Direction and Ranging of an Incoming Sound from an Arduino Sound Sensor System

Authors: Luke Mantle & Mauricio Soroco

Mentor: Dr. James Charbonneau

Science One

2019-2020

Abstract

Sound detection and ranging has many applications related to echolocation, navigation, and geolocation. All rely on accurately pinpointing the location of the source with the use of sound delays. For this study, we assembled a device that could pinpoint the location of a sound source with the use of differences in times of arrival of its sound. It consisted of three sound sensors connected to an Arduino circuit board, which sent data to be processed in Python. The device was found to function correctly to some accuracy. We found that the magnitude of its error varied with the relative location of the sound source and sensors. This led to the production of a model for the error that lets the machine generate a two-dimensional probability distribution as a heat map for the location of the sound source by combining any single measurement with the experimentally determined uncertainty in the equipment. The probability distribution generally matched the machine's actual distribution of outputs.

Introduction

Sound detection and ranging have many applications that include both civil and military uses [4, 6, 7, 9]. The ability to accurately pinpoint the location of a wave source via time delay measurements is shared among many of these [7]. In the field of astronomy, major advances emerge from the capabilities of the measuring equipment to detect emissions from celestial objects accurately [5, 10, 11]. Radio astronomy studies celestial objects by detecting their radio emissions using large radio telescopes [3, 10]. However, the picture resolution achieved by an individual telescope is limited by several factors including the size of the telescope's antenna aperture [2, 3, 10]. To overcome this limitation, a type of interferometry known as Very-long-baseline-interferometry can be employed, in which many distant telescopes are synchronized together to emulate a larger singular telescope [2, 10]. To accomplish this, the delays between the times of arrival of the radio signals at different telescopes are combined and processed [1, 10]. This study employed a similar technique known as time difference of arrival (TDOA) to build an apparatus that could pinpoint the location of a sound source, described later in the *Theory* section. This approach is a widely used in location-finding systems, as outlined in [8]. The apparatus involves three Arduino sound sensors synchronized together through an Arduino circuit board. We hypothesised that three sound sensors, if sufficiently distant and without interference, can measure the range and direction to a sound source.

Methods and Apparatus

Three sound sensors were connected to an Arduino circuit board (Figure 2). A program developed in the Arduino Integrated Development Environment (IDE) read each of the sound sensors' outputs (HIGH or LOW) on each cycle of the program. It recorded the time, in microseconds, when the first HIGH from each sensor was received, representing the time of arrival (TOA) of the sound at that microphone. These three TOAs were then read and processed using a Python program.

Theory

Suppose the program is started at time $T = 0$. A sound is made at some unknown time T_0 after this. The time for the sound to travel to each of the sensors is t_n (n = 1, 2, 3… for each of the sensors). The TOAs of the sound T_n measured from when the program is started to when the sound reaches each sensor (i.e. the times recorded by the Arduino program) will be equal to the times of travel t_n plus T_0:

$$T_n = t_n + T_0. \tag{1}$$

The difference between times of travel D_{mn} (also known as the TDOA) given as a function of the times of travel t_m and t_n (where m and n correspond to any two sensors) is then:

$$D_{mn} = t_m - t_n. \tag{2}$$

Although only T_n is measured (and not t_n), D_{mn} can still be calculated by taking the difference between T_m and T_n:

$$D_{mn} = T_m - T_n = (t_m + T_0) - (t_n + T_0) = t_m - t_n. \tag{3}$$

Therefore, by measuring the TOAs T_m and T_n, the difference in travel times D_{mn} can be determined. The travel times themselves, and therefore the path distances, however, cannot be determined, because the unknown T_0 does not cancel without performing the subtraction in equation (3). As such, the subsequent equations must be used to determine the source's location, based only on differences in travel times, D_{nm}.

