



Lab S-2: Direction Finding: Time-Difference or Phase Difference

Pre-Lab: Read the Pre-Lab and do all the exercises in the Pre-Lab section *prior to attending lab*.

Verification: The Exercise section of each lab should be completed **during your assigned Lab time** and the steps marked *Instructor Verification* signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, demonstrate the step to your instructor. Turn in the completed verification sheet before you leave the lab.

Lab Homework Questions: The Lab-Homework Sheet has a few lab related questions that can be answered at your own pace. The completed Lab-HW sheet should be turned in at the beginning of the next lab.

1 Pre-Lab

Please read through the information below prior to attending your lab.

Objective: The objective of this lab is to learn how the outputs from two spatially separated sensors that receive signals from the same source can be used to estimate the direction to the source of the signal. The key to this processing is phase difference or time difference of arrival (TDOA) at the two receivers.

1.1 Publishing MATLAB Code

When documenting MATLAB code for lab reports, the “publish” feature in MATLAB provide an easy way to produce .html file directly from an M-file. This publish feature is provided with one of the tabs in MATLAB’s edit window. The following help describes the process:

http://www.mathworks.com/help/matlab/matlab_prog/publishing-matlab-code.html

The basic idea is to write comments according to some simple formatting rules. The following example from the MATLAB documentation illustrates the process with an example that comes from DSP (i.e., summation of harmonic sinusoids in a Fourier series).

```
edit(fullfile(matlabroot,'help','techdoc','matlab_env', ...  
             'examples','fourier_demo2.m'))
```

- Use the command above to open the M-file `fourier_demo2.m` in the MATLAB editor
- Run the `fourier_demo2.m` M-file to see the plots that it creates.
- Publish `fourier_demo2.m` to create `fourier_demo2.html`, and then open `fourier_demo2.html` in a web browser and view the plots that show the sums of harmonic sinusoids.

1.2 Overview

There are four specific cases that will be considered in this lab:

1. Sinusoidal source where the receiver signals have different phases.
2. Sinusoidal source with a wide receiver separation so that phase ambiguities must be taken into consideration.



3. Speech signal at the source so that time difference of arrival (TDOA) must be estimated.
4. Noise signal at the source which requires a *template matching* method to find the TDOA.

1.3 Direction Finding

Why do mammals have two ears? One answer is that a brain can process acoustic signals received at the two ears and determine the direction to the source of the acoustic energy. It might be tempting to think that the ears can sense direction based solely on amplitude changes, but phase plays a very important role also. Using sinusoids, we can describe and analyze a simple scenario that demonstrates “direction finding” in terms of phase differences. This same principle is used in many other applications including radars that locate and track airplanes.

1.4 Direction of Arrival (DOA) Sensing with Microphones

Consider a simple measurement system that consists of two microphones that can both hear the same source signal. If the microphones are placed a small distance apart, then the sound must travel slightly different paths from the source to the receivers. When the travel paths have different lengths, the signals arrive at different times. Since time shift corresponds to phase, we say that the received signals arrive “out of phase.”

The received signal at one microphone, called $r(t)$, is a delayed copy of the transmitter signal $s(t)$. If the time delay from source to receiver is τ_{sr} , then we can write

$$r(t) = s(t - \tau_{sr})$$

where $s(\cdot)$ is the transmitted (sinusoidal) signal.¹ The travel time τ_{sr} can be computed easily once we know the speed of sound and the locations of the source and receiver(s).

We can compare the signals from two receivers in many different ways. Consider the case of one source transmitting the signal $s(t)$ to two receivers. The received signals could be labelled with subscripts 1 and 2:

$$\text{Receiver \#1: } r_1(t) = s(t - \tau_{sr_1})$$

$$\text{Receiver \#2: } r_2(t) = s(t - \tau_{sr_2})$$

where τ_{sr_1} is the propagation time from the source to Receiver #1, and τ_{sr_2} the propagation time from the source to Receiver #2.

