-5, with sampling rate 100 per second, the time delay is -5/100=-0.05 second: microphone 1 leads (领先) microphone 2 by 0.05 seconds.
- Now lets go to the code

The Microphone Array

- Cross-Correlation!
    - Signal 1 from microphone 1: [0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
    - Signal 2 from microphone 2: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10]
- Calculate the Cross-Correlation of the two signal, and find the maximum at best_lag=-5 (最佳时延)
- It means that, we can delay Signal 1 by 5 samples, (therefore, add 5 zeros in the end of Signal 2), and get:
    - Signal 1 after delay: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
    - Signal 2 with zero padding: [0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,0,0,0,0,0]
- Now add them up and get: [0,0,0,0,0,0,0,0,0,0,2,4,6,8,10,12,14,16,18,20,0,0,0,0,0]. Great! We had enlarged the signal!
- This is called time domain beamforming (时域波束形成).

Note that the way the time delay added here is not correct in real world engineering, the correct one is as below, which is, when the speech reach the microphone, the longer the distance, the small the values - Sound is attenuated by a factor. However, here is only one simplified example, therefore we ignore this attenuation in the experiment.

The Microphone Array

- Go through the code
  - Is_add_special_noise=1: add exactly the opposite noise to adjacent two microphones with correct delay. This is for understanding of the code and the validity of autocorrelation.
  - Is_add_special_noise=0: the more general case. Add random Gaussian noise to microphones.
- Test different SNR: 1000, 30, 0, etc. (Is_add_special_noise=0/1, 2 microphones)
- Test ‘Is_add_special_noise’: 0/1 (SNR=30 & 0, 2 microphones)

The Microphone Array

- Is_add_special_noise = 1 (SNR=30 & 0, 2 microphones)

```
% just to see the real lag, cannot use Real_lag = L_TD(1)-L_TD(2) in your
% code when add the signal from different microphones
Real_lag = L_TD(1)-L_TD(2);
error = Lag_12_estimate - Real_lag
```

  - SNR=0, the real lag is -2278 but the estimated lag is -2277, although it is close, it fails to fully remove the noise.

```
% SNR_dB = 1000; % thi
SNR_dB = 0; % this ind
% SNR_dB = 30; % this
```

```
error =

     1
```

  - SNR=30, the real lag is -2278 and the estimated lag is -2278, it can fully remove the noise.

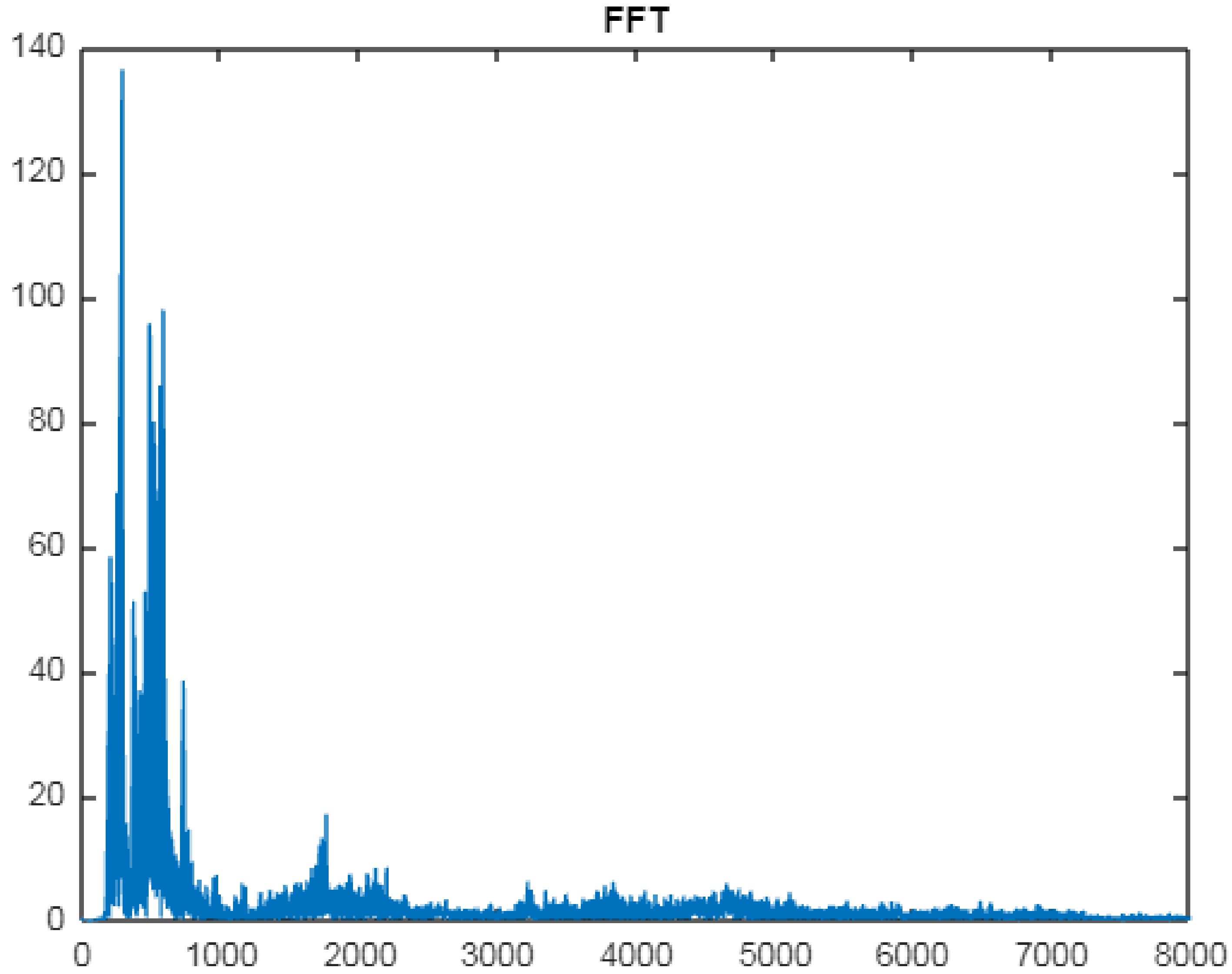
```
% SNR_dB = 1000; % t
% SNR_dB = 0; % this
SNR_dB = 30; % this
```

```
error =

     0
```