By multiplying the delay

$$D_{mn} = t_m - t_n \tag{4}$$

by the speed of sound, we obtain the difference in path distances s_{mn}:

$$s_{mn} = s_m - s_n. \tag{5}$$

According to the Pythagorean Theorem, the path distance from the source at (x_0, y_0) to any sensor at (x_n, y_n) is

$$\sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} = s_n. \tag{6}$$

It follows that, as shown in Figure 1, the path difference, s_{mn}, to any two sensors will be related to these sensors' positions and the position of the source by:

$$\sqrt{(x_0 - x_m)^2 + (y_0 - y_m)^2} - \sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} = s_{mn}. \tag{7}$$

So, given any x_m, y_m, x_n, y_n, and s_{mn}, one can create a parametric equation $f(x, y) = s_{mn}$ that represents all the possible points where the source could be located, consistent with the path difference s_{mn} to those sound sensors.

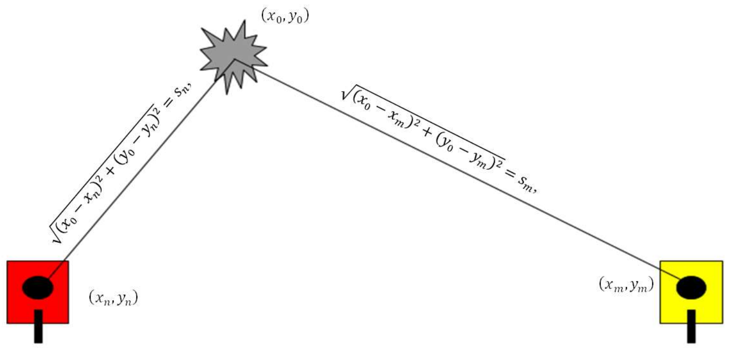
(x_0, y_0)
$\sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} = s_n,$
$\sqrt{(x_0 - x_m)^2 + (y_0 - y_m)^2} = s_m,$
(x_n, y_n)
(x_m, y_m)

***Figure 1**: distances from two sound sensors to sound source.*

Model

We created a mathematical model in the dynamic graphing calculator, Desmos (*Appendix*) for the measurements and calculations performed by the apparatus. Based on the coordinates of three sound sensors (x_1, y_1), (x_2, y_2), and (x_3, y_3) and of a source (x_0, y_0), the program simulates the TDOAs between each pair of sensors and draws the parametric equation (7) for each pair. The delay multiplied by the speed of sound must be used to obtain a distance. However, for simplicity in the Desmos model, the speed of sound was set to unity in both the simulation and signal analysis.

The Desmos model was used to give a first impression of how the apparatus might function in theory, and to identify the best distribution of sound sensors to give the least error and fewest false solutions (*Discussion: Sensitivity in sound source locations*). On this basis, we chose to distribute the sensors in an equilateral triangle (Table I). The machine measured the TOA at three sensors, T_1, T_2, and T_3. Based on equations (2)-(5), it determined the path length differences between each of the three sensor pairs: s_{12}, s_{13}, and s_{23}. Then, it created three curves defined by the parametric function in equation (7). The point where these three curves intersect is the source's location. The Python program also produced a graphical "heatmap" representing the error in the location estimate generated by the apparatus. The error was determined by analyzing all the experimental trials (*Discussion: Error propagation in the heat map*).

Apparatus

Six sound source locations were selected, for which the apparatus ran repeatedly (with roughly 40 trials per location). Our procedure involved starting the program and clapping once at the given location. The program saved the locations of the sensors, the measured location of the source, and the TOAs to a file for later analysis.

