Evaluating a simple time-delay algorithm for the three-dimensional angle-of-arrival localization of sound

EEE4114F
Digital Signal Processing

University of Cape Town

Callum Tilbury and Matthew Hayes
TLBCAL002 and HYSMAT002

24 June 2020

Abstract

This project aimed to investigate the performance of a simple angle of arrival algorithm, different methods, from the literature, were investigated. An algorithm based off cross-correlation and basic linear algebra was then selected, and its performance was tested over a variety of conditions. These conditions included the type of sound event, the distance from the sound source to the microphone array, cases when two sound events overlapped, and the addition of "non-ideal" effects. To quantify the performance of the algorithm, it was evaluated using the average absolute error between the predicted angle and the true angle.

Overall, the performance of the algorithm depended largely on the use-case. It performed better with some sound classes, and much worse with others. Neither the incident angle, nor the synthetic non-ideal effects seemed to affect its performance markedly; however, there was much room for further investigation. Importantly, the distance from the source to the microphones, and the overlapping of sounds did have a noticeable effect on the average accuracy.

1 Introduction

Human beings have learnt the ability to figure out which direction a sound is coming from. For a computer to do this, multiple microphones can be used along with a combination of signal processing methods. There are many algorithms that process data from a microphone array to calculate an angle on the horizontal plane, the azimuth angle, and an angle in the vertical plane, the elevation angle. Using these angles the direction to a sound source can be found as shown in figure 1.

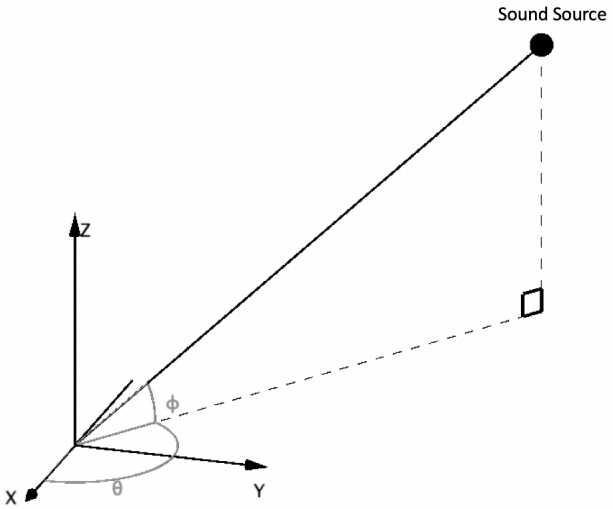


Figure 1: Diagram showing the azimuth angle (θ) and the elevation angle (ϕ)

There are several conventions for the way the azimuth and the elevation angles are defined. This report uses the definitions shown in figure 1, where the azimuth angle, θ, is the angle in the xy plane, from the x-axis to the component of the vector from the sound source to the origin, in the xy plane. The elevation angle, ϕ, is described as the angle from the xy plane to the vector from the sound source to the origin.

Many methods exist to calculate the direction of arrival, some are more complex than others. Some methods use neural-networks [1] and head related transfer functions [2], while others use simpler signal processing techniques along with linear algebra [3]. This leads to the question: How well does a straightforward angle of arrival algorithm, based on the time difference of arrival (TDOA) and some simple linear algebra, perform? And how will its performance change in differing conditions such as: distance to the sound source, type of sound, echo, overdrive, and multiple sounds at the same time (i.e sounds overlapping)?

Based off the literature, the hypothesis was that the algorithm would perform well in producing small errors when used in fairly simple contexts, and it was assumed that it would work equally well across a variety of sound classes. It was further predicted that the position of the sound source would not affect the accuracy significantly. When two sounds overlap, and non-ideal effects such as reverb, echo, and overdrive are added, it was expected that large errors would result.

This report starts with a literature review, where the theory of the angle of arrival algorithm is discussed, as well as a few alternative methods. This is followed by a description of the chosen method, which includes a short discussion on the data set used for testing the algorithm. Finally the results are discussed and conclusions are made.