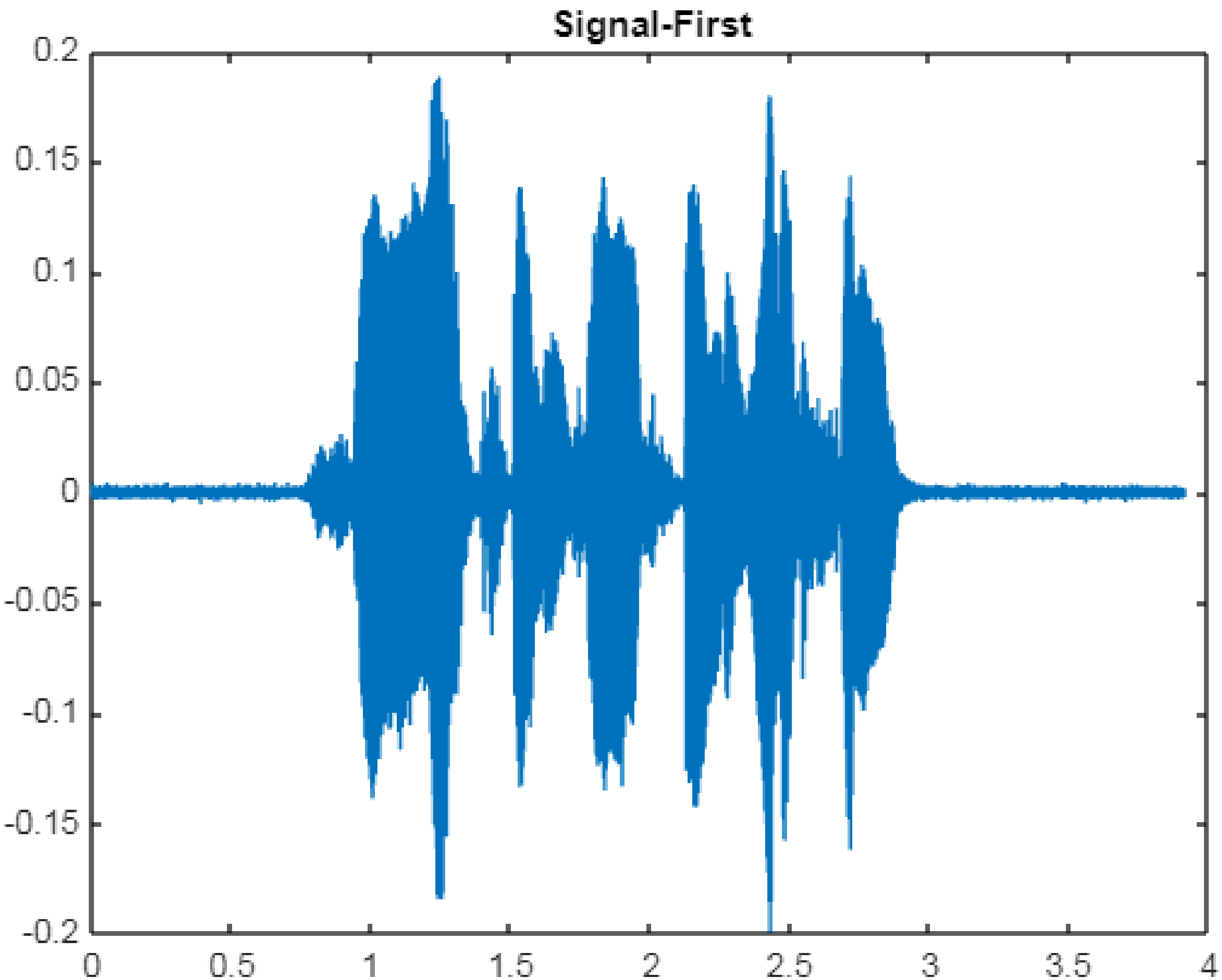
The Microphone Array

- The 'FFT' figure of the speech



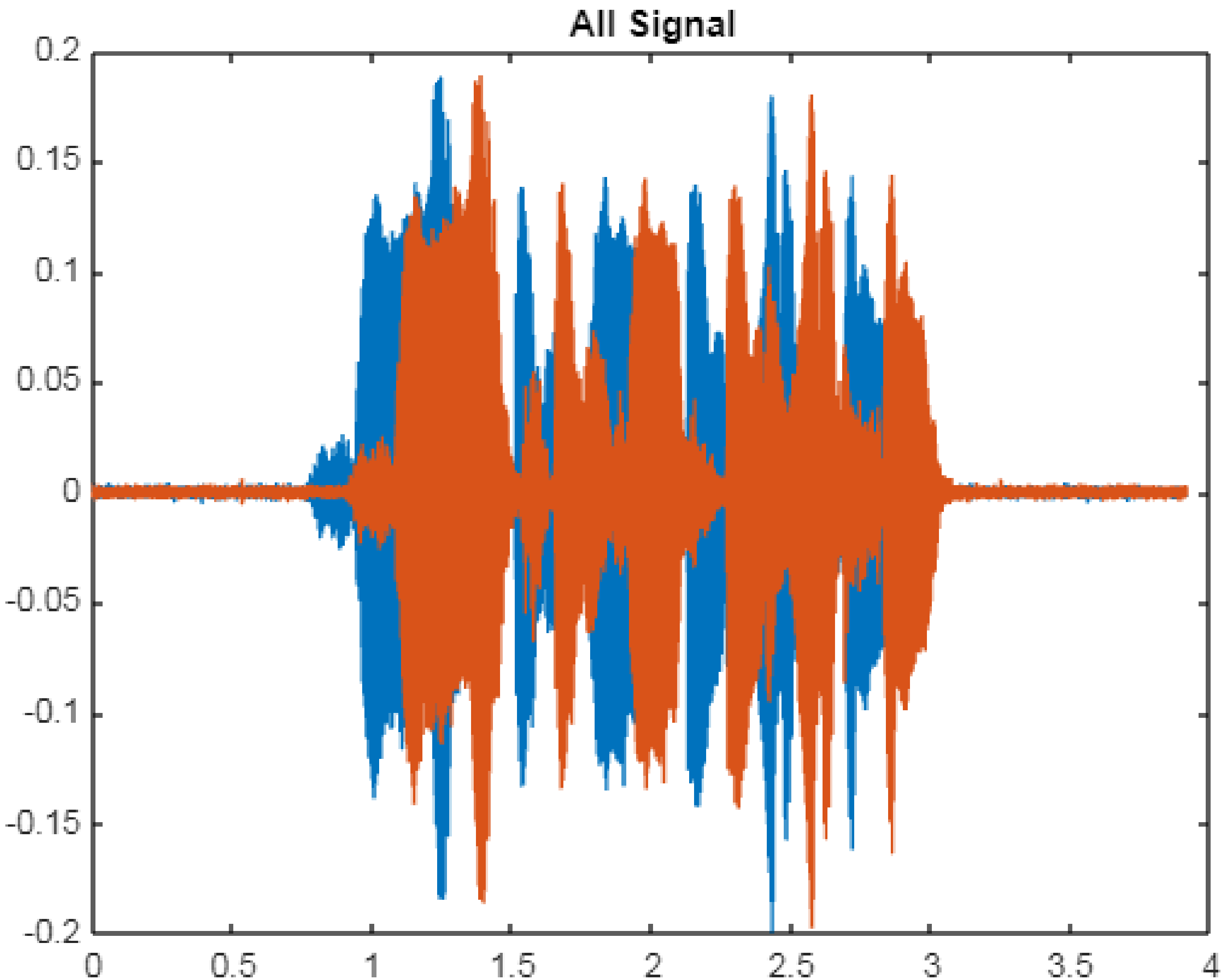
The Microphone Array

- The signal from the first microphone