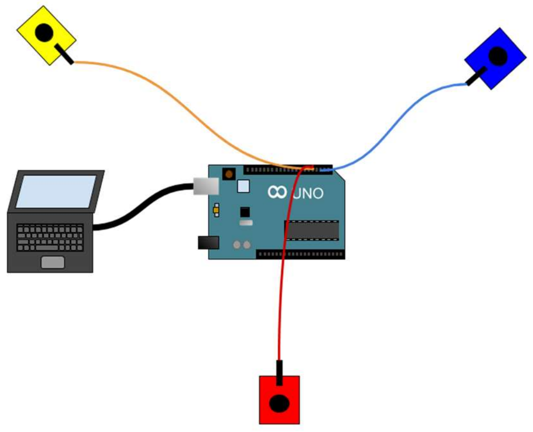

Figure 2*: Setup with 3 sound sensors connected to an Arduino circuit board that sends data to a computer for analysis (figure not to scale)*

The analysis was based on the above theory (*Theory*). We assumed that the sound travelled at a constant speed, with no reflection or refraction or obstruction by any of the equipment (such as the wires, tabletop, laptop etc.). Furthermore, we assumed that the sound can be modelled as emanating from a point source.

Results I – Preliminary measurements

Table I

Positions of sound sensors (m)	sensor 1	0, -0.415
	sensor 2	0.455, 0.420
	sensor 3	-0.455, 0.420

Positions measured by ruler

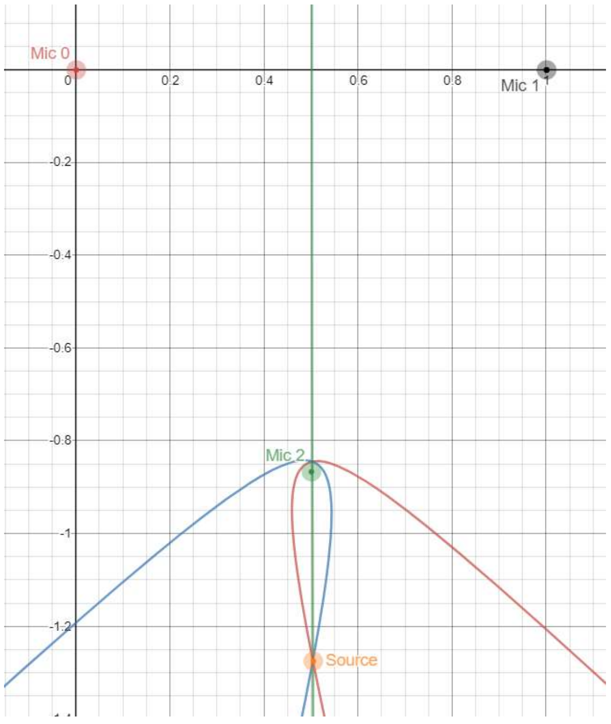

Figure 3*: Desmos simulation with sound source at a location resulting in two solutions. Notice there are two points of intersection: one where the source is and one just above the bottom sound sensor. (Due to limitations of Desmos, this figure does not adhere to the color scheme used in the subsequent figures.)*

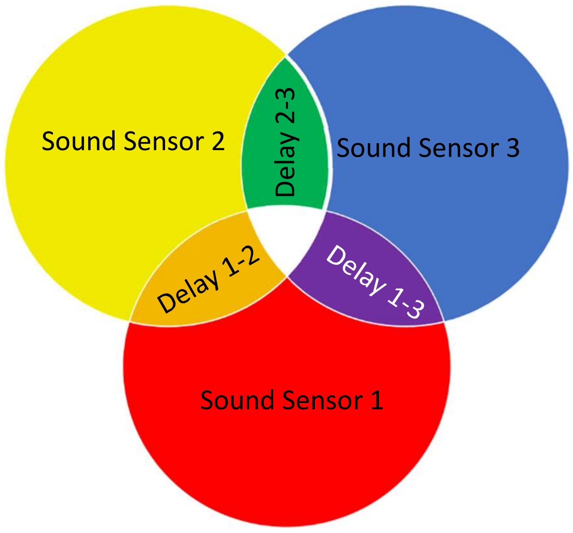

Figure 4: *The colours of the sound sensors and the parabolic representation of the delays.*

For clarity, the apparatus outputs have been colour-coded following Figure 4. Black dots indicate the actual ruler-measured position of the source (handclap). The uncertainty in the position of the hands was approximately 0.12 m (due to the size of the hands). The area of the black dot represents this uncertainty. The black dots were plotted manually on top of the apparatus measurement. Their locations weren't involved in calculations of the apparatus.