2 Literature Review

Sound localization has been studied for many years; consequently, there is a substantial amount of literature on the topic. Modern research has attempted to use novel techniques to improve the shortcomings of previous work – including the use of Head-Related Transfer Function (HRTF) models [2, 4, 5], neural networks [1, 6], physical moulds of artificial pinnae [7], and various other new approaches. In this report, however, the focus was on implementing a straight-forward *time difference of arrival* algorithm – arguably a simpler and more traditional localization algorithm – and attempting to understand its limitations. This simple method uses the received audio signals from four microphones in a tetrahedral arrangement, to determine the angle at which a sound source exists in space, relative to the microphones. Fundamentally, the localization process can be broken into two distinct parts: firstly, estimating the time delay[1] between the pairs of received signals in the array; and secondly, calculating the angle of arrival of the sound source using these estimated time delays. Each of these will be discussed briefly below.

2.1 Time Delay Estimation

2.1.1 Signal Model

Consider two signals, $f_1(t)$ and $f_2(t)$, arriving at microphones M_1 and M_2 respectively. Though any digital signal will in fact be discrete and finite-time, the theory will first be developed for the continuous and infinite-time case. To simplify matters, an explicit assumption is made that the received signals are approximately identical, except for a time-shift, and possibly an attenuation factor[2] [10]. That is,

$$f_2(t) \approx \gamma f_1(t - t_0) \tag{1}$$

where t_0 is the time delay, and γ is the attenuation factor. Assuming the distance between the microphones is small, the latter should be approximately unity, (i.e. $\gamma \approx 1$).

2.1.2 Cross-correlation

The cross-correlation between the two continuous signals, $f_1(t)$ and $f_2(t)$, is defined in [11] as:

$$(f_1 \star f_2)(\tau) := \int_{-\infty}^{\infty} f_1^*(t) f_2(t + \tau) dt \tag{2}$$

where $f_1^*(t)$ is the complex conjugate of $f_1(t)$. Substituting (1) into (2) yields:

$$(f_1 \star f_2)(\tau) = \gamma \int_{-\infty}^{\infty} f_1^*(t) f_1(t + \tau - t_0) dt \tag{3}$$

$$= \gamma (f_1 \star f_1)(\tau - t_0) \tag{4}$$

which is just the autocorrelation of f_1, shifted by the time delay, t_0, and scaled by γ.

The output of the cross-correlation function indicates the degree of similarity between the two signals at various time-shifts. It can be easily shown that the autocorrelation function has a global maximum at $\tau = 0$ (i.e. with no time shift, which is logical as a signal is perfectly similar to itself) [12]. That is,

$$(f_1 \star f_1)(0) \geq (f_1 \star f_1)(\tau) \quad \forall \tau \in \mathbb{C} \tag{5}$$

Hence, the following conclusion can be made about the time delay [13]:

$$t_0 = \underset{\tau \in \mathbb{C}}{\operatorname{argmax}} (f_1 \star f_2)(\tau) \tag{6}$$

[1] Sometimes called the *interaural* time difference [8], as homage to the neuroscientific theory of how humans and animals localize sound, using their two ears [9].

[2] Benesty et al. directly address the complexities of noise, multi-path propagation, etc. and quantify these effects in [10], but such considerations are outside of this report's scope.

2.1.3 Discretization

In reality, of course, the signals received at the microphones cannot be infinite, nor represented in a continuous form. Instead, they are real-valued, finite-length, discrete sampled representations of the real-world audio signals. Consequently, a discrete cross-correlation formulation must be used, as shown in [14]. Moreover, instead of a time delay of t_0, there is now an integer *sample* delay of n_0. Assuming the signals $f_1[n]$ and $f_2[n]$ contain the same number of samples, N, the sample delay can be calculated as:

$$n_0 = \underset{n \in \mathbb{Z}}{\operatorname{argmax}} \sum_{k=0}^{N-1} f_1^*[n] f_2[((k+n))_N] \tag{7}$$

where $((x))_N = x \bmod N$.