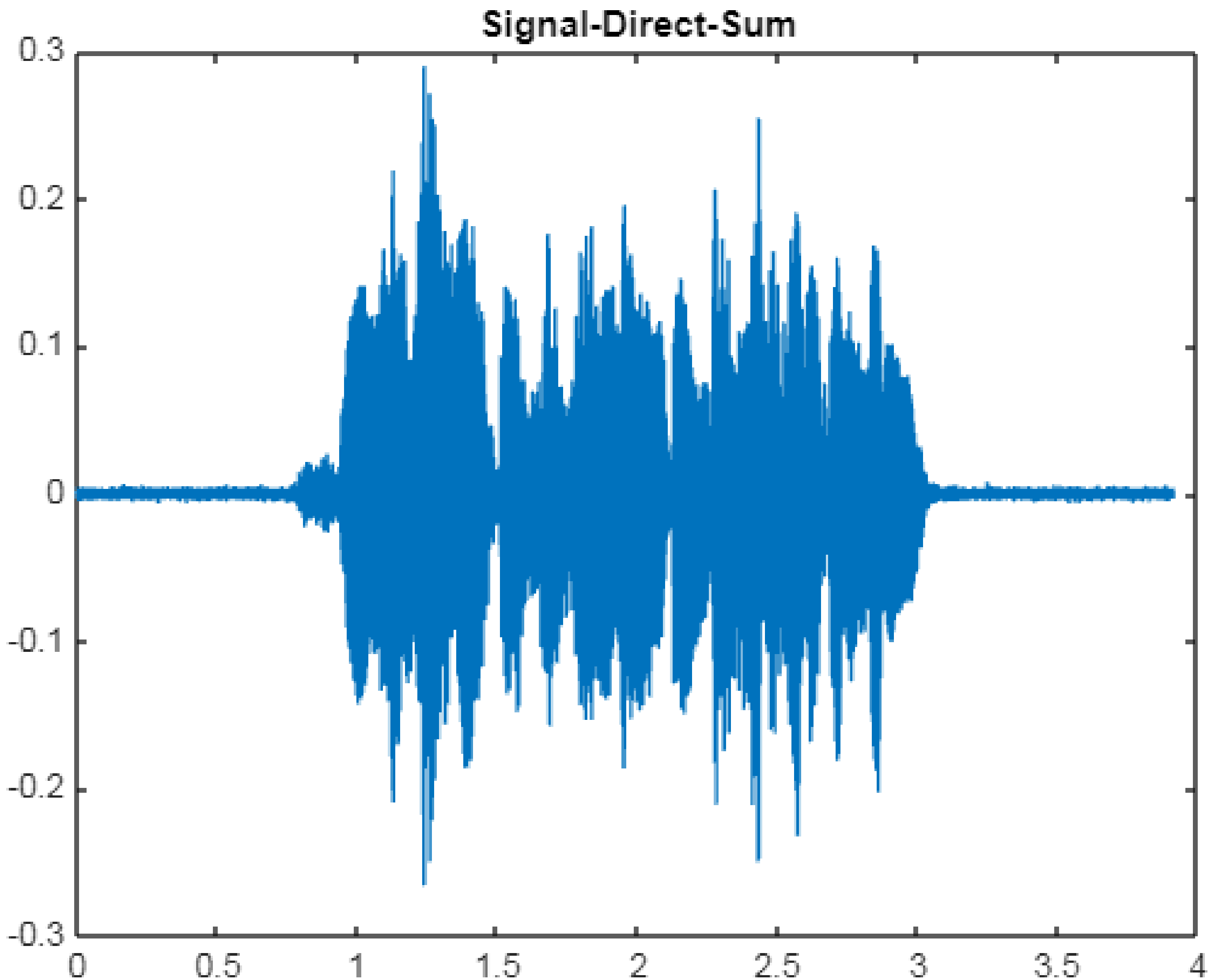
The Microphone Array

- The ‘All Signal’ figure with different color for different microphones



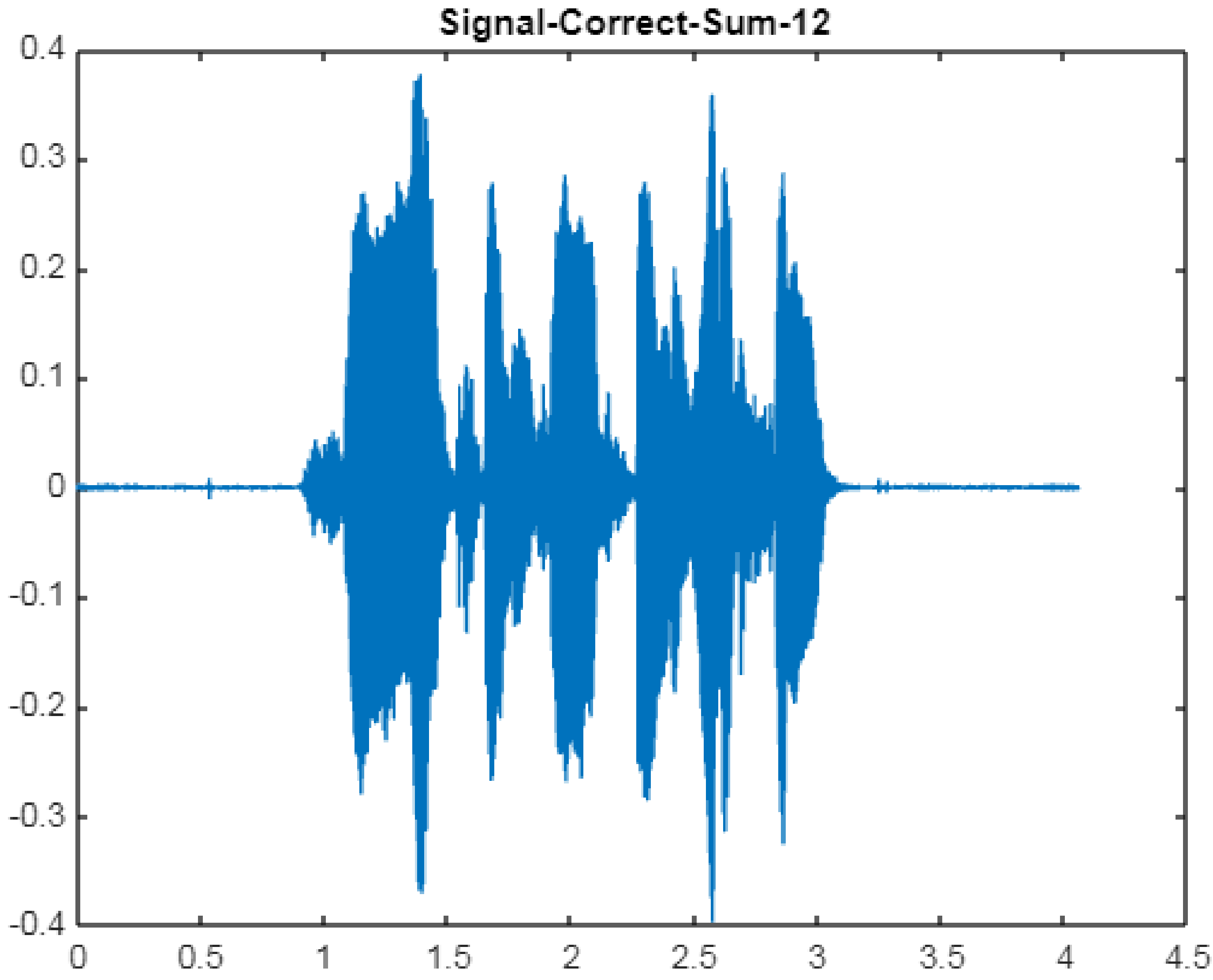
The Microphone Array