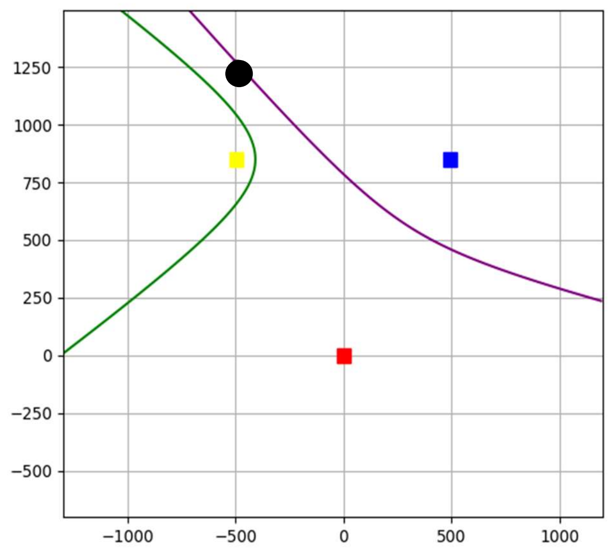

Figure 5: *Graph produced by a single measurement from the apparatus. One of the curves (orange) is absent because error in measurement resulted in the delay between its two sensors being unphysical. Colour scheme as per Figure 4.*

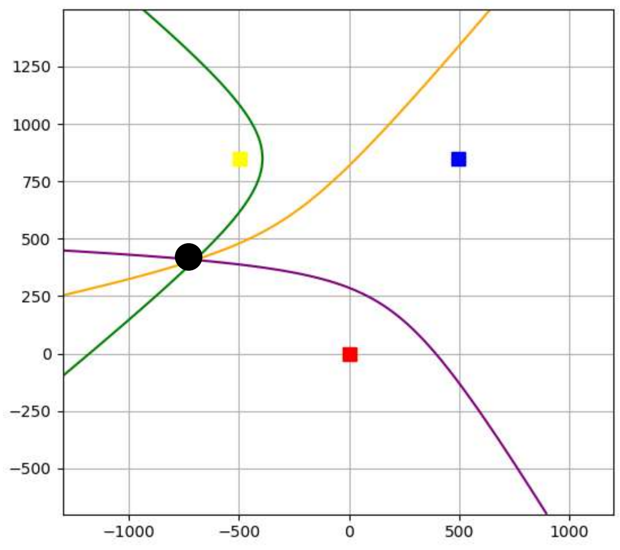

Figure 6: *Graph produced by a single measurement from the apparatus, yielding a single intersection point. Colour scheme as per Figure 4.*

Discussion I

The measurement apparatus could successfully output a graphical interpretation of the position of a sound source and save the data.

Sensitivity in source locations

The apparatus was sensitive to certain source locations in two main scenarios. Firstly, if the source was directly inline with the centre of the equilateral triangle and the closest sensor (Figure 3), the apparatus would find two solutions for the source's location; the first some distance away from the sensor, and the second much closer on the other side. Secondly, if the source was in line with two sensors, the resulting TDOA was close to the limiting case for a real-valued curve.

Consequently, small measurement errors left one of the curves absent (Figure 5). In these cases, without a point of intersection between the three curves, there was no immediately obvious solution for the source's location, leaving only a general idea of its direction and range. However, the heat map (*Discussion II*) still provided an estimate for the source location. Both sensitive situations were consistent with the Desmos model.

