Arduino Sound Localization

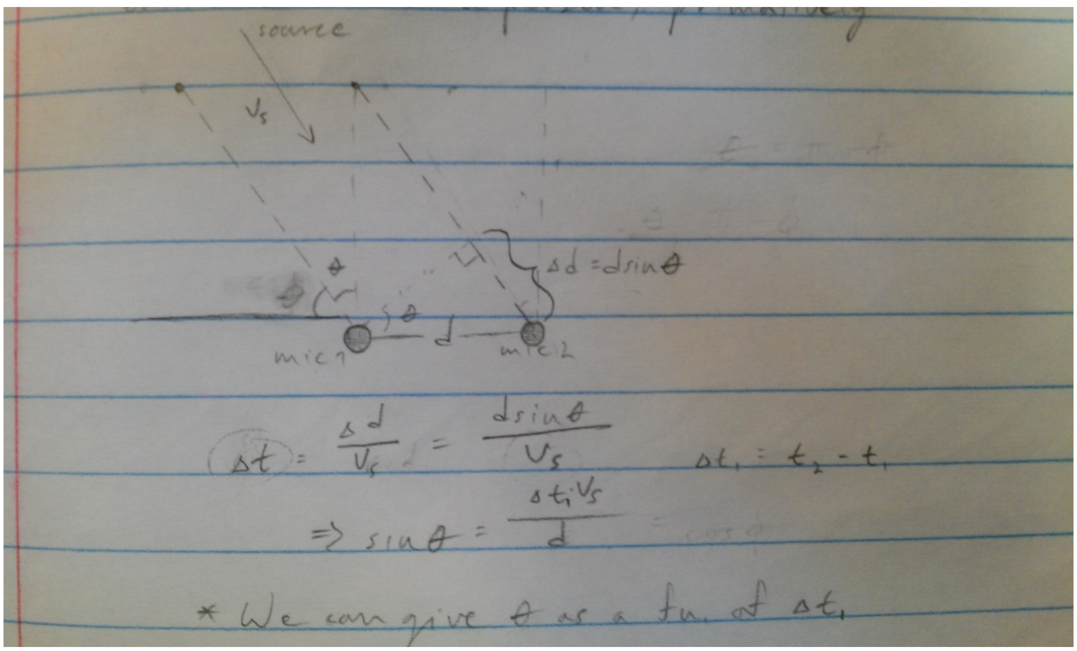
PHYS 406

Hiep Du

May 16, 2014

1. Introduction

For my project, I wanted to build an array of microphones using an Arduino Uno to attempt to reconstruct the incidence angle of a transient sound source relative to the two-microphone baseline. This project was inspired by one of the chapter exercises from “Science of Sound”. It involved calculating the distance between two microphones by using the leading edge time difference of a sound source hitting them and given the incident angle of the sound source. Using the very same analysis of the geometry of the system, one can work out an expression to describe the incident angle of the sound source as a function of the distance between the microphones and the leading edge time difference.

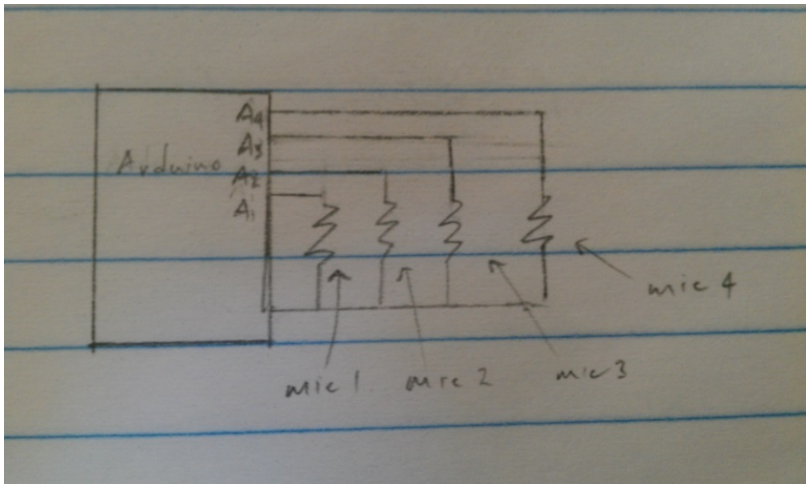


Derivation of $\sin\theta$ as a function of Δt and d, V_s = speed of sound in air