- The ‘Direct-Sum’ figure from two microphones: it is not good!



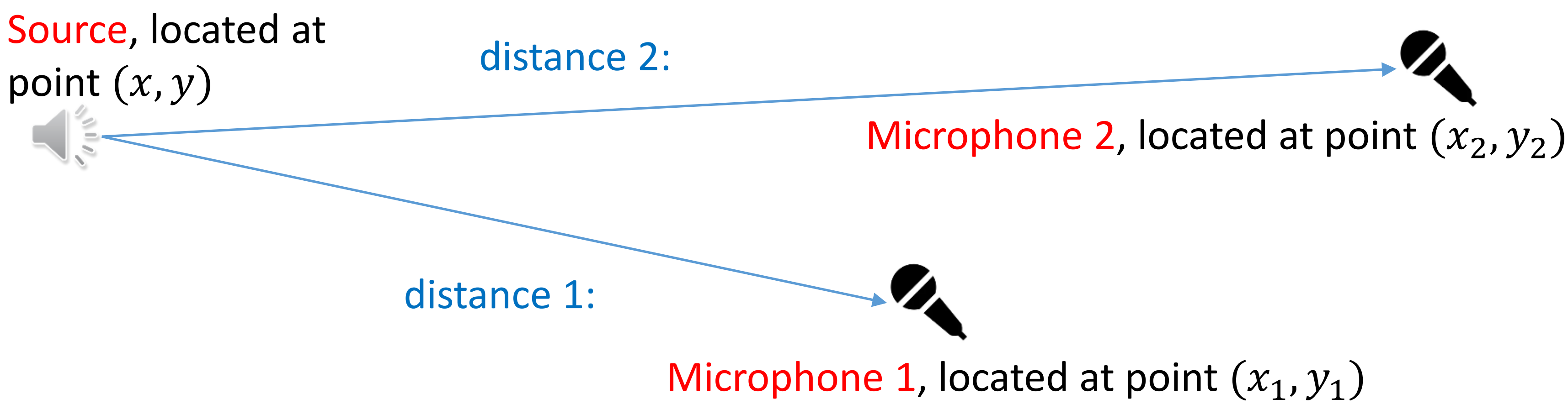
The Microphone Array

- The ‘Correct-Sum-12’ figure from two microphones: it is good!



