Acoustic Algorithm

MAE 4273 Experimental Fluid Dynamics
Fall 2018

Jonathan C. Cook
Jared Hartzler
Christian Negratti
Jacob Shafer

United States of America
Date Received: October 1, 2018
Date Submitted: December 7, 2018

Abstract

The research is intended to further develop and prove novelty ideas regarding the infrasound produced by tornadoes during formation and existence for increased warning times as well as location of a storm. The facility used to progress the research is a triangular array of three microphones set up at Oklahoma State University idealized for recording infrasound, which have the ability to record continuously. Although the research for localization focuses on infrasound, the localization technique can be approached with data in the lower acoustic frequency range because there is no significant drop off up to 100 Hertz. The analysis of the data being emitted by a subwoofer is given a direction by the tasks of concomitancy, characteristic signature, coherence and directionality. The arrival of an output signal by the concomitant nodes provides evidence of the retrieval of a source when output, and the absence thereof. The characteristic signature is accomplished in the Fourier domain by comparing the known input frequency of the speaker, with the frequency output or retrieved by the microphones. The coherence is solved for using two methods. One method is the coherence squared method which is a ratio determining how ideal the recording microphones are. The other coherence method involves the magnitudes and standard deviations in the frequency domain, which provides knowledge for the which of the two nodes in a given triangle to solve for the differences and find the location of the point. Given the knowledge of the second coherence method, a time distance of arrival (TDOA) technique is determined to solve for sources at unknown locations.

Contents

Nomenclature

C_{xy} Coherence with magnitudes and standard deviations
$C_{xy}(f)$ Magnitude squared Coherence (Range 0 to 1)
c Speed of sound (m/s)
d Distance from the source to the microphone (meters)
f Frequency (Hz)
t Time (seconds)
$P_{xy}(f)$ Phase angles of input and output
$P_{xx}(f)$ Real portion of input signal
$P_{yy}(f)$ Real portion of output signal
α Angle in a triangle
β Angle in a triangle
γ Angle in a triangle
$\Psi(f)$ Magnitude at a frequency
σ Standard deviation

1 Introduction

Acoustic localization is the identification of a sound source as well as determination of the location and frequency of the signal produced by a source. Infrasound is located at a frequency lower than a humans threshold of hearing, 20 Hertz (Hz), as shown in Figure 1. Infrasound is transmitted by naturally occurring events such as tornadoes and earthquakes, as well as man made events such as nuclear explosions. Recent

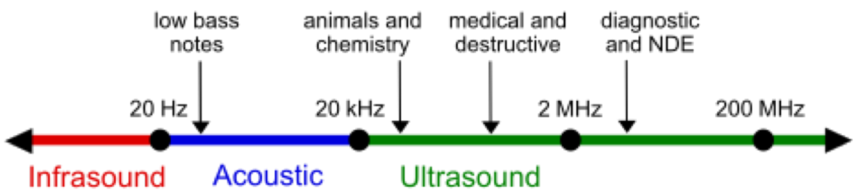


Figure 1: Depiction of the sound spectrum
Source:https://en.wikipedia.org/wiki/Frequency

developments by Elbing et al. (2018) have discovered the production of infrasound by tornadoes during formation and as well as existence of the tornado. The goal of the research is to increase warning times of tornadoes from minutes to possibly close to an hour. In a recent study, by the use of an array of 3 microphones located on the campus of Oklahoma State University, the team managed to localize a rain-wrapped tornado storm by analyzing the infrasound produced by the event [3].

On May 11, 2017 a tornado in Perkins, Oklahoma (15 miles south of Stillwater) touched down. The formation of the tornado provided two periods of interest through the recording of the microphones. The analysis of the recording on the 3 microphones, and the code to determine existence of active tornadoes by measure of infrasound, determined two tornadoes to be present during the event. While the second tornado was believed to exist by Elbing's team, the National Weather Service neither confirmed nor denied the existence of the second tornado. The claim made by Elbing's team is backed by satisfying the four criteria necessary to locate an event; characteristic signature, coherence, concomitancy and bearing angle [4].

The goal of this research is to both encourage development of localization techniques for the team, as well as provide logical backing of the previously stated methods by the team. The equipment is the same 3 microphone array used by Elbing et al. (2018), however rather than waiting on naturally occurring events, the focus of the following experiment is to generate sound by a subwoofer and determine the location of the subwoofer [3].

2 Objectives

Develop an algorithm to localize an acoustic source through the infrasound array established on the campus of Oklahoma State University. Satisfy the localization by observation of behaviors of the signal by methods of characteristic signature, concomitancy, coherence and location of the source.

3 Experimental Methods

3.1 Test Facility and Model

An arrangement of 3 microphones will be used to record the data for the Acoustic Localization project. The microphones have previously been setup and remain in position with the ability to record data at the Design and Manufacturing Lab (DML) on the campus of Oklahoma State University (OSU). However, prior to performing the experiment, the data acquisition methods need to be verified based on real time recordings, and any maintenance needs to be performed. In addition to making sense of the live data, a baseline needs to be established in terms of what the spectra looks like on a typical day. Ideally, the tests performed would be done on a day that has little activity across the frequency ranges being recorded. It is known that construction, high winds, and potential storms can lead to an increase in noise for the array, thus mitigating these known variables can improve results. Ways to achieve this would be to perform the tests outside of the normal business hours during the work week and on clear days with little to no potential for storms.

The objective of the experiment is to emit sound from a known source in a location which is heard by the continuously recording microphone array. To successfully achieve this, there are four criteria which must be achieved: characteristic signature of the source, coherence, concomitancy, and directionality. Experiments will be aimed at verifying each of the four criteria separately.

Experiments will be performed outside of the DML within 50 yards of the 2 microphones of the array that are located at surface level (Microphone 3 is on roof and will not be accessible). A subwoofer with an extension cord will be placed at a measured distance away from the microphones and signal transmitted through the subwoofer via a MATLAB function (provided in the appendix). The four criteria will be tested using the setup with variations in distance, angle, and intensity of the signal being generated. Potential tests could include multiple sources being generated at the same time and movement of one of the sources in relation to the microphones.