Here, one can see that the geometrical analysis that produces the sine of the angle of incidence θ is rather simple. It is easy to find the average speed of sound in air, and we are able to decide on the distance between the microphones. All that is left is to find the leading edge time difference and then calculate the angle of incidence. However, the bulk of the work involved in this project lies in the implementation of this physical system. This part of the project was a completely new challenge that I had not endeavored before. I lacked not only the skills in circuitry, but also the experience necessary to have insight on potential issues that may arise when dealing with a physical system. Much of the information I learned and used throughout this project was through Arduino resources as well as online bulletins.

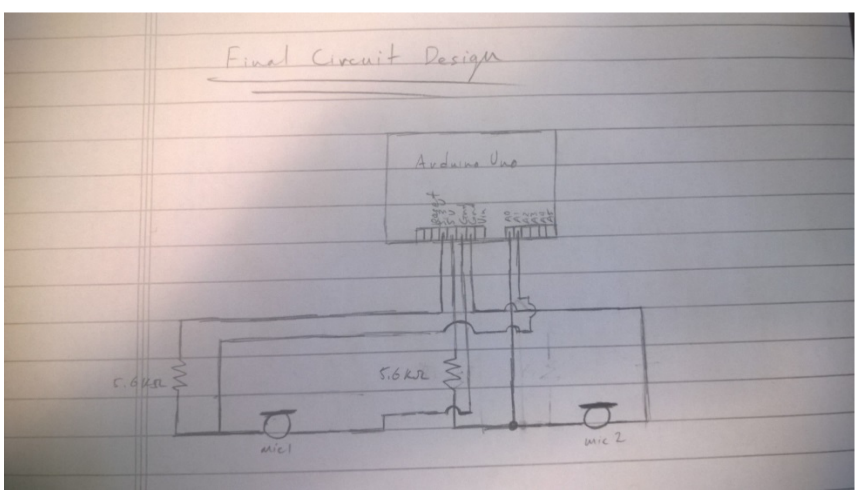
2. Strategy & Implementation

As mentioned, this was the first time I would be working with any sort of physical system as well as circuits. This led to a rather naïve initial design scheme.



Primary circuit diagram for a 4 microphone array

Immediately, my attempts at constructing this circuit were met with many issues. After much troubleshooting and experimentation with different resistors, I settled on a simple two-microphone array. The microphones are powered separately by the board's 5V pin and 3.3V pin separately. I decided to do this because, upon analysis using MATLAB, the data from each microphone will be easy to distinguish as they will have different values, while still achieving their purpose: to record any and all sounds that may strike it. This changes the capacitance between the plates of the condenser microphones, therefore creating a variation in the current, which is read by the analog pins on the Arduino as a changing voltage.