Suppose that the geometry of the problem is planar as shown in Fig. 1. The two receivers are located on the y -axis at $(0, 0)$ for receiver #1 and $(0, d)$ for receiver #2. The inter-sensor spacing (d) is positive, i.e., $d > 0$. The source is located at a distance of d_1 at an angle of θ . Thus the distance from the source to the receivers is d_1 for source to receiver #1, and d_2 for source to receiver #2. With some elementary geometry (or complex number vector addition), it is relatively easy to derive the following formula for d_2

$$d_2 = d_1 \sqrt{1 - \frac{2d}{d_1} \sin \theta + \frac{d^2}{d_1^2}} \quad (1)$$

Now we must make some approximations that will lead to a simple result. When $d_1 \gg d$, we can drop the second-order term to obtain

$$d_2 \approx d_1 \sqrt{1 - \frac{2d}{d_1} \sin \theta} \quad (2)$$

¹For simplicity we ignore propagation losses. Usually, the amplitude of an acoustic signal that propagates over a distance R is reduced by an amount that is inversely proportional to R or R^2 .

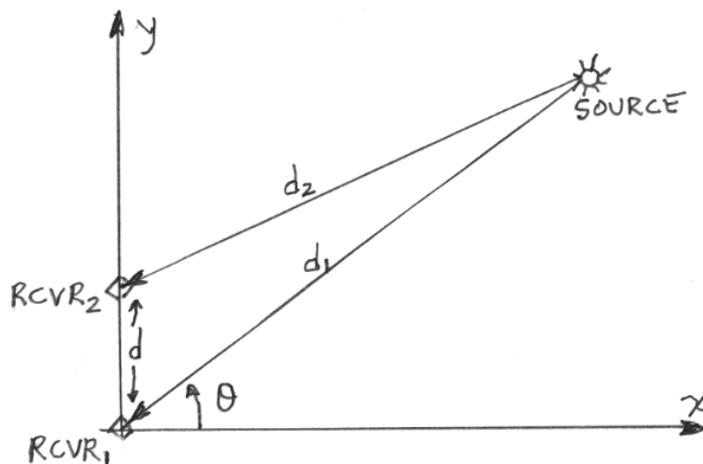


Figure 1: Source with two receivers. The angle θ is shown as positive, but θ could be negative if the source lies in the fourth quadrant. When $\theta > 0$ the signal arrives first at receiver #2. The two receivers on the y -axis cannot distinguish left from right, so we assume the source does not lie in the second or third quadrants.

Next we approximate the square root because when $d_1 \gg d$ the term $(2d/d_1) \sin \theta$ is very small and we can use the first two terms of a Taylor series expansion for the square root

$$\sqrt{1 - \epsilon} = 1 - \frac{1}{2}\epsilon + \text{higher order terms} \quad (3)$$

Thus,

$$d_2 \approx d_1 \left(1 - \frac{1}{2} \frac{2d}{d_1} \sin \theta \right) \quad (4)$$

and we obtain a very simple formula for θ :

$$d_1 - d_2 \approx d_1 - \underbrace{(d_1 - d \sin \theta)}_{\approx d_2} = d \sin \theta \quad (5)$$

This difference in propagation distance leads to two observations:

1. The time it takes the source signal to propagate to the two receivers is different. Thus, if $d_1 > d_2$ the signal arrives first at receiver #2. Assuming that the velocity of propagation is c , the time difference is

$$\Delta \tau = \tau_1 - \tau_2 = ?$$

Please determine this formula. The righthand side will depend on the inter-sensor separation d , angle θ , and velocity c .

2. Since time delay is related to phase, when the source signal is a sinusoid the phase is different at the two receivers. Once again, it is possible to write a simple formula for the phase difference:

$$\Delta \varphi = \varphi_1 - \varphi_2 = ?$$

Once again, determine the formula for the righthand side of this equation, which will depend on the frequency ω , as well as d , θ , and c . Pay attention to the sign of the phase with respect to time delay.

Note: In a later section the 2π -ambiguity of the phase will be an issue. In other words, since the phase of a sinusoid is ambiguous by integer multiples of 2π , there is more than one value for the phase difference.



It is useful to cross check the formulas that were derived above. For example, when the phase difference (or, equivalently, TDOA) is zero what do you expect to get for the angle θ ?

1.5 Template Matching

In some situations, it is hard to determine the time shift between two signals, e.g., when there is no prominent peak or other feature that can be tracked between the signals. For this case, we need a procedure to match a segment of one signal with the corresponding segment in the other signal.