3.2 Procedure

3.2.1 Concomitancy

Concomitancy is the appearance of a signal during the time a source is being transmitted, and the absence results when the source is removed.

1. Using the south-eastern microphone, place the subwoofer 10 meters away with the speaker facing the microphone.

2. Record the time (UTC) that tests are being performed to ensure the correct data is being pulled from the server. Record each time the signal source frequency is changed during this test.

3. Begin transmitting at a frequency of 33 Hz for 3 minutes, then turn off the signal for a duration of 2 minutes. Repeat this process in 3-minute segments for a duration of 20 minutes (square wave).

3.2.2 Characteristic Signature

Characteristic signature is the verification of an amplified signal produced being in direct correlation with the frequency received by the microphones. The equipment will be validated by increasing the intensity of the source generator in addition to verifying the rise in magnitude on the spectra shown in the computer stand.

1. Using the south-eastern microphone, place the subwoofer 10 feet away with the speaker facing the microphone.

2. Record the coordinated universal time (UTC) that tests are being performed to ensure the correct data is being pulled from the server.

3. With a tone generator, transmit a known signal for a duration of 5 minutes at each frequency. Record the time (UTC) that each tone is being transmitted.

4. Using the acquired data, verify that the recorded frequency is accurate with what was being transmitted. This will prove that the signal generator can produce the frequency it claims.

5. Upon completion of testing, turn off the signal generator and wait 20 minutes before proceeding to further tests. (Data is compiled in 20 minute segments; this will ensure that the data is not overlapping)

6. Starting at 35 Hz, transmit the signal in increments of Δf for a duration of 5 minutes each.
7. Using the acquired data, verify that the recorded frequency is accurate with what was being transmitted.
8. Upon completion of testing, turn off the signal generator and wait 20 minutes before proceeding to the next tests.
9. Vary the distances from the microphones while keeping the same frequency and volume to identify how the magnitude is changed.

3.2.3 Coherence

Coherence refers to the retrieval of a signal on multiple sources within the same time frame where the data received on both sources agree with one another. During previous tests, the subwoofer was located at close range to the southeastern microphone, and the location relative to the array was not specified.

1. Position the source at a distance (d) away from microphones 1 and 2, so that the sound waves produced will reach these microphones before microphone 3.
2. Record the time (UTC) that tests are being performed to ensure the correct data is being pulled from the server.
3. Begin transmitting at a frequency of 33 Hz for a duration of 10 minutes and record the starting and stopping time.
4. Wait 10 minutes before proceeding to the next test.
5. Reposition the source at a distance (d) away from microphones 2 and 3, so that the sound waves produced will reach these microphones before microphone 1.
6. Begin transmitting at a frequency of 33 Hz for a duration of 10 minutes and record the starting and stopping time.

3.2.4 Directionality

Directionality is the calculated bearing angle of a source which identifies where the source is originating from. This can be achieved from both a stationary source and a moving source. First, the tests will hope to verify the directionality of a source at a fixed location. If successful, the source will be put in motion and tested to see if it can be tracked as the source of sound travels.

1. Position the subwoofer at d = 50 feet away from the Southeastern microphone facing inwards towards the array.
2. Begin transmitting a known signal of 33 Hz at a recorded time (UTC) for a duration of 5 minutes.
3. Wait 20 minutes before proceeding to the next test.
4. Move the subwoofer to a new angle relative to the array and perform the same test as in step 2.
5. If data analysis proves that the bearing angle can be determined, perform the test moving the source between two points over a 5-minute period.
6. Wait 20 minutes before proceeding to the next test.
7. The final test for directionality, is to use the same frequency signal used in this portion of tests and vary the distance away from the array. Keep a record of the minimum and maximum distance the source is from the array, moving it in a straight line.

3.3 Instrumentation

3.3.1 Materials

The sound emitted by a subwoofer source will be recorded by the 3 microphones sensing sound in the local area. Each microphone has a direct current (DC) power supply and are model APS−1303, produced by Aktakom. The preferred sound of recording for data analysis will be the low frequency signal emitted by either one or two subwoofers. The microphone then outputs a digitized signal by means of a dynamic signal analyzer (USB−4432, National Instruments). The digitized signal produced by the analyzer is processed to the computer or workstation (Precision T3500, Dell) located indoors and secured at all times. The computer contains a data acquisition software package of Sound & Vibration Measurement Suite by National Instruments. An ideal sample rate is 1 kHz or 1000 samples per second. The data is collected and grouped into 20 minute observation windows, where it can then be processed.

3.3.2 Matlab Code

The tone produced by the frequency and the duration of the signal will be acquired by a tone generation program written in Matlab. The tone generation program will produce a set of frequencies. Additionally, the program will record the beginning and ending times of each tone in the coordinated universal time zone. All of this data will be output to a single spreadsheet which will be named in regard to the time of the first sound emitted. The data then needs to be retrieved from the computer workstation for analysis (6).

3.3.3 Subwoofer Limitations

It was notable that during experimentation the subwoofer had difficulty producing infrasound. On the fast Fourier transforms (FFT) the best infrasound frequency the subwoofer produced was 17 Hz which was the highest tested infrasound frequency. The lowest tested frequency, 4 Hz, was not noticeable on the FFT's at all. It is believed that this is due to the subwoofers size. Larger subwoofers are better at producing lower frequencies but even a significantly larger 15" subwoofer is designed with playing bass in music in mind which does not require the production of infrasound. Producing infrasound with the 10" subwoofer used is possible, but far from ideal and requires staying in the high end of the infrasound region or slightly above the infrasound region [1].