Reading (*)

- Once we get the time delay (lag) between microphones 1 and 2, the distance difference between microphone 1/2 and the source (which is, $d_1 - d_2$) can be calculated.
- Then the source must be in a hyperbolic curve (双曲线)
- If there two more microphone, namely, microphones 3 and 4. the microphone pairs 1&3 and 1&4 can give two more hyperbolic curves
- **The three curves can then finally locate the source correctly.**
- If there enough time, will go back to this (I don't think so this year)
- Please refers to H. C. So. @ http://www.ee.cityu.edu.hk/~hcso/



Advance part 1 of Experimental Report 2 (20 points)

Advance part 1: Advance part 1: follow the settings of 4 microphones with (SNR=30/10/-10, Is_add_special_noise=0) {microphone locations: (0,0)m, (20,0)m, (0,10)m and (20,10)m; source located at (1,1)m}:

1. correctly add the signals from the 4 microphones with correctly estimated lags lags (this is your method), explain your method with necessary texts, equations, and/or flowchart.

Hint:

a. you can calculate the cross-correlation between microphones 1&2, 1&3 and 1&4, and move the signal received from microphones 2,3,4 by delaying some samples or adding some zeros; and then you can sum them up to get a good clean speech.

b. Therefore, you should change the ‘main.m’ file to achieve this task

Advance part 1 of Experimental Report 2 (20 points)

Advance part 1: follow the settings of 4 microphones with (SNR=30/10/-10, Is_add_special_noise=0) {microphone locations: (0,0)m, (20,0)m, (0,10)m and (20,10)m; source located at (1,1)m}:

1. correctly add the signals from the 4 microphones with correctly estimated lags lags (this is your method), explain your method with necessary texts, equations, and/or flowchart.

2. Show the figures under 3 SNR cases (SNR = 30,10,-10dB)

2.1. the four signals received from 4 microphones, with four colors, in one figure. (Please refer to page 25 in 'Lesson 7 experiment.pdf')

2.2. the signal the after 'correctly add the signals from the 4 microphones with correct lags '. (Please refer to page 27 in 'Lesson 7 experiment.pdf')

3. Give necessary analysis

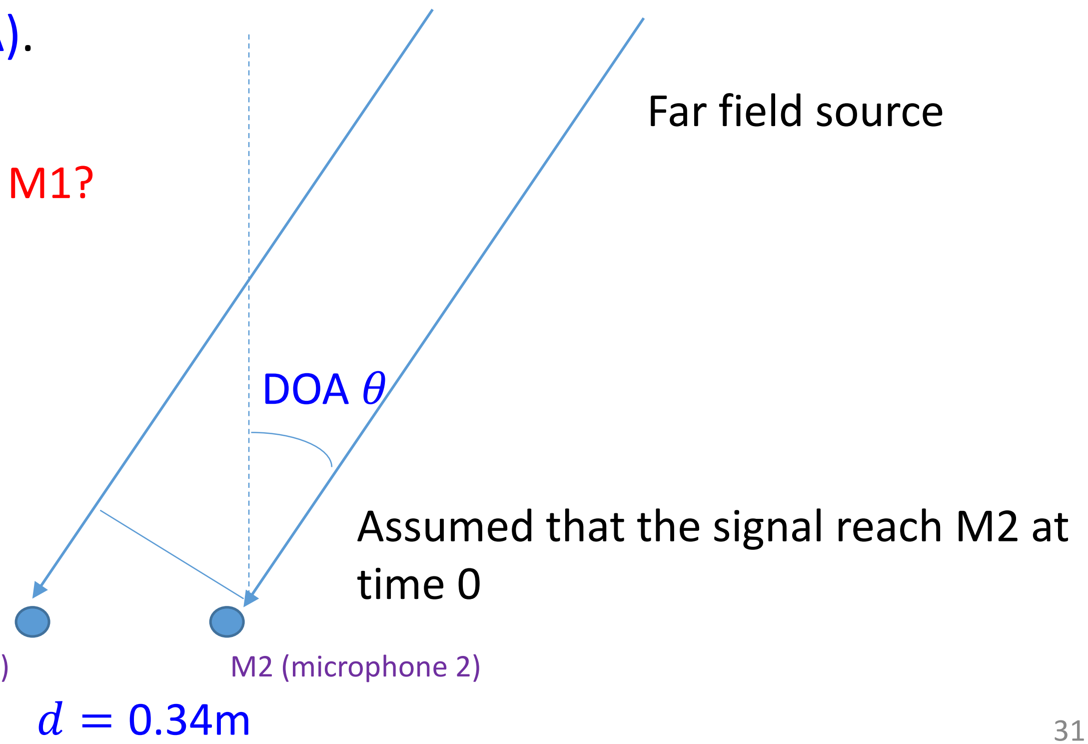
Advance part 2 of Experimental Report 2 (30 points)

Advance part 2:

Now you are given 2 microphones, with id = 0:1:1. These microphones are located at (0, 0+0.34*id)m. The source is now located at least 60 meters away.

In engineering (in some cases), if the source is located >= 10*D away from the microphone array, where D is the size of the microphone array. It is called 'far field' source. In this condition, the target is to estimate the Direction of Arrival (DOA).

