

---

**Lab S-1: Complex Exponentials – Source Localization**

---

**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 Introduction and Overview

The goal of this laboratory is to provide an application where phasors (complex exponentials) are used to locate a source signal. A sensor pair is used to find the direction to the source, and then directions from two (or more) sensor-pairs are combined to “triangulate” the location of the source.

## 2 Pre-Lab

There are many applications where it is important to be able to locate the precise position of an object. The Global Positioning System (GPS) provides this capability by using multiple satellites that emit radio frequency (RF) signals that can be processed in a single receiver to calculate 3-D position anywhere on the earth. Another application would be locating the position of a cell phone caller in order to provide a **911** service the ability to automatically determine the source location of a distress call. This can be done through a process called *triangulation*, if multiple cell towers were receiving the wireless signal. This same principle is used in many other applications including radars that locate and track airplanes.

In this lab exercise we will use sinusoids to describe and analyze a simple scenario that demonstrates *acoustic source localization* in terms of phase differences and simple geometry. One approach to source localization consists of two distinct steps:

1. *Direction Finding* (DF) which can be done using a pair of receivers to determine the direction of arrival of the acoustic signal (Fig. 1). This step actually consists of three parts when the received signal is a sinusoid: the phase difference between the received signals is measured, then the phase difference is converted to time delay, and finally the direction is found from the relative time delay.
2. *Triangulation* which uses two (or more) DF measurements made by distinct receiver-pairs to calculate the location of the source (Fig. 2). The positions of the receiver pairs must be known.

### 2.1 Direction Finding (DF) with Microphone Pairs

Consider a simple measurement system that consists of two microphones that can both hear the same source signal as shown in Fig. 1. If the two microphones are placed a short distance apart, then the sound must travel different paths from the source to the receivers. Usually the travel paths have different lengths, so the two signals arrive at different times. This time difference can be measured and then used to calculate the

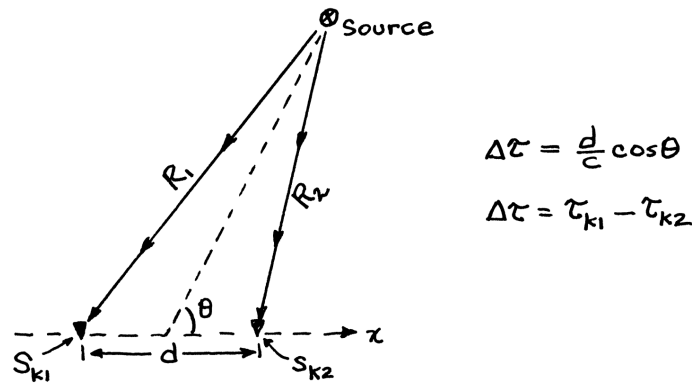
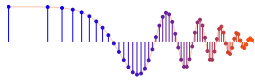


Figure 1: Estimating the angle of arrival ( $\theta$ ) using a pair of receivers. As drawn here the signal arrives at receiver  $S_{k_2}$  first, so the difference between arrival times,  $\tau_{k_1} - \tau_{k_2}$ , would be a positive number.

direction ( $\theta$ ) from the receiver pair to the source. A convenient reference point for measuring the angle is the midpoint between the receivers. For simplicity, we assume that the receivers are placed along the  $x$ -axis and that the angle  $\theta$  is measured with respect to the  $x$ -axis.

The received signal at a microphone, denoted as  $r(t)$ , is a delayed copy of the transmitter signal  $s(t)$ . Thus we could write

$$r(t) = s(t - \tau_d)$$

where  $\tau_d$  is the amount of time it takes for the signal to travel from the source location to the receiver and  $s(\cdot)$  is the transmitted (sinusoidal) signal.<sup>1</sup> The travel time  $\tau_d$  can be computed easily once we know the speed of sound ( $c$ ) and the locations of the source and the receivers.