3.4 Test Conditions

The microphones are located outside and can go for periods of time without being touched. Ensure the microphones have no objects or overgrowth in front of the retrieval side. The local atmospheric pressure and temperature should be recorded as well as a short description of the wind conditions including speed, general direction, and gust speeds. Due to the experiment being subject to interference, record the local sound without the subwoofer generating an infrasonic frequency to record the infrasound which is already present in the area. Retrieval of data while train and airplane interference are present should be noted and regenerated if possible. This data should not be discarded as it could prove to be useful to later understand the appearance of data while being affected by a source of interference. Knowledge of data containing interference could prove beneficial when questioning possible outlying data later.

4 Results

The microphones record the pressure at any given point in time at a rate of 1000 samples per second. The data is grouped together by twenty-minute segments with a six second delay between each data file. Therefore, one data file will contain 1,200,000 pressure readings for each microphone. Because the analysis was divided between four different criteria, the data was processed different based on which condition was being tested. The code used to process all the data is provided in the appendix.

4.1 Concomitancy

The first condition tested was the concomitancy of the signal to verify that the microphone received the signal in the time a source was producing a tone. To satisfy this condition, a plot of the raw data (pressure versus time) could be used to show that when the signal is generated an influx in pressure is produced. As well, once the signal is removed, the pressure recorded returns to the level before the test. In this test, the source was placed within 15 feet of microphone 2 and a single frequency was generated for a duration of 3 minutes.

4.2 Characteristic Signature

To satisfy characteristic signature, the received signal had to be representative of the frequency at which the source was transmitting. Therefore, the data had to be plotted in the frequency domain.

4.3 Coherence

There is a coherence between the input and the output of a function. Often times these functions could be defined with differential equations and then solved. Many of these problems are taking into the Fourier domain to make solving easier and more accurate. This problem is the comparrison of one recieved signal to the other. The coherence begins with the collection of data in the time domain. The time domain is taken to the Fourier domain, where there is a ratio of the phase shifts over the real values of the function. The ratio can range anywhere from 0 to 1. For the value of coherence to be 1, the system would be perfect. Perfect is not always achievable, but the high sampling frequency and the reduction of potential aliasing is encouraged as well as the synchronization of the nodes to increase the value of coherence (6) [2].

4.4 Localization

The localization of a signal can be defined in many ways by many different researchers. The use of time distance of arrival is made difficult with the small subwoofer used in this experiment. A big pressure wave produced by a loud gunshot or explosion would do much better for making TDOA work. With that being said, one could also use the knowledge of the Fourier domain to help with what is being solved in the time domain. For example, if you know the closest microphone is one of the microphones then your code knows which way to solve for the location, this is not made possible if all of the microphones are not displaying the same results when there is not a signal input, or when there is a signal. One thing noticed during this experiment, which will be discussed, is the roof hears sounds louder than the microphones on the ground even though the roof is not closer than one of the microphones on the ground. The reason for this is the surrounding surface of the roof. There would need to be an extensive look taken at every microphone, and a constant decided on what should be filtering the incoming signal in the time domain [3].

5 Data Analysis

5.1 Concomitancy

Concomitancy is essentially confirming the microphones receive the emitted signal when the signal is emitted and does not receive the signal when it is not emitted. Figure 2 also shows that in the absence of a source, the ambient noise level for each microphone is different and introduces the need for data to be scaled. It is likely that the environment immediately surrounding the microphones has a profound impact on the received pressure waves. As stated, microphone 1 is located on the roof of the fabrication lab at OSU, while the other two microphones are fixed to a base within the ground. Likely, the hard surfaces of the roof and the absence of grass surrounding the microphone make the sound louder on the roof. By averaging the pressure of all 3 microphones over a period with no source, it was found that the pressure from microphone 2 needed to be multiplied by 5 and microphone 3 pressure multiplied by 4 to be equivalent to the conditions of microphone

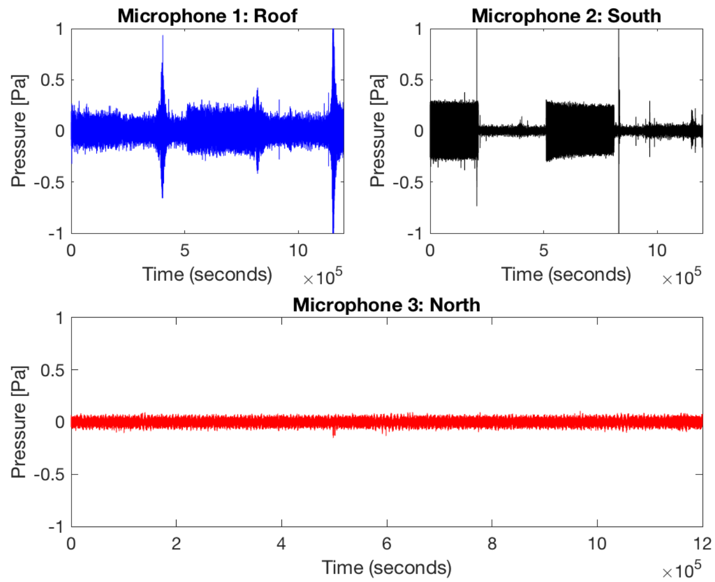


Figure 2: The raw data received by the microphones while the source only pointed at Microphone 1 and 2

1. Figure 3, which overlaps the pressures from the three microphones, gives a great illustration for the need to calibrate the data.