The calculation of (7) can be performed fairly easily in a modern programming language. There are many packages that provide built-in time-domain cross-correlation functionality, and finding the argmax of the resulting array is trivial.

It is important to reiterate here that the discretized situation means that instead of a continuous time delay, t_0, there is a integer sample delay, n_0. This places an unavoidable limitation on the accuracy of the algorithm [3]. Suppose the received audio signals were sampled at a rate, f_s. This means that the time between any two samples—i.e. the sampling period—is $T_s = \frac{1}{f_s}$. This period is also the minimum recognizable difference between the signals received at two different microphones. Thus, when calculating the angle of arrival—which is shown later—using the discretized time value, the resulting angle value is discretized too. That is, the output of the algorithm can only assume a set of discrete values, introducing an inevitable error. The worst case of this error is when the real-world time delay, t_0, falls exactly halfway between the two samples of the signal, hence maximising the discretization error. Mathematically:

$$\max \left| \frac{n_0}{f_s} - t_0 \right| = \frac{T_s}{2} \tag{8}$$

This error, which is dependent on the sampling frequency, then propagates into the angle calculations below. The analytical solution of the final angle error due to discretization is outside of the scope of this report; however, this discussion provides a good insight as to why some of the errors may occur.

2.1.4 Fourier Domain Approach

Alternatively, the cross-correlation can be calculated using Fourier domain methods. The cross-correlation theorem [15] in continuous time states that:

$$(f_1 \star f_2)(\tau) \xrightarrow{\mathscr{F}} F_1^*(\Omega) F_2(\Omega) \tag{9}$$

where $F_1(\Omega)$ and $F_2(\Omega)$ are the Fourier transforms of $f_1(t)$ and $f_2(t)$ respectively. Using a discrete approximation of this theorem, the sample delay can be calculated as:

$$n_0 = \underset{n \in \mathbb{Z}}{\operatorname{argmax}} \ \mathrm{IDFT}\{F_1^* F_2\}[n] \tag{10}$$

where IDFT is the inverse discrete Fourier transform. This approach can be highly efficient when using the fast Fourier transform algorithm [16], depending on the size of the input. Empirical analysis would be required to determine which of the two methods is ideal in a given context; however, for the scope of this project, only the time-domain approach was used.

2.1.5 Alternative Methods

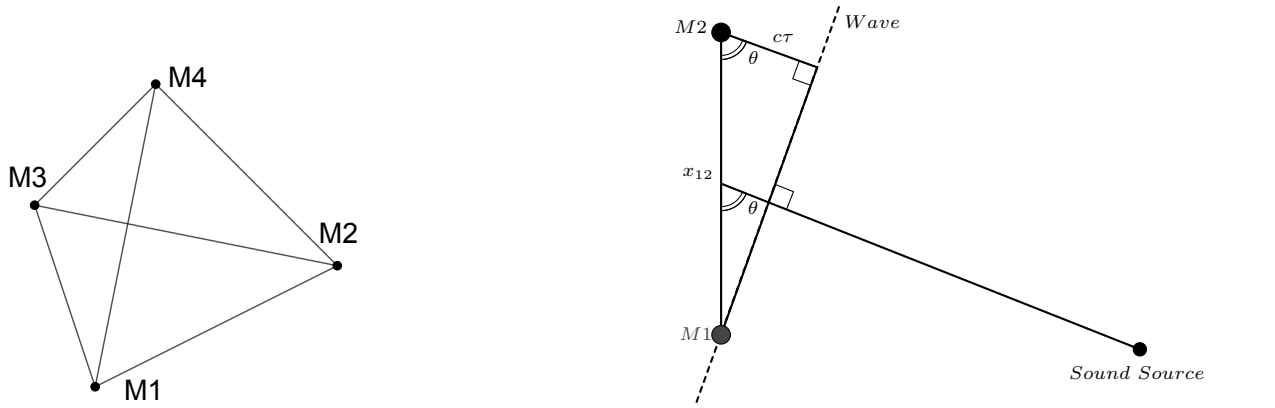
It is worth noting that there are many other methods that are conceptually similar to the time-delay approach. For example, almost a century ago, Friis et al. used the *phase* difference of received signals to estimate the