Sensitivity in Sound Type

The apparatus could only function properly on sharp sounds as it relies on an amplitude threshold detection (*Methods and Apparatus*). This is a limitation from the Arduino sound sensors rather than our analysis method. While testing the Arduino sound sensors, we found they consistently detected sharp sounds like snaps and claps but struggled with longer sounds such as constant notes played from a phone speaker. Notes of different wave types (rectangular, sinusoidal, triangular) generated from a smartphone application were not reliably detected. Consequently, for each measurement, a singular handclap approach was used. We found that the machine worked more reliably with certain clap types. Many factors might be associated with this result such as hand size and shape. Often, we found the larger and slightly damp (wet) hands produced better results. We experimented with an artificial clapping contraption involving two wooden wedges tied at one end with elastic bands. The wedges were pulled apart and released to make a sharp sound. This limited the variation of the sound generated and the movement of the sound source between trials (since we tied the wedges to a chair which remained fixed between trials). The position accuracy achieved with the fixed wedges is likely somewhat better than that achieved with freehand clapping. However, the intensity of the sound was higher using manual clapping. This resulted in more reliable detections.

Ultimately, we relied on handclaps to produce sound. Despite the variation in sound type caused by clapping, our apparatus functioned more consistently with them. We did not quantitatively measure the sensitivity in sound type.