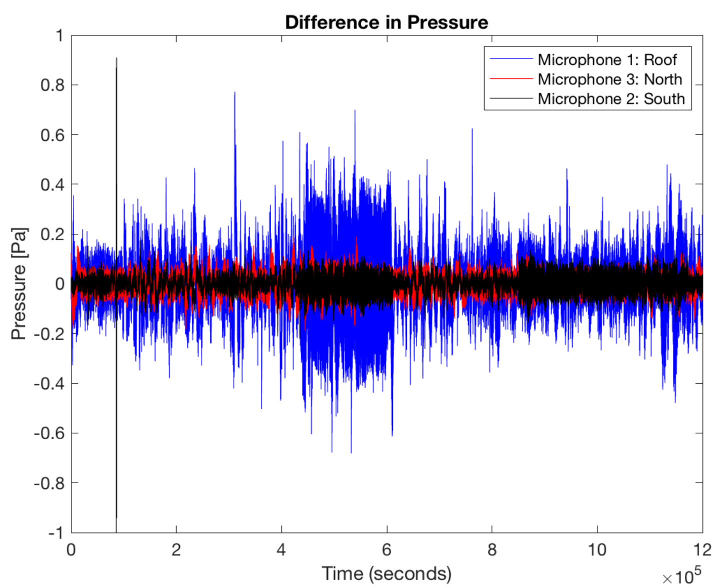


Figure 3: Depiction of the raw data plotted over one another for all three plots

5.2 Characteristic Signature

When analyzing the data, a characteristic signature must be identified to verify that what is recorded matches the signal being produced. The characteristic signature should correlate with the supplied signal frequency being output by the tone generation program. To satisfy the characteristic signature requirement, the frequency of the signal being emitted is identified and found in the magnitude versus frequency plots. By identifying the highest peak of the plot and checking it against the emitted frequency you can verify that you have a valid characteristic signature. Each microphone should show the same characteristic signature; however it will vary in magnitude depending on where the signal is being transmitted. This magnitude variation is caused by the sound losing intensity as it travels over longer distances.

The following displays a comparison of in the Fourier domain of decibel magnitude plots for a test where a 75 hertz tone were played over time we can look at the graphs and find the characteristic signatures below. The roof microphone receives a higher decibel magnitude in the domain. The reason for the louder sounds is the hard roof and other machinery on top of the roof. Given a reduction in sound to make all of the plots equal while not receiving a signal, the sound source could be determined by knowing which microphone is closer because of a higher rise in decibels. The data right now does not comply with the rise because there is no constant multiplied to make the three microphones appear to receive the same signal when there is no input.

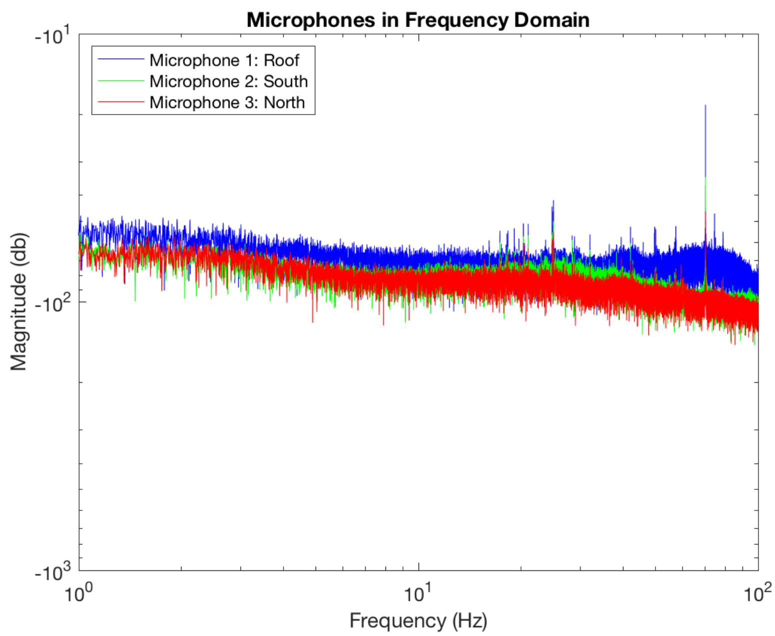


Figure 4: Fourier domain decibel magnitude plots [1].

Using Matlab code listed in the appendix (6), a tone of 35 Hz was generated and played for a duration of 3 minutes then stopped. After 2 minutes, another tone of 42 Hz was generated for a period of 3 minutes then stopped. Finally, after two more minutes of silence a 49 Hz tone was played for a duration of 3 minutes. The source was 50 feet southwest of microphone 2 and pointed in the direction of the center of the array. Figure 3 shows the data for this test and the spikes in the three regions tested. This code would be good for future analysis of sound with the microphones. Pulses of sound waves would give good data for locating a source.

In figure Figure 4 there are distinct over lapping peaks at 25, 50 and 75 on microphone two. The peaks at 25 and 50 are present because We then separate the plots, zoom in and clip the bottom part of the data to remove the ambient noise from the environment to allow us to focus on the characteristic signature in figure Figure 5 [1].

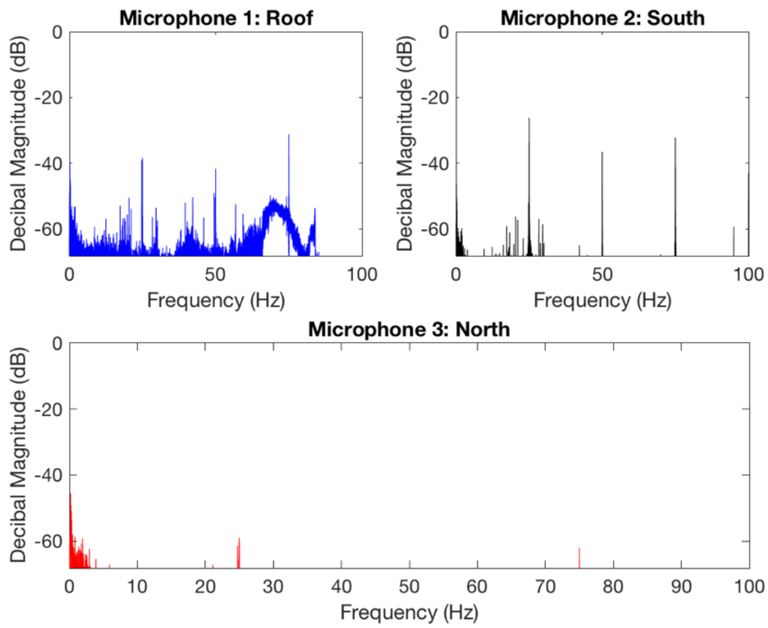


Figure 5: The clipped frequency domain to better determine the received frequencies [1].