Consider the following problem where we want to find one vector inside of another one. For a specific case suppose that \mathbf{xx} is a length-10 vector and \mathbf{yy} is a length-30 vector. If the vector \mathbf{xx} is “inside of” the vector \mathbf{yy} , then there is some index k such that the subvector $\mathbf{yy}(k:k+9)$ is equal to \mathbf{xx} . If we want to write a MATLAB function that finds the match, we must find the starting index k , given \mathbf{xx} and \mathbf{yy} . This requires that we test all possible values for k . For the sizes given above, there are 21 possible length-10 subvectors within the length-30 \mathbf{yy} vector, i.e., $\mathbf{yy}(k:k+9)$ for $k=1, 2, 3, \dots, 21$.

In practice, the test should not be an equality test because real signals acquired at sensors always have a small amount of random perturbations in their values (called noise). As long as the difference $\mathbf{yy}(k:k+9) - \mathbf{xx}$ is very small, we would have a match. One way to quantify “very small” is to use a metric such as the *Average Magnitude Difference Function* (AMDF) which was first popularized in speech processing for pitch period estimation.

$$E_{\text{AMD}}(k) = \frac{1}{L} \sum_{n=1}^L |y[n+k-1] - x[n]|$$

where L is the length of the shorter vector, $x[n]$ in this case, and k is the starting index in the longer vector. For each value of k , $E_{\text{AMD}}(k)$ is a scalar value that measures the match. If there was a perfect match, explain why the minimum value of $E_{\text{AMD}}(k)$ versus k is zero. The match may not be perfect, so the practical procedure is to do this test for many values of k , and then find the best match at the index of the minimum. *Note:* when $y[n+k] = 0$ the value of $E_{\text{AMD}}(k)$ is $M_x = \frac{1}{L} \sum_{n=1}^L |x[n]|$, so we should expect the minimum value of $E_{\text{AMD}}(k)$ to be much less than M_x . This fact can be useful in setting a threshold for detecting the minimum when the signals are noisy.

1.6 Random Parameters Generated by LabDFgen2015

A simple MATLAB function is supplied to generate random parameters for your work in this lab.

```
[r1,r2,fs] = LabDFgen2015('UserID',Part)
```

The first input is a string such as your login/UserID in order to identify different individuals doing this lab; the second argument is an integer that selects one of the three parts of the Lab Exercise.² The output displayed in the MATLAB command window gives the values of the parameters needed in one of the specific sections below. For parts 2 and 3, the function also returns vectors for the two receiver signals $r1$ and $r2$, and the sampling rate fs . The function has been converted to MATLAB’s p-code format to hide the MATLAB code, because the parameter generation process involves knowing the solution to the exercise.

²Your instructor might require a fixed UserID so that there would be only one solution to the lab.



2 Lab Exercise

For the instructor verification, you should demonstrate that you understand concepts in a given subsection by answering questions from your lab instructor. It is not necessary to do everything in the subsections, i.e., skip parts that you already know. The Instructor Verification is usually placed close to the most important item, i.e., the one most likely to generate questions from your instructor.

2.1 Direction from Phase Difference

Use the `LabDFgen2015(UserID, 1)` function with its second argument set to 1 to generate two phase measurements, along with an intersensor distance d , and the propagation velocity c . These values are displayed in the MATLAB command window. Use the theory of direction finding to determine the direction to the source (θ). Write your method and results (and explanation) on the verification page

Instructor Verification (separate page)

2.2 Ambiguous Direction

Start from the phases generated previously, i.e., `LabDFgen2015(UserID, 1)`. Since the phase of a sinusoid might not be unique, the same is true of the phase difference. By changing $\Delta\phi$, show that there is at least one more direction *in the first or fourth quadrant* that satisfies the equations used in the previous part.

Note: this ambiguity occurs in this sensor array scenario because the sensor spacing is larger than $\frac{1}{2}\lambda$, where $\lambda = c/f$ is the wavelength of the propagating wave.

Determine this second source direction (θ_2). Write your results (and explanation) on the verification page.

Instructor Verification (separate page)

2.3 Template Matching Function

Write a MATLAB function that can find the best location of a short vector inside of a long vector. The output of the function should be an integer index that gives the best estimate of the starting index of the matched subvector within the longer vector. The metric for matching should be the AMDF (Average Magnitude Difference function) criterion. An outline of the M-file is given in Fig. 2.

Note: the `plot` function called in the middle of the function should only be used during debugging; once the function is working correctly that line should be commented out.