Results II – Error Propagation

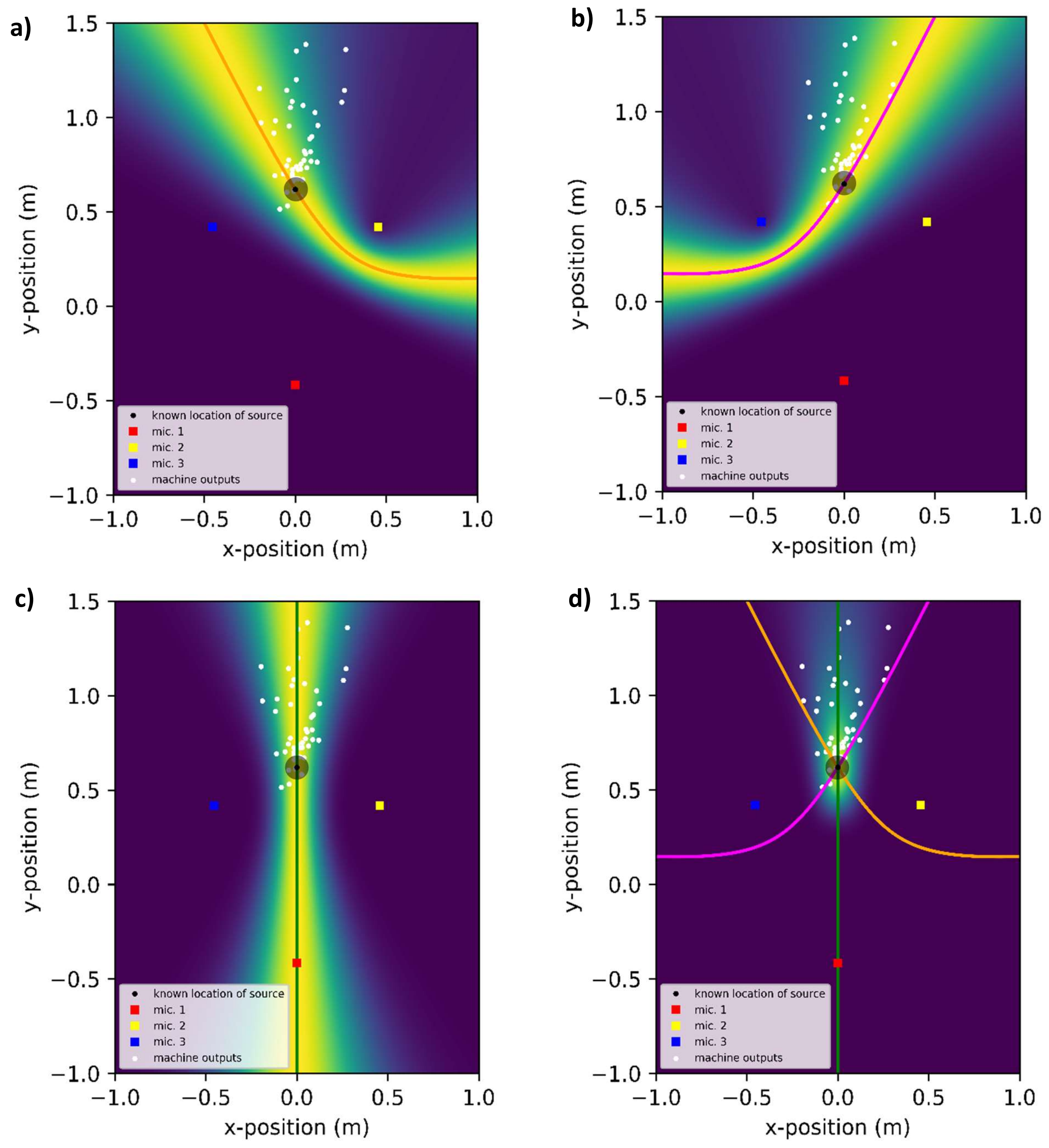

Figure 7: *a, b, c: graphs produced from a set of measurements (white) with the sound source at a fixed location (black). Each graph pertains to one pair of sound sensors. Each curve is a parametric function between the two sound sensors in question. Heatmap represents the probability distribution function associated with the curve: yellow is high, blue is low. Colour scheme as per Figure 4.*

Figure 7: *d: overall probability distribution function obtained by multiplying 7a, b, c.*

Table II

Average standard deviation of delays	Time ($u[D]$)	492.0 μs
	Distance ($u[s]$)	168.8 mm

Discussion II

Error propagation in the heat map

There is an uncertainty associated with each of the TOAs, T_n, measured by the sound sensors; however, we lacked the means to experimentally determine these uncertainties directly. There was no method of reliably producing a sound at a specific time, accurate to the order of microseconds, after the initiation of the Python program controlling our device. Nevertheless, the distribution in TDOAs between any two sensors can be determined from a set of trials with the source in the same location. We did this by analyzing raw data. Histograms such as Figure 8 (*Appendix*) were produced for sets of TDOAs from each pair of sensors for each source location. Built-in Python functions were used to find the standard deviations of all these distributions (one distribution for each pair of sensors for each source location). These standard deviations varied little from one pair of sensors to the next and from one set of samples to the next. Thus, their average was used as the uncertainty in any delay measured for a pair of sensors, $u[D]$ (Table II).

The uncertainty in a single delay results in an uncertainty in the exact position of the curve generated by the parametric function. This uncertainty was visualized by creating a two-dimensional probability distribution based on the distribution of the delay and the parametric function. An equation $N(x)$ for the normal distribution of the delay was formed, with the average being D_{avg}, and the standard deviation, $u[D]$. For each point in the two-dimensional space, the delay D' that would be recorded if the sound were truly at that point was calculated. The probability density for each point was defined as $N(D')$. Thus, one such two-dimensional probability density plot can be created for each pair of sensors (Figure 7 a, b, & c). To combine these three into one, we multiplied the three individual plots together, giving an overall probability distribution (Figure 7 d).

Conclusion

A working apparatus was developed with three synchronized Arduino sound sensors separated by approximately half a meter, that could determine the direction and range of an incoming sound source to some precision. Each position measurement was described by an intersection of parametric functions over a two-dimensional probability density plot. An ideal model of the apparatus was created in Desmos, with adjustable coordinates for the sound sensors and source.

Recording sound with microphones, instead of taking TOAs with sound detection sensors, would allow for more precise determination of TDOAs by matching the signals from one microphone to another. Such an apparatus would also be able to function over a wider variety of sound types, compared to our apparatus which instead required sharp sounds because of the way its sound sensors work via a threshold detection approach.

The addition of a fourth sensor would allow the apparatus to work in 3 dimensions, or more accurately in 2 dimensions (*Appendix*).

References

[1] Bhatti, J. A., Humphreys, T. E., & Ledvina, B. M. (2012). Development and Demonstration of a TDOA-Based GNSS Interference Signal Localization System. Paper presented at the *Position Location and Navigation (PLANS), IEEE Symposium,* 455-469.