Consider the case of one source transmitting the signal  $s(t)$  to two receivers labeled  $S_{k_1}$  and  $S_{k_2}$  in Fig. 1:

$$\text{Sensor } S_{k_1}: r_{k_1}(t) = s(t - \tau_{k_1})$$

$$\text{Sensor } S_{k_2}: r_{k_2}(t) = s(t - \tau_{k_2})$$

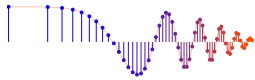
The index  $k$  denotes different receiver pairs, because later on we will need two receiver pairs in order to perform triangulation (see Section 3.3). The time delay  $\tau_{k_1}$  is the propagation time from the source to receiver  $S_{k_1}$ , and  $\tau_{k_2}$  the propagation time from the source to receiver  $S_{k_2}$ . Finally, we assume that the source signal is a sinusoid with frequency  $f$ , the received signals are also sinusoids at  $f$ , but with different phases.

- (a) Assume that the source is at  $(x_T, y_T)$  and the receivers have coordinates  $(x_{k_1}, y_k)$  and  $(x_{k_2}, y_k)$ . Since the  $y$ -coordinates are equal in Fig. 1, we can draw an  $x$ -axis through the sensors to use as a reference line for measuring the angle  $\theta_k$  at the  $k^{\text{th}}$  microphone pair.
- (b) Assume that the sinusoidal signal at the source is  $s(t) = \cos(2\pi ft)$ . Write the formulas for the received signals  $r_{k_1}(t)$  and  $r_{k_2}(t)$ .

### 2.1.1 Direction Finding Approximation

The angle of arrival ( $\theta$ ) in Fig. 1 can be computed (approximately) from the time difference between the two sinusoids arriving at the two receivers. If  $\Delta\tau = \tau_{k_1} - \tau_{k_2}$  is the time difference between the arrivals at

<sup>1</sup>For 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  $R$ .



receivers  $S_{k_1}$  and  $S_{k_2}$ , then the formula for calculating the angle is

$$\Delta\tau = \frac{d}{c} \cos \theta \quad (1)$$

where  $c$  is the velocity of sound, and  $d$  is the receiver spacing in meters, both of which have to be known values. It is useful to have a single reference point for  $\theta$ , so the midpoint between the two receivers is chosen. The formula in (1) involves two important constraints:

1. It is an *approximation*, so the value calculated for  $\theta$  might not be exact. The approximation improves as the distance between the source and receivers increases. When  $R_{k_1}$  (and  $R_{k_2}$ ) is more than 10 times greater than the inter-receiver distance  $d$ , the formula (1) has very little error; when the ratio is 50 or more the approximation is almost perfect.
2. The formula (1) cannot tell “up from down.” Therefore, we must restrict our attention to sources in a half-space. For this lab exercise, we assume that any source of interest lies above the receivers, i.e.,  $y_T > y_k$  in Fig. 2.

### 2.1.2 Data Generation M-file

The first task is to write a MATLAB program to generate data that simulates the propagating signal from the source to the receivers. The terminology *sensor* is used to refer to an individual microphone. The M-file must be called **rcvpairsigs**, and its functionality is defined in the comments below. The output of this function is the complex amplitudes of the sinusoids at all the sensors (receivers). The inputs to **rcvpairsigs.m** must be the receiver locations in  $(x, y)$  space and the source location  $(x_T, y_T)$ . Note that multiple sensors can be done by giving a matrix of receiver positions.

```
function Camps = rcvpairsigs( xT, yT, xySensors )
%RCVPAIRSIGS generate data for a sinusoid emitted from a single
%           source location and received at multiple sensors (microphones)
%
% xT = x position of the source
% yT = y position of the source
% xySensors = 2 by Ns matrix of (x,y) positions of the receivers
%           where Ns = number of sensors (receivers)
% Camps = matrix of COMPLEX amplitudes, one for each sensor
%           size(Camps) is 1 by Ns
c = 333;    %-- sound velocity in meters/sec
freq = 222; %-- frequency in Hz
```

The output from the **rcvpairsigs.m** function is a vector of complex amplitudes, one for each sensor (microphone). Since the signal arriving at any one microphone is a sinusoid, we only need to know its amplitude, frequency and phase. The frequency is fixed at  $f = 222$  Hz for this lab; the amplitude and phase of the sinusoid are encoded in the complex amplitude.

## 2.2 Testing the Data Generation

The **rcvpairsigs.m** program can assign any fixed amplitude and phase to the source sinusoid, so  $A = 1$  and  $\varphi = 0$  is acceptable. However, both signals that are going to be used by one receiver-pair must be generated by one call to **rcvpairsigs.m**. This provides a single snapshot of the source signal propagating to both microphones in one receiver-pair.