The threshold should be a small number that depends on the expected noise level because the subtraction in E_{AMD} produces errors on the order of the noise level. Picking the threshold to be 2–3 times the noise level would be a reasonable value. Use the following test case where the noise level is 0.05:

```
vShort = 0.7*[1, 1, -1, 1, -1, 1, 1];  
vLong = rand(1, 20);  
vLong(6:12) = vShort;  
vLong = vLong + 0.05*randn(size(vLong)); % add noise
```

If you turn off the noise, the function should give an exact match with $E_{\text{AMD}} = 0$ at its minimum. Furthermore, in order to test whether the threshold works, make both vectors random so there would be no match.

Instructor Verification (separate page)



```
function kShift = templateMatch(vShort,vLong)
%
% length(vShort) must be less than (or equal to) length(vLong)
% kShift = best estimate of the starting index
% kShift can only be nonnegative
% use kShift=-1 to indicate the No Match Condition when the AMDF is too big
L = length(vShort);
kEnd = ?? %- depends on length(vLong) and length(vShort)
for kk=1:kEnd
    E(kk) = ??
end
[Emin,kmin] = min(E);
plot(0:kEnd-1,E),title('Diagnostic Plot of E vs. shift')
kShift = ??;
threshold = ??;
if Emin>threshold
    kShift = -1;
end
```

Figure 2: MATLAB code for template matching with AMDF.

2.4 Direction from a Speech Signal Source

In this part, use the `[r1,r2,fs]=LabDFgen2015(UserID,2)` function with its second argument set to 2 to generate two time-shifted speech signals, along with parameter values for the intersensor distance d , the propagation velocity c , and the sampling rate f_s . The values of distance d and propagation velocity c are displayed in the MATLAB command window.

Determine the direction to the source (θ). Write your method and results on the verification page.

Instructor Verification (separate page)

3 Homework: Direction from a Noise Signal Source

Use the `[r1,r2,fs]=LabDFgen2015(UserID,3)` function with its second argument set to 3 to generate two time-shifted noise signals, along with an intersensor distance d , the propagation velocity c , and sampling rate f_s . The values of distance d and propagation velocity c are displayed in the MATLAB command window. Use your template matching function (based on the AMDF) to obtain an estimate for the TDOA from which you can determine the direction (θ). Write your method and results on the Lab-HW page.

In order to use your template matching function it is necessary to take a small part of one signal and try to find it within the other signal. One suggested approach would be to take the “center-half” of the first signal and match it to the second signal. *Center-half* means drop the first 25% and last 25% of the points in the vector. Once you get the best match, the shift index must be compensated for the center-half operation. Finally, the shift index must be converted to a time (in secs) using the sampling rate.



Lab Direction Finding INSTRUCTOR VERIFICATION SHEET

Turn this page in to your instructor before the end of your scheduled Lab time.

Name: _____ UserID: _____ Date: _____

Part 2.1 Direction from phases of two sinusoids. Method and results.

Verified: _____ Date/Time: _____

$d =$ $c =$ $\varphi_1 =$ $\varphi_2 =$ $\Delta\varphi =$

$\theta_1 =$

Part 2.2 Second direction from phases of sinusoids. Method and results.

Verified: _____ Date/Time: _____

$\theta_2 =$

Part 2.3 Exhibit your template matching M-file on a test case.

Verified: _____ Date/Time: _____

Part 2.4 Direction from time-shifted speech signals. **List the measured time shifts.** Method and results.

Verified: _____ Date/Time: _____

$\tau_1 =$ $\tau_2 =$ $\Delta\tau =$ $\theta =$



Lab: Direction Finding

LAB HOMEWORK QUESTION

Turn this page in to your instructor at the very beginning of your next scheduled Lab time.

Name: _____ UserID: _____ Date: _____

Use your template-matching M-file to estimate direction when the transmitted signal is random noise.

1. Run the `LabDFgen2015(UserID,3)` function to obtain the two noise signals and the system parameters. List the system parameters below
2. Determine the relative shift index obtained from template matching with the AMDF. To describe how you found the shift, sketch a plot of AMDF versus shift (k) that you can make in MATLAB.
3. Then convert the shift index to TDOA (in secs).
4. Determine the direction angle θ . Show the calculation:
5. Turn in this page at the very beginning of your next scheduled lab period.