[2] Cohen, M. H. (1973). Introduction to very-long-baseline interferometry. *Proceedings of the IEEE, 61*(9), 1192-1197.

[3] Counselman, C. C. (1973). Very-long-baseline interferometry techniques applied to problems of geodesy, geophysics, planetary science, astronomy, and general relativity. *Proceedings of the IEEE, 61*(9), 1225-1230.

[4] Gray, A. C., Anderton, M., Crane, C. D., & Schwartz, E. M. (5). Design, construction, and implementation of an inexpensive underwater passive SONAR. *Applied Acoustics, 148*, 251-263.

[5] Harwit, M. (2019). *Cosmic discovery: the search, scope, and heritage of astronomy*. Cambridge University Press.

[6] Ho, K., & Chan, Y. T. (1997). Geolocation of a known altitude object from TDOA and FDOA measurements. *IEEE Transactions on Aerospace and Electronic Systems, 33*(3), 770-783.

[7] Jin, B., Xu, X., & Zhang, T. (2018). Robust Time-Difference-of-Arrival (TDOA) Localization Using Weighted Least Squares with Cone Tangent Plane Constraint. *Sensors (Basel, Switzerland), 18*(3), 778. https://10.3390/s18030778

[8] O'Keefe, B. (2017). Finding Location with Time of Arrival and Time Difference of Arrival Techniques. *ECE Senior Capstone Project,*

[9] Smith, W. W., & Steffes, P. G. (3). Time delay techniques for satellite interference location system. *IEEE Transactions on Aerospace and Electronic Systems, 25*(2), 224-231.

[10] Thompson, A. R., Moran, J. M., & Swenson Jr, G. W. (2001). Interferometry and synthesis in radio astronomy: Wiley Interscience.

[11] Zheng, H., & Zhang, Y. (2008). Feature selection for high-dimensional data in astronomy. *Advances in Space Research, 41*(12), 1960-1964.

Appendix

Desmos Model

Link to mathematical model created in Desmos:

3 sensors:

https://www.desmos.com/calculator/m6atywitb6

4 sensors:

https://www.desmos.com/calculator/fpyxvlumpw

Figures

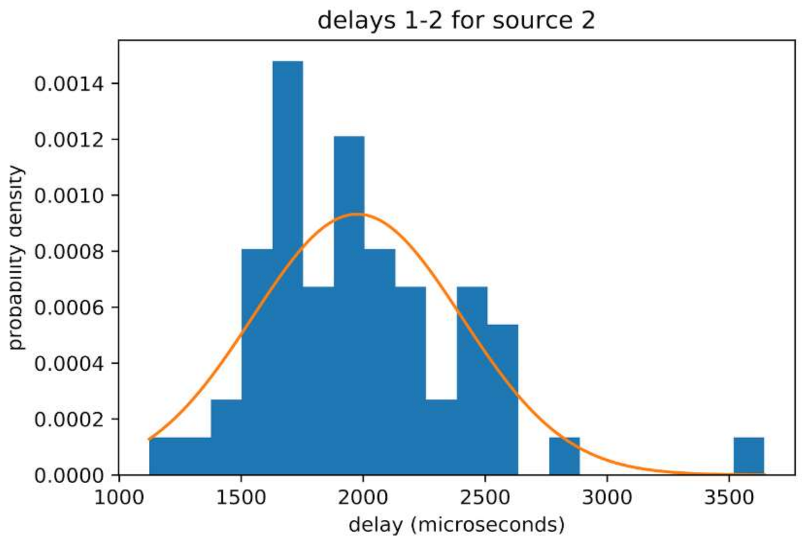

Figure 8: *An example of a distribution of delays (blue) between times of arrival at sensors 1 and 2, for the sound source at location 2, and a normal distribution fit to these values. (orange).*

Python and Arduino Code

Link to Python code and Arduino code uploaded to GitHub:

https://github.com/lukemantle/Direction-and-ranging-with-difference-in-times-of-arrival