- (a) In order to test your M-file for `rcvpairsigs.m`, put the source at  $x_T = -10$  m and  $y_T = 17.32$  m and two sensors (receivers) at  $(-0.4, 0)$  m and  $(0.4, 0)$  m.<sup>2</sup> Use your M-file to get the complex amplitudes and then determine the phases of the received signals at the two sensors.
- (b) For the case described in the previous part, determine the true values of the received phases calculated from the theory (by hand); and check with the M-file output.

### 3 Lab Exercise: Source Localization

#### 3.1 Phase Difference and Time Shift

- (a) From the results in Sect. 2.2(a), calculate the phase difference (see comments below). Then use the phase difference to determine the relative time shift between the two sinusoids.  
Note: recall the formula that relates phase to time shift for a sinusoid.
- (b) Calculate (by hand) the true value of the time shift from the propagation distances and the velocity of sound. Verify that you get exactly the same value as in the previous part.
- (c) Finally, convert the time shift to an estimated angle  $\theta$ . Compare  $\theta$  to the true value which can be calculated from the geometry defined in part (a). Note that the midpoint of this receiver pair is  $(0, 0)$ .

**Instructor Verification** (separate page)

##### 3.1.1 Comment About Phase Difference

Computing the difference of two angles might be done in the obvious way as a subtraction ( $\angle z_1 - \angle z_2$ ), but it is better to use the fact that complex conjugating negates the angle of a complex number. Thus, finding the angle of  $(z_1 z_2^*)$  gives the same result as subtracting the angles. Questions might arise because angles are equal “modulo- $2\pi$ .” It is usually best to have the smallest angle, i.e., the one between  $-\pi$  and  $+\pi$ .

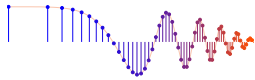
Think about the pros and cons of these two methods for the direction finding problem. For example, what happens when  $\angle z_1 = 0.9\pi$  rad and  $\angle z_2 = -0.95\pi$  rad? How far apart are these two angles? Subtraction gives  $1.85\pi$  rad, but  $-0.15\pi$  rad is the same angle.

#### 3.2 Direction Finding (DF) Processing Function

Write a short M-file that takes the complex amplitudes from the data generation function and outputs the DF angle  $\theta$ . In other words, put the processing steps in Sect. 3.1 into an M-file. Call the function `DFproc`, and define its inputs and outputs as needed.

**Instructor Verification** (separate page)

<sup>2</sup>Sensor #1 must be to the left of Sensor #2, as in Fig. 1.



### 3.3 Information Needed for a Triangulation Function

We can locate the position of a source if we have angles from *two receiver-pairs*. It is a simple problem

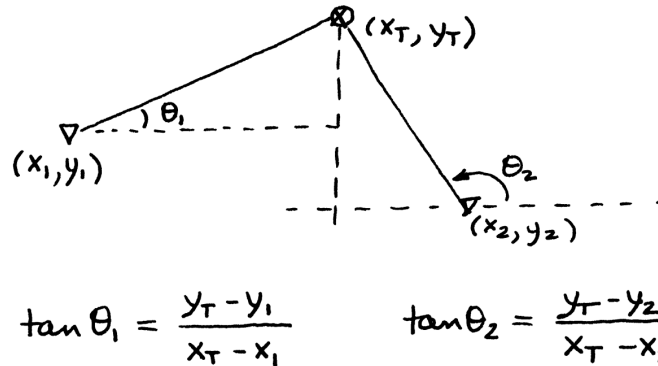


Figure 2: Determine source location by intersecting two rays that emanate at angles  $\theta_1$  and  $\theta_2$  from two receiver-pairs.  $(x_k, y_k)$  are the **midpoints** of the receiver-pairs that perform direction finding to estimate  $\theta_k$ .

in geometry to find the source location from the two angles. As shown in Fig. 2, we would draw rays emanating from the sensor pairs and find where the rays intersect. The receiver-pair midpoints are the origin points for each ray. As an example of this calculation, assume we have one receiver-pair with its midpoint at  $(x_1, y_1) = (0, 1)$  and a second receiver-pair with midpoint  $(x_2, y_2) = (1, 0)$ . If the angles from DF processing are  $\theta_1 = 0$  and  $\theta_2 = 2\pi/3$ , then the intersection gives a source position at  $(0.4226, 1)$ .