When will the signal reach M1?



Basic knowledge

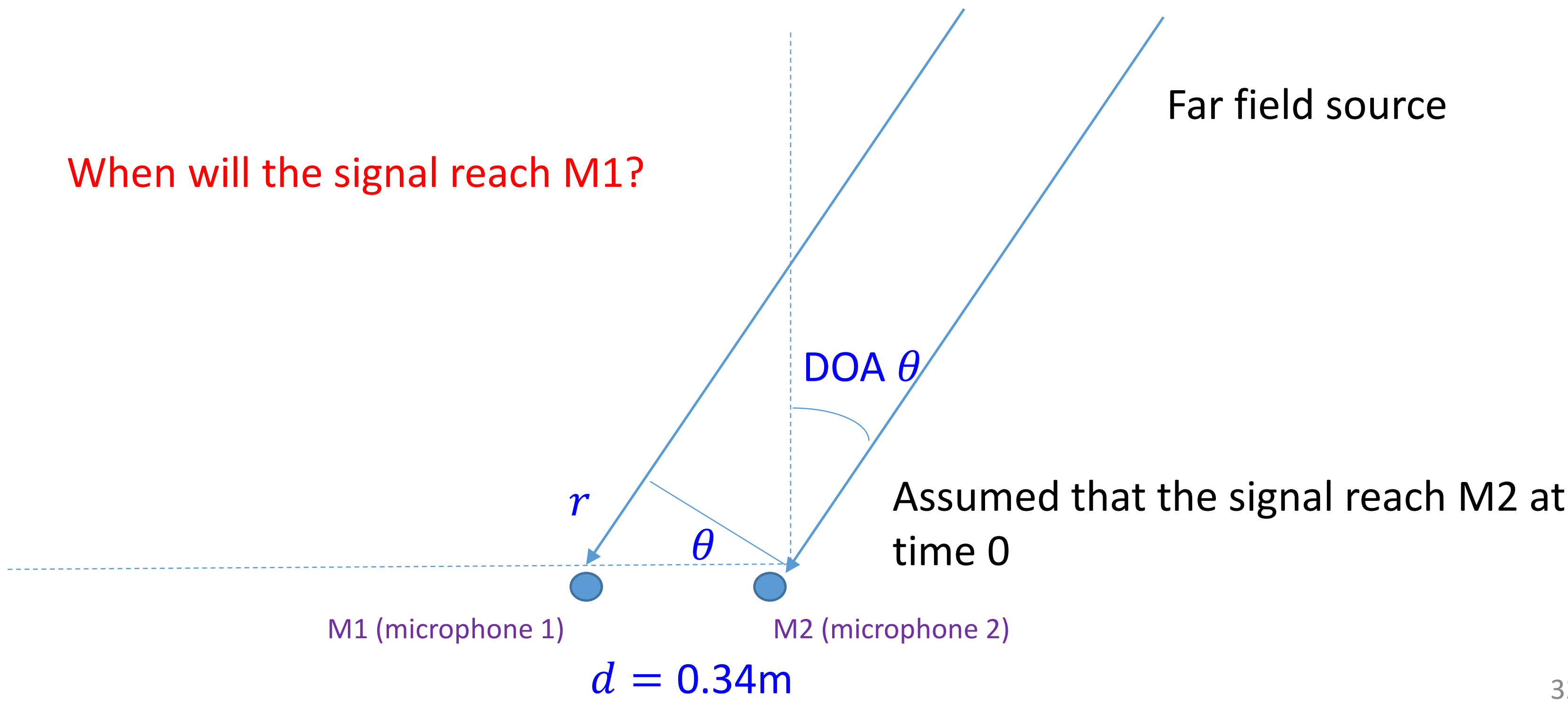
$r = d\ \sin(\theta)$ $c = 340m/s$ is the speed of sound

At time $t = r/c$, the signal reach M1!

For example, if $\theta = \pi/4$, $r = 0.17$m, then at $t = 0.0005$s, the signal reach M1!

If the sampling rate of the signal is 16000, then 16000*0.0005=8, M1 lags M2 8 samples, or says, M2 leads M1 8 samples (M2 lags M1 -8 samples).

When will the signal reach M1?



Basic knowledge

$r = d\sin(\theta)$ $c = 340m/s$ is the speed of sound

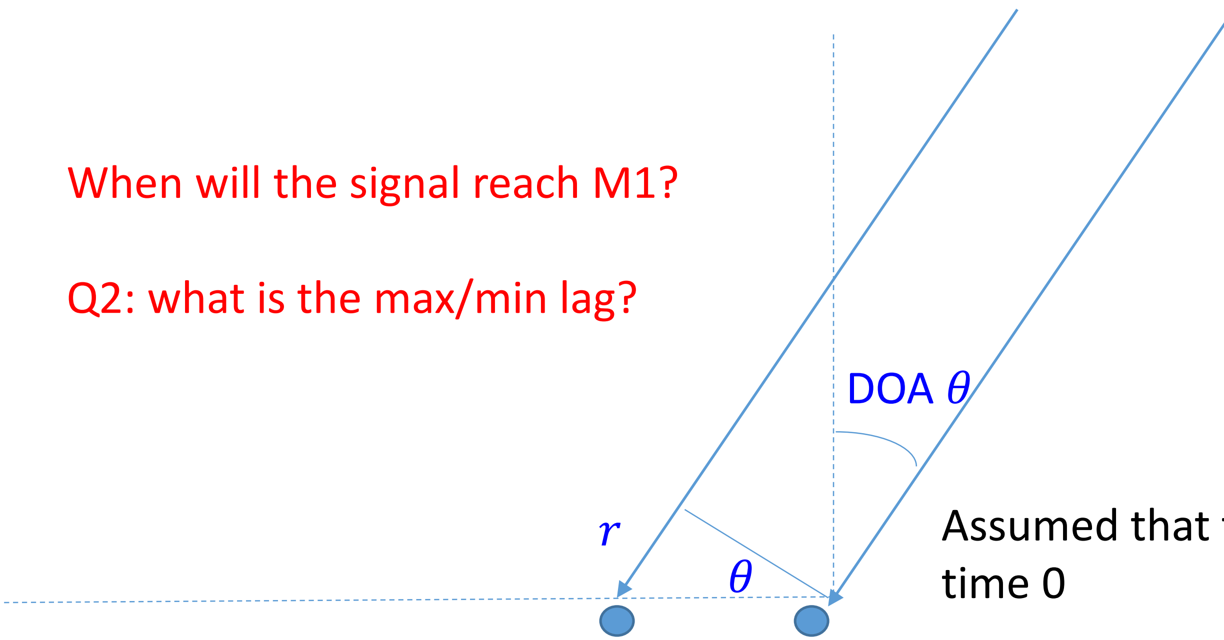
At time $t = r/c$, the signal reach M1!