5.3 Coherence

Coherence is the retrieval of a signal on multiple microphones within the same period where the data for multiple mics are all characteristic to the signal generated. This method can help delineate between the generated signal and background noise. In the established array, microphones are continuously collecting data at a rate of 1000 samples per second. There is background noise that cannot be controlled when testing, such as wind, construction equipment, and buried electric cables. A plot without filtering out a 60 Hz signal from electric lines on microphone 3 is displayed in the appendix in Figure 12. Figure 6 shows the magnitude plots of the array and a definitive increase in the 70 Hz region for the 3 microphones because a signal of 70 Hz was being output from the source. The plot could be clipped and the peaks closer examined by the method displayed in Figure 5. By comparing the magnitudes of the peaks the relative distance from the sources can be determined to assist in locating the source. The Matlab code for this method is located in the appendix (6).

$$C_{xy} = \frac{|\Psi_x(f)\Psi_y(f)|^2}{\sigma_x^2\sigma_y^2} \tag{1}$$

Coherence can also be used to help localize the source based on which microphones show the greatest coherence. Figure 7 shows the experimental setup, where the source was orientated in a way that its produced signal would be recorded by microphones 1 and 2 first. As a signal travels through space, the magnitude of the sound decreases. Which means that for the microphone furthest away from the source, the received signal is expected to be weaker in magnitude. In Figure 6, the data shows the microphone on the roof receiving a louder signal at 70 Hz, although microphone 2 is closer. The method with the use of standard deviations will help delineate these errors. The reason for the need to solve a coherence of this method to determine

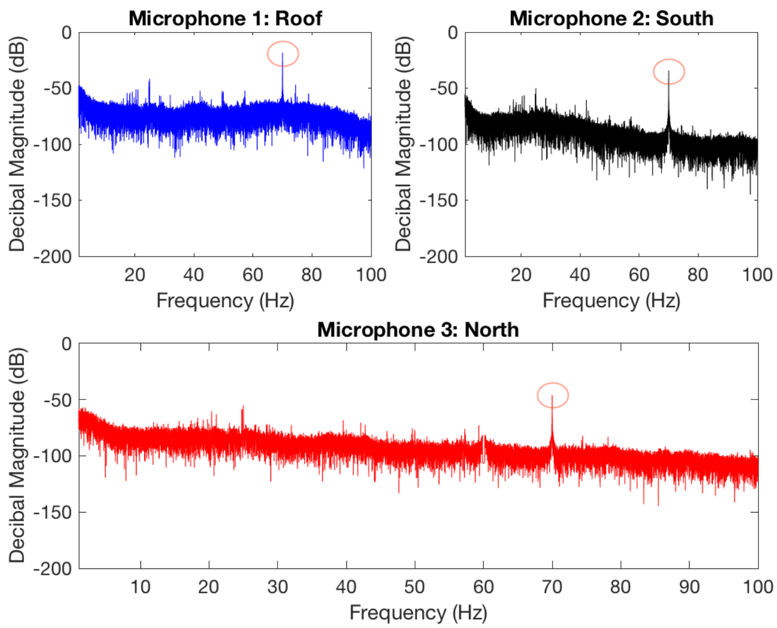


Figure 6: Frequency domain [1].

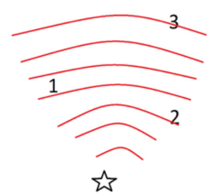


Figure 7: The experimental setup, where the source transmits a 33 Hz signal. The transmitted signal will first be received by microphones 1 and 2.

the relative changes in decibels between the microphones is because of the different material surrounding the microphones. The microphone on the roof is receiving a louder signal, which would mean the source is closer to that point except the source was known to be further away from the roof than the microphone 2. Without addressing these issues, the solver will not have an ability to determine a general direction before solving TDOA.

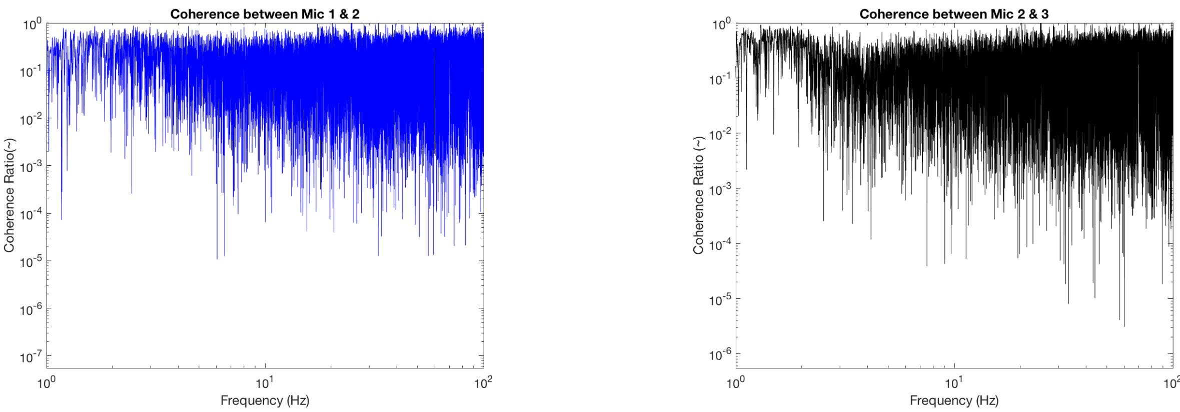
To confirm the results, the test source was moved to transmit a signal that would first reach microphones 2 and 3. Thus, it is expected that the coherence of the 33 Hz signal will be strongest between the second and third microphones. Figure 12 shows the figures for this test and while it is hard to see the difference of the magnitudes for 33 Hz, the magnitude of the coherence for mics 2 & 3 is greatest.