To calculate the source location from the angles of arrival, the M-file **triangulate.m** has been written for you. You can download the p-code version **triangulate.p** from the Lab link. Use the example above to verify that **triangulate** works correctly.

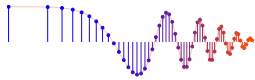
```
function xyLoc = triangulate( angles, xyMidPts )
%TRIANGULATE determine source location from angle of arrivals
%               at multiple receiver-pairs
%
% angles = Ns-element vector of angles
%               where Ns = number of receiver-pairs
%
% xyMidPts = 2 by Ns matrix of the (x,y) midpoints of the receiver-pairs
%               x coords are in the first row; y coords in the second row
%
% xyLoc = 2 by 1 vector of the (x,y) location of the source.
```

### 3.4 Complete Simulation for Source Location

Now it is time to put all the steps together and show the entire system that performs direction finding and source localization. Use your previously written functions to produce data and calculate the angles of arrival, and the P-code version of the M-file **triangulate** to get the source location. Recall that the frequency is fixed at  $f = 222$  Hz and the speed of sound is assumed to be  $c = 333$  m/s. Do the following cases:

- (a) Use two sensor pairs, i.e., four receivers altogether, at the following locations:

Sensor $S_{1_1}$ at $(-1.2, 0)$	Sensor $S_{2_1}$ at $(1.9, 1)$
Sensor $S_{1_2}$ at $(-0.8, 0)$	Sensor $S_{2_2}$ at $(2.1, 1)$



The first sensor pair is centered at  $(-1, 0)$  m, while the second sensor pair is centered at  $(2, 1)$  m. Use your function `rcvpairsigs.m` to create data for a source at  $(1, 5)$  m. Then apply the function `DFproc` to compute the angles, and the function `triangulate` to get an estimate of the source location.

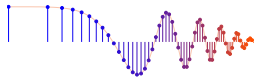
**Instructor Verification** (separate page)

- (b) Repeat the signal generation and processing when the source location is  $(1.9, 1.1)$  m.
- (c) Summarize your results by giving the true locations and the estimated locations for both cases. Calculate the *estimation error* as the distance between the true and estimated locations. Explain why the error for the second case is worse.

**Instructor Verification** (separate page)

### 3.5 Homework

Derive the equations needed for the inner workings of `triangulate`, and use these to write MATLAB code.



## Lab: Source Localization INSTRUCTOR VERIFICATION SHEET

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

Name: \_\_\_\_\_ UserID: \_\_\_\_\_ Date: \_\_\_\_\_

Part 3.1 Determine the phase difference, time shift and angle  $\theta$ :

$$\Delta\varphi = \quad \Delta\tau = \quad \theta =$$

Provide brief explanation.

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 3.2 Write a MATLAB function DFPROC to obtain the direction  $\theta$  from the measured complex amplitudes. Write the code below.

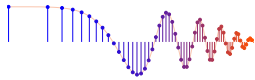
Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 3.4(a,b) Location estimate for source at (1, 5) m; repeat for source at (1.9, 1.1) m.

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 3.4(c) Calculate the *estimation error* as the distance between the true and estimated locations for both cases. Explain why the error is not zero, and why the error for the second case is worse.

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_



## Lab: Source Localization LAB HOMEWORK QUESTION

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

Name: \_\_\_\_\_ UserID: \_\_\_\_\_ Date: \_\_\_\_\_

Derive the equations needed for the inner workings of **triangulate**.

- The inputs and outputs are defined by `xyLoc = triangulate( angles, xyMidPts )`
- Writing equations for this situation can be done several ways, but an easy approach is to generate two linear equations in the two unknowns  $(x_T, y_T)$  by using the tangent definitions in Fig. 2.
- Put the equations in standard matrix form  $\mathbf{Ax} = \mathbf{b}$ , so they can be solved via matrix inverse, or with MATLAB's backslash operator `x=A\b`.
- Write the few lines of MATLAB code needed to implement your **triangulate** method.