For example, if $\theta = \pi/4$, $r = 0.17\text{m}$, then at $t = 0.0005\text{s}$, the signal reach M1!

If the sampling rate of the signal is 16000, then 16000*0.0005=8, M1 lags M2 8 samples, or says, M2 leads M1 8 samples (M2 lags M1 -8 samples).

When will the signal reach M1?

Q2: what is the max/min lag?



Basic knowledge

If the sampling rate of the signal is 16000, $r = d\sin(\theta)$

$t = \frac{r}{c} = -\frac{0.34}{340} = -0.001$, $16000 * -0.001 = -16$, M2 leads M1 -16 samples (M2 lags M1 16 samples).

Far field source $\theta = -\pi/2$

M1 (microphone 1) M2 (microphone 2)

$d = 0.34\text{m}$

$t = \frac{r}{c} = \frac{0.34}{340} = 0.001$, $16000 * 0.001 = 16$, M2 leads M1 8 samples (M2 lags M1 -16 samples).

$\theta = \pi/2$ Far field source

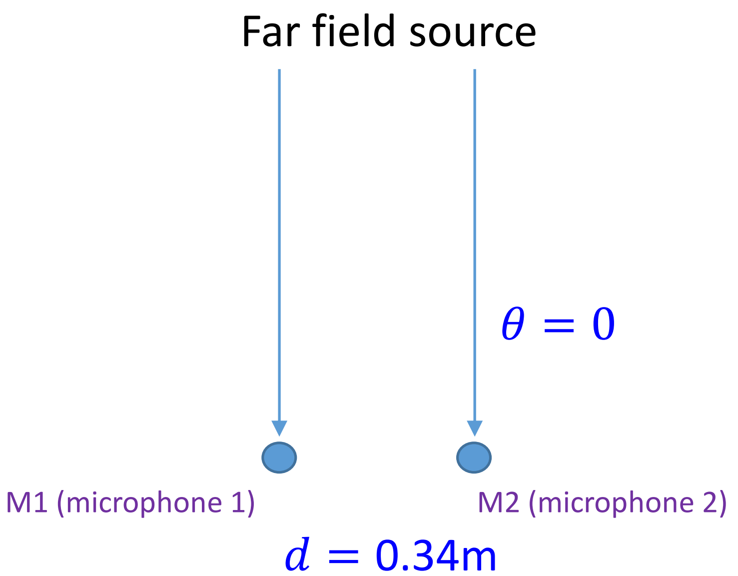
M1 (microphone 1) M2 (microphone 2)

$d = 0.34\text{m}$

Basic knowledge

If the sampling rate of the signal is 16000, $r = d\,\sin(\theta)$

$t = \frac{r}{c} = \frac{0}{340} = 0,\ 16000 * 0 = 0$, M2 leads M1 0 samples (M2 lags M1 -0 samples).



Conclusion:

- Under the sampling rate 16000, and $d = 0.34$m, the range of the 'M2 leads M1' of this microphone array is [-16, +16], totally 33 numbers, referring to 33 θ numbers.
- Using cross-correlation between M1 and M2 can calculate the lag between M1 and M2, and then calculate the θ, this is called time domain DOA estimation method.

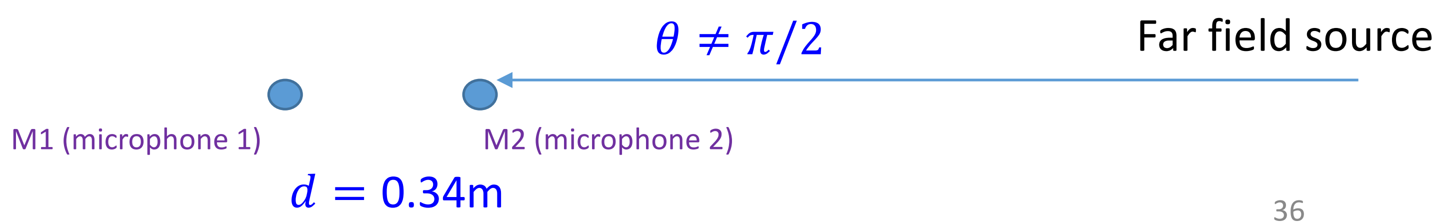
Basic knowledge

- Note that, the range of the 'M2 leads M1' is related to sampling rate N and distance between two microphones d.

- For example, if the sampling rate of the signal is $N = 44100$, $d = 0.34$, then the range of the 'M2 leads M1' of this microphone array becomes [-44.1, +44.1]. There is no 0.1 lag therefore the range is [-44, +44], totally 89 numbers, referring to 89 θ numbers.