This idea of using coherence plots to determine which nodes are receiving the greatest magnitude is crucial for locating a location of an unknown source with an unknown time sent. There may be an assumption that simply using the Fourier domain plots and this idea of coherency will solely provide a method for solving the location of an unknown source. However, the frequency domain only helps determine the order in which the TDOA will be solved and which differences between the triangles should be considered. The TDOA method for locating a source will be discussed in section 5.4.2.

There is also a coherence method which relates an input and an output. In this case, the input is the retrieval of pressure from one microphone to the retrieval of the other. The equation is as follows.

$$C_{xy}(f) = \frac{|P_{xy}(f)^2|}{P_{xx}P_{yy}(f)} \tag{2}$$

This equation is known as the magnitude squared coherence, and is performed by Matlab. The closer to a



(a) Coherence between Mics 1 and 2

(b) Coherence between Mics 2 and 3

Figure 8: Magnitude-squared coherence

value of 1 the magnitude squared coherence becomes, the more ideal the input and output of the system. Figure 8 displays two of the magnitude squared coherence plots, while the last plot is located in the appendix (Figure 13).

5.4 Directionality

5.4.1 Location of Nodes

The 2 dimensional spacing of the microphone reciever nodes are given. The direct spacing in between the nodes, as depicted in Figure 9b, is the information of the location of the triangle of receiver nodes the group was provided with [3]. The coordinates in relation to one another are discovered by application of the Law of Cosines [7].

$$\frac{cos(\alpha)}{a} = \frac{cos(\beta)}{b} = \frac{cos(\gamma)}{c} \tag{3}$$

$$c = acos(\beta) + bcos(\alpha) \tag{4}$$

The points relative to one another are then able to be plotted, choosing a point in the positive [x, y] domain to be plotted.

5.4.2 Location of Source

The terrain of the testing provides for 3 dimensions of coordinates, and therefore should be more than three microphones because there are two solutions to the intersection of three spheres. The problem has been

(a) Law of Cosines Triangle [7] (b) Location of Microphones [3]

Figure 9: Triangle of microphone array

simplified to two dimensions due to complexity of the problem. The idea is to make a 2 dimensional solver work, and then begin considering the third dimension to generate more accurate solving points. The points are plotted on a two dimensional grid as shown in Figure 10.

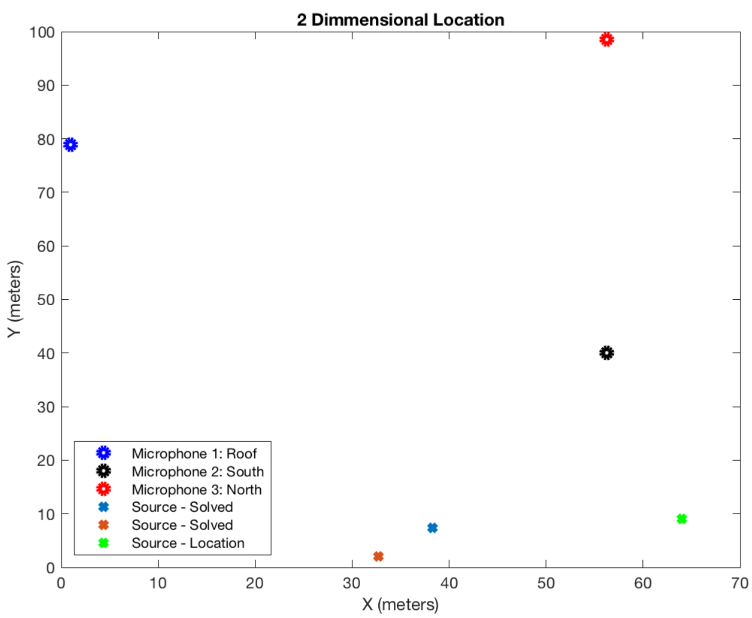


Figure 10: Location of microphones, source and sources solved for.

The location is found by time distance of arrival (TDOA) [5]. For this problem, a 3 dimensional code would be more feasible. Given an arrangement of four microphones, two separate triangles could be used in determining differences in time of arrival between four separate nodes. Although, there are only 3 microphones to work with, so the following 2 dimensional TDOA equations are used. The time difference of arrival between each node is used because the original time of the source sent is unknown [5, 6].

$$t_{23} = t_3 - t_2 \tag{5}$$
$$t_{21} = t_1 - t_2 \tag{6}$$
$$t_{31} = t_1 - t_3 \tag{7}$$
$$\tag{8}$$

Where the triangle inequality is known to be the following [6].

$$t_{31} \leq t_{23} + t_{21} \tag{9}$$

The difference of time received between the nodes are found by the pressure waves put out by the source. The distance between each node is known, so the approximate time of travel can be found by relating the speed of sound (c) and the distance between the node. The code will need to step through the recieved signal and plot the sound source, this could be done a number of ways. For now the time domain graphs are looked at to see where the first pressure wave of signal hits the closest micropone. The sample rate is 1000 Hz, so there will no be much of a difference on the 20 minute time domain plot before the pressure wave hits the next nodes. This is where the relationship of position to one another comes into play. The code looks for the peaks in the range of what appears to be the pressure wave hitting the first microphone, the code then spaces itself by a range discovered by the distance between the node and the speed of sound. The placement of the microphone is limited for these testings due to the buildings, so not much time was put into solving for more than one location other than where it is at. The code will need additional logic to solve a 360 dimensional TDOA of the system. The following equations are how the location of the source is found [6].

$$t_{23} = \frac{1}{c}(\sqrt{(x-x_3)^2 + (y-y_3)^2} - \sqrt{(x^2-y^2)} \tag{10}$$
$$t_{21} = \frac{1}{c}(\sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x^2-y^2)} \tag{11}$$

The [x, y] coordinates in the equation are the location of the source. The location of the source can then be minimized to a point. As displayed in Figure 10, there are multiple solutions to this problem. As microphones are increased, and the code and equations begins to solve for domain which applies to the real world problem of received signal, there will be less solutions to the problem and a more accurate location of the source [6].