direction of arrival of short radio waves [17]. More recently, Cobos et al. in [18] used the time-frequency domain in a similar way, ultimately also comparing the phases of incident waves. Unfortunately, this approach results in an unavoidable spatial-aliasing effect—which is a function of the distance between the microphones—due to the 2π phase ambiguity. Other approaches include using the power difference of arrival [19], or the interaural level difference [20]. None of these approaches were considered in further detail for this report, though.

2.2 Angle of Arrival

2.2.1 Two-dimensional Approach

Figure 2a shows a simple representation of the tetrahedral microphone array. Consider specifically two of these microphones, M_1 and M_2, and the plane formed by these two points and S, the source of audio signal – for simplicity, S is assumed to be a point source. Provided the source is in the far-field, the incident wave on the microphones can be approximated to be planar [10]. Let the distance between M_1 and M_2 be x_{12}, and the angle between the midpoint of x_{12} and S be θ. Finally, let the time between the plane wave hitting M_1, and M_2 be τ. Figure 2b shows a top-down view of this arrangement.



(a) Simple graphic showing a tetrahedral microphone arrangement.

(b) The triangular arrangement formed by the incident plane wave and two of the microphones.

Figure 2: Graphics showing the microphone arrangement for the project

Notice that the arrangement in figure 2b forms a simple right-angled triangle. Once the time delay, τ, between the wave hitting M_1 and M_2 has been estimated, fairly simple trigonometry can be used to find the angle of arrival. The angle of arrival for this two-microphone setup is easily calculated as:

$$\theta = \arccos\left(\frac{c\tau}{x_{12}}\right) \tag{11}$$

where c is the speed of sound in air.

Kunin et al. [21] state that to find both angles in three-dimensions, a cross type microphone can be used, and then the method above can be applied to the horizontal and the vertical branches. However, this process is a bit more complicated if the microphones are not aligned with the global x, y and z axes.

2.2.2 Generalized Three-Dimensional Approach

The simple two-dimensional approach can be generalized to calculating the angle of arrival in three-dimensional space. Let the displacement from the sound source to the microphone arrangement be normalized and divided

by the speed of sound. The result is defined as the propagation vector, $\boldsymbol{k}$. That is,

$$\boldsymbol{k} = \frac{1}{c}\frac{x_o - x_{src}}{||x_o - x_{src}||} = \frac{1}{c}[\cos(\phi)\cos(\theta) \quad \cos(\phi)\sin(\theta) \quad \sin(\phi)]^T \tag{12}$$

where x_0 and x_{src} are the positions of the center of the microphone array and the source of the sound respectively and θ and ϕ are the azimuth and elevation angles respectively [18, 22]. The sensor vector between the i^{th} and j^{th} sensor in an array is defined as

$$\boldsymbol{x}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i \tag{13}$$

where x_n is the position of the n^{th} sensor [22]. Thus, looking at figure 2b, if one knows θ, then one can calculate τ using $\frac{x_{12}\cos(\theta)}{c}$. In this case x_{12} is the x-component of $\boldsymbol{x}_{12}$, where the y-component is zero. In general, as shown in [3], the relationship between the sensor vector, the propagation vector, and the time delay is:

$$\tau_{ij} = \boldsymbol{k}^T \boldsymbol{x_{ij}} \tag{14}$$

For a given microphone array, let the matrix $\boldsymbol{V}$ contain all the sensor vectors, and the vector $\boldsymbol{\tau}$ contain all the time delays. That is,

$$\mathbf{V} = [\boldsymbol{x}_{i_1 j_1} \quad \boldsymbol{x}_{i_2 j_2} \quad ... \quad \boldsymbol{x}_