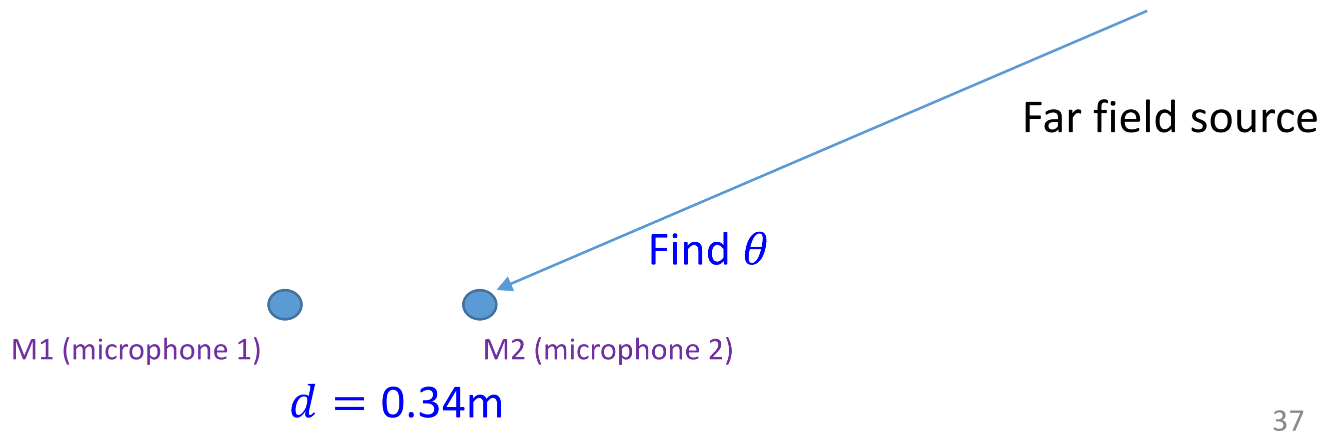
- However, in this case, when M2 leads M1 44 samples, $\theta \neq \pi/2$!

- In fact, $\theta = \arcsin(r/d)$, $r = \frac{M}{N} * c$, where N is the sampling rage, d is the distance between two microphones, M is the lag, here ranging in [-44, +44].



Basic knowledge

- In fact, $\theta = \arcsin(r/d)$, $r = \frac{M}{N} * c$, where N is the sampling rage, d is the distance between two microphones, M is the lag, here ranging in [-44, +44].
- For example, $N = 44100$, $d = 0.34$, c=340, M2 leads M1 M =+32, $\theta = 46.5^o$ (or 0.2584π)
- M2 leads M1 M =+44, $\theta = 86.14^o$.
- If the real θ is 90^o, and assume that that are no noise, the real lag is +44.1, however, this method cannot give the estimation '+44.1' but '+44' only, and estimated that the DOA is 86.14^o, the error is large!
- Such method is also called grid based method (means can only estimate the target in a grid, and when target is not in the grid, will give an estimation result of the grid most close to the target)



Advance part 2 of Experimental Report 2 (30 points)

Advance part 2: Now you are given 8 microphones, with id = 0:1:7. These microphones are located at (0, 0+0.085*id)m. The source is now located at least 60 meters away. (It is far field source in engineering). The sampling rate of the signal is $N = 44100$. (using the sound signal given)

1. Using only microphones with id 0 and 1(two microphones), using cross-correlation to calculate the lag (range: [-11, 11]). Please list the corresponding DOAs for all lags from -11 to +11 in one table. (there are totally 23 numbers)

Advance part 2 of Experimental Report 2 (30 points)

Advance part 2: Now you are given 8 microphones, with id = 0:1:7. These microphones are located at (0, 0+0.085*id)m. The source is now located at least 60 meters away. (It is far field source in engineering). The sampling rate of the signal is $N = 44100$. (using the sound signal given)

2. Now, assume that the signal comes from one of the 23 DOAs from above (randomly in every independent run), together with random Gaussian noise (zero mean), and now you are going to estimate the DOA! Design your main function to describe this system as well as your method to estimate the DOA. Please give the flow chart of your program, the estimation result (correct detection percentage* or other indicators) of the DOA versus SNR(dB), and your analysis.

*: if the estimated DOA is exactly equals to the true DOA, this is one correct detection. In 1000 independent runs, if correct 980 times, then the correct detection percentage is 0.98; 1.0 correct detection percentage means the methods can correctly estimate the DOA in this SNR case. Note that the larger the SNR, the higher the 'correct detection percentage'

Advance part 2 of Experimental Report 2 (30 points)

Advance part 2: Now you are given 8 microphones, with id = 0:1:7. These microphones are located at (0, 0+0.085*id)m. The source is now located at least 60 meters away. (It is far field source in engineering). The sampling rate of the signal is $N = 44100$. (using the sound signal given)

3. Now, assume that the signal comes from one of the 23 DOAs from above (randomly in every independent run), together with random Gaussian noise (zero mean), and now you are going to estimate the DOA using all the 8 microphones! Design your main function to describe this system as well as your method to estimate the DOA. Please give the flow chart of your program, the estimation result (correct detection percentage* or other indicators) of the DOA versus SNR(dB), and your analysis. Note that you can compare your result here to point 2 above and give analysis.

*: if the estimated DOA is exactly equals to the true DOA, this is one correct detection. In 1000 independent runs, if correct 980 times, then the correct detection percentage is 0.98; 1.0 correct detection percentage means the methods can correctly estimate the DOA in this SNR case.
Note that the larger the SNR, the higher the 'correct detection percentage'

Extra part of Experimental Report 2 (+10 points)

Extra part: Now you are given 8 microphones, with id = 0:1:7. These microphones are located at (0, 0+0.085*id)m. The source is located at least 60 meters away. (It is far field source in engineering). The sampling rate of the signal is $N = 44100$. (using the sound signal given)

Now, assume that the signal comes from $[-\pi/2, +\pi/2]$ (randomly in every independent run), as describe above, the grid base method can only estimate 23 DOAs. If the real DOA is not in the 23 DOA grid, the error will be large! Is there any method to get better estimation? Please try it and give your result.