5.4.3 Localization Moving Forward

For the location of the source to be successful, the magnitude of the source needs to continue to drop with distance. For the testing done during this experiment, due to the differences in surrounding material, the magnitude in decibels did not appear to drop with distance. The following equation is known for an ideal world [6].

$$Intensity \approx \frac{1}{r^2} \tag{12}$$

Where every time the signal doubles in distance, there is a drop of 3 decibels. An ideal world is not necessary for the time difference of arrival technique, but a drop in decibel per distance traveled is. The coherence C_{xy} method where the ratio of magnitudes at each frequency over the standard deviations can equalize the differences caused by the local density of the material. The differences between each node of interest will need a coherence solved before the code decides which general direction to solve for the source in. This technique could provide for a TDOA technique which could solve for a location of an unknown source and original time sent of source for a location in any direction.

6 Conclusion

The goal of the experiments was to develop an algorithm to localize an acoustic source through the infrasound array established on campus of Oklahoma State University. The team did not have the funding for a source with the capability of consistently transmitting acoustics in the infrasound region, thus the localization was done using tones above the infrasound region. Through analysis of the data plotted in the time domain, a hand method is developed. In conclusion, the coherence method for differences between nodes is necessary to remove the hand work from the problem.

Acknowledgments

The collection and application of knowledge in this report was made possible with the help of Dr. Omer San, Dr. Alan Noell, Dr. Joe Conner, Dr. Ronald Delahoussaye and Dr. Rick Gaeta.

References

[1] Cook, J.C. and Weber, G.O. (2018) *Measurements and Instruments Laboratory 6: Fourier Transforms* Oklahoma State University, https://github.com/jchriscook/AcousticAlgorithm.

[2] Drongelen, W.V. (2015). *Lecture 7: LTI Systems, Convolution, Correlation, and Coherence, Dr. Wim van Drongelen* University of Chicago, https://www.youtube.com/watch?v=2zOcFl39NP4.

[3] Elbing, B. R., Petrin, C. E. and Van Den Broeke, M. S. (2018). *Detection and characterization of infrasound from a tornado* arXiv:1809.00038 [physics.ao-ph] (https://arxiv.org/abs/1809.00038).

[4] Elbing, B. R. and Gaeta, R. J. (2016). *Integration of infrasonic sensing with UAS (invited).* The 8^{th} Atmospheric and Space Environments Conference, AIAA Aviation, AIAA2016-3581, Washington, DC (June 13-17)(doi:10.2514/6.2016-3581).

[5] Gustafsson, F., and F. Gunnarsson. *Positioning Using Time-Difference of Arrival Measurements.* 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., doi:10.1109/icassp.2003.1201741.

[6] Reddy, P.G. *Efficient Time of Arrival Calculation for Acoustic Source Localization Using Wireless Sensor Networks.* Cleveland State University, 2011.

[7] Wikipedia - The Free Encyclopedia *Law of cosines* https://en.wikipedia.org/wiki/Law_of_cosines#/media/File:Triangle-with-cosines.svg

Appendix - A

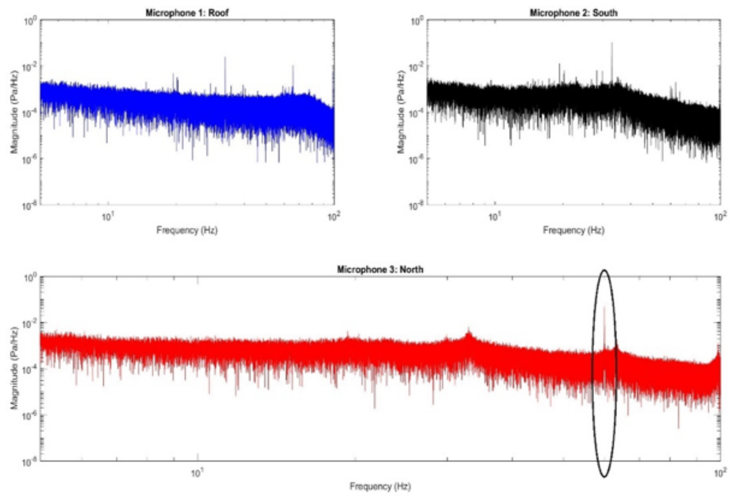


Figure 11: Shows the 60Hz signal characteristic of a buried electricity cable near the location of Microphone 3 before filtering it out.

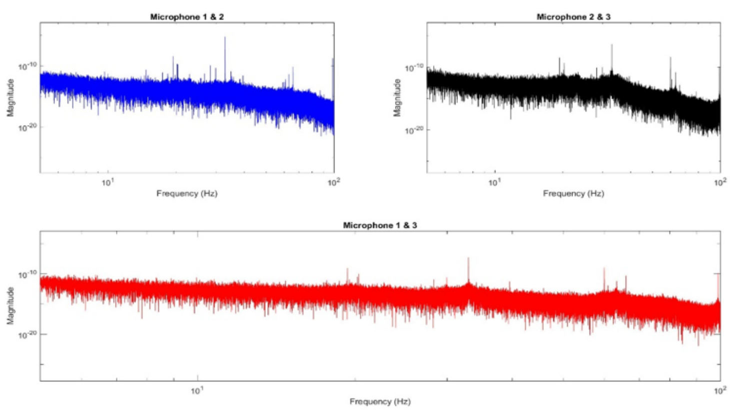


Figure 12: Shows the coherence between microphones when the source is located closest to microphones 1 and 2.