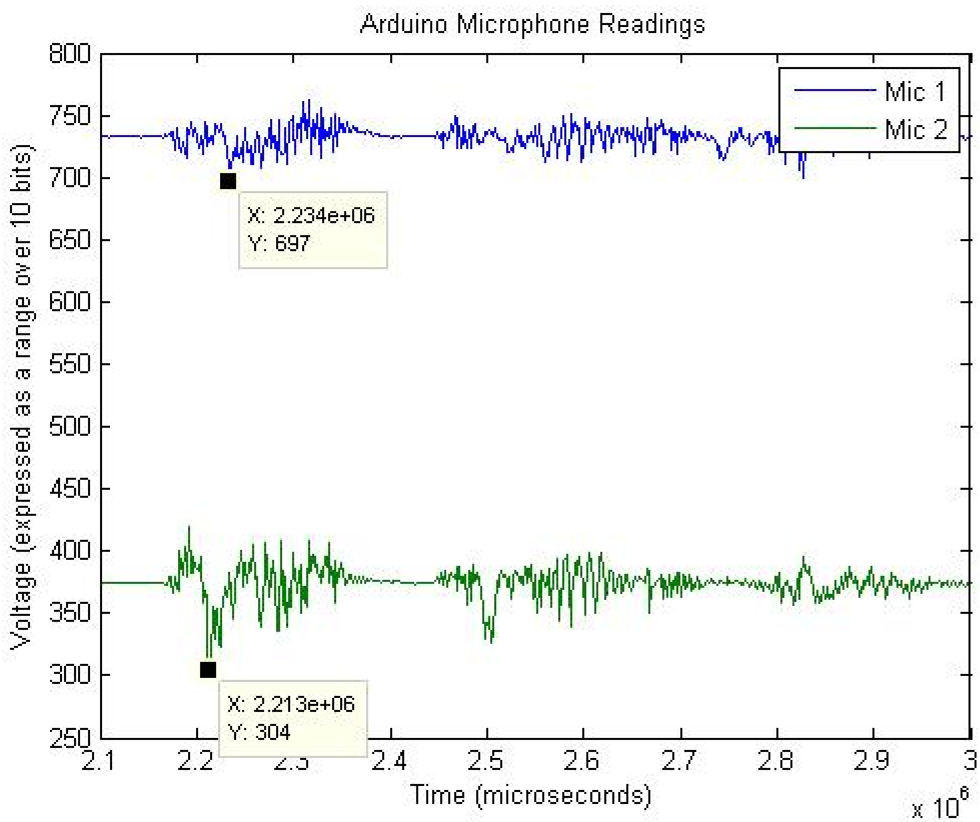


Final circuit design for a 2 microphone array

3. Testing and Analysis

Upon finally finding a working configuration for the microphone array, I needed some way to analyze the data that could be printed by the Arduino board. I just needed to the test for a few seconds to acquire data samples from a transient sound. I experimented with various sounds; I found that simply speaking into the array from some arbitrary angle worked best (versus clapping or tapping on a surface). Since I was running the test at 115,200 Hz, there would be more than enough data points to look at. The .txt file that the data is saved into should be attached, named “data.txt”.

To analyze the data, I wrote a simple MATLAB script to read the values and plot the values recorded from the first microphone and the second microphone versus time (in microseconds).



As one can see, I have labeled two data points on the plot. I chose to analyze these two points, as a minimum would represent a point where the sound disturbance was at a maximum. Using MATLAB, I was able to find times at which these two data samples were at a minimum. All that’s left is to take the difference and plug it into the expression given above. The MATLAB script will be attached as well, named “data_analysis.m”.

4. Results

time1 = 2234232 μs

min1 = 697

time2 = 2213236 μs

min2 = 304

timeDiff = -0.0210 s

v_s = 340.29 m/s

d = 0.16 m

angle = -9.0000e+01 + 2.5737e+02i

This result was baffling to me. How could the angle possibly be a complex number? Upon testing other points on the same data plot and getting the same results, I concluded that there must be something wrong within the argument of $\theta = \sin^{-1}(\frac{\Delta t * v_s}{d})$. It seems that it is out of the range of real values for arcsine. Since v_s and d are known physical values, I deduced that Δt must be too large! The leading theory I have for this is that, despite the looping frequency we commanded on the Arduino being on the order of 10^{-6} s, it may not be actually achieving this resolution. In order to get a reasonable value for the angle, the time resolution must be at most on the order of 10^{-4}. Unfortunately, this seems to be a property of the Arduino board itself. I am sure that, with better equipment, this experiment would indeed be successful.

5. Conclusion

In the context of my goal, I would consider this experiment a failure as I did not manage to reconstruct the angle of incidence of a transient sound source. Despite this, I am not discouraged and will continue to explore this topic further and potentially expand it to a more complex system. It was very interesting to see first-hand the effects of sound being of finite speed. In conclusion, this project was very frustrating, but a valuable learning experience in the application of real electrical systems. I wish that I was not so caught up with the “front-end” work involved with implementing a physical system. I would have liked to explore further the physics behind this deceptively simple idea. I am sure that there are several subtleties with the acoustics involved with this concept. Nevertheless, this was a fun experience and I am inspired to continue to explore the realm of acoustics! My next topic of interest is to explore noise cancellation/reduction with the implementation of a PID feedback control system. I am sure that this experience will at least make this next endeavor less painful that it could potentially be.