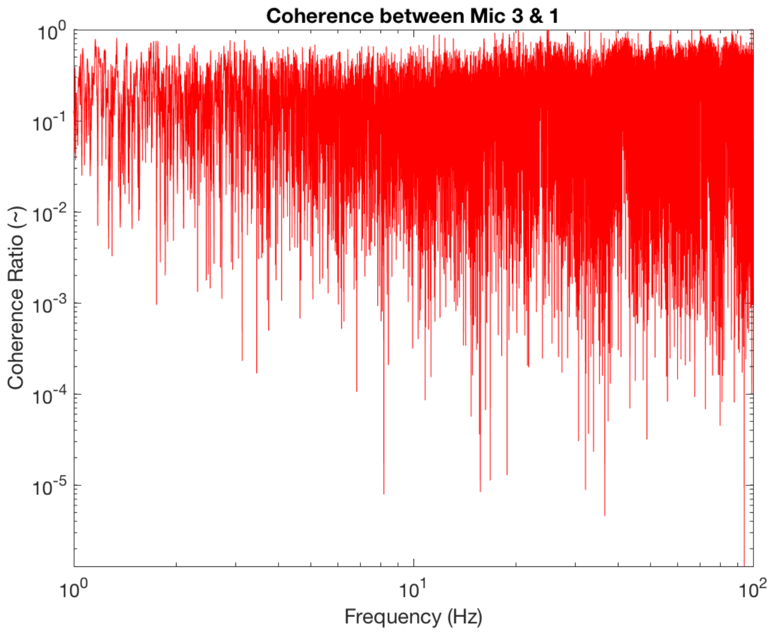


Figure 13: Coherence between microphones 3 and 1.

Appendix - B

Appendix for Matlab code used during the project and referenced in the paper.

Matlab Code

Sound Wave Propagation

```
%Tone Generator
%Christian L Negratti
%Oklahoma State Mechanical Engineering
%November 2018
%This code is used to produce square wave tones
%This code records the starting and ending times of each tone in UTC

%alter these variables before testing
f = 35:7:70;    % frequency range tested - first:step:last
dt =  1;        % seconds - length of each tone
t = 1;           % seconds - pause time between tones

%Formats and makes colum titles in the output variable
clear y                              %clear any values currently in the output variable
y(1,1) = "Frequency (Hz)";       %Formatting output variable headers
y(1,2) = 'Start Time (UTC)';
y(1,3) = 'End Time (UTC)';

x = 1;  % Counter
while x < length(f) + 1

    y(x+1, 2) = string(datetime('now', 'TimeZone', 'UTC')); %print start time
    y(x+1, 1) = string(f(x));   %adds frequency to output
```

```

    fs = 8192 ;                                       % samp. frequency - do not change
    values = 0 : 1/fs : dt;                           % time point values
    amp = 5;                                           % amplitude
    w = amp * square(2 * pi * f(x) * values);          % produces square wave
%     w = sin(2 * pi * f(x) * values);          % produces sin wave
    sound(w)                                          % produces the tone

    pause(dt)                                         % program does not wait for tone
    y(x+1, 3) =datetime('now', 'TimeZone', 'UTC'); % print end time
    pause(t)                                          % pause between tones
    x = x + 1;                                        % Next Step in the cycle
end

%disp(y)     %optional, uncomment to display output

filename = string(y(2,2)) + '.mat'; %create file name that data will be exported to

display('Output table is saved as ' + filename)

save(filename, 'y') %saving freqs and times to .mat file for future reference
```

Coherence Method

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Coherence%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            Coherence12 = (amplitude1(1:n).*amplitude2(1:n)).^2 / (SD1^2 * SD2^2);
            Coherence23 = (amplitude2(1:n).*amplitude3(1:n)).^2 / (SD2^2 * SD3^2);
            Coherence13 = (amplitude1(1:n).*amplitude3(1:n)).^2 / (SD1^2 * SD3^2);

            figure(5)
            %figure(7)
            subplot(2,2,1)
           %stem(freq,Coherence12(1:n), '-b');
            loglog(freq,Coherence12(1:n), '-b');
            xlim([5 100]);
            ylim([0 0.0012]);
            title('Microphone 1 & 2');
            xlabel('Frequency (Hz)');
            ylabel('Magnitude');
            %figure(8)
            subplot(2,2,2)
           %stem(freq,Coherence23(1:n), '-k');
            loglog(freq,Coherence23(1:n), '-k');
            xlim([5 100]);
            ylim([0 0.0012]);
            title('Microphone 2 & 3');
            xlabel('Frequency (Hz)');
            ylabel('Magnitude');
            %figure(9)
            subplot(2,2,[3, 4])
            %stem(freq,Coherence13(1:n), '-r');
            loglog(freq,Coherence13(1:n), '-r');
            xlim([5 100]);
            ylim([0 0.0012]);
            title('Microphone 1 & 3');
            xlabel('Frequency (Hz)');
            ylabel('Magnitude');
```

Plotting Class

Only pieces of the class used will be presented here. The code can be found at https://github.com/jchriscook/AcousticAlgorithm. The Matlab function mscohere code found at https://www.mathworks.com/help/signal/ref/mscohere.html.

```
%% PlotingClass.m
```

```
% a class def file which has to be the same name as the file
classdef PlottingClass
   properties
      % properties located on github

   end

   %% Fourier Domain - Cook and Weber
       function [] = Perform(obj)

      %% Coherence - Cook
           thisone2_2 = figure;
           set(thisone2_2, 'Visible', obj.figs);

           % * Mic 1 to Mic 2
           thistitle = 'Coherence between Mic 1 & 2';
           [cxy12, f12] = mscohere(obj.R1, obj.R2, [], [], [], 1000);
           obj.dbplots(f12, cxy12, '-b', 0, 1, 0, 100, ...
               thistitle, 'Frequency (Hz)', 'Coherence Ratio(~)')
           thisplot = ' - cxy12';
           obj.Putintofiles(thisone2_2, thisplot);

    ...

         end
         %% Save the figures
       function [] = Putintofiles(obj, thisone, thisplot)
          type = '.png';
          tog = strcat(obj.newname, thisplot, type);
          slash = '/';
          loc = strcat(obj.imagedir, 'Images', slash, tog);
          saveas(thisone, loc);
          pause(obj.figs)
          close(thisone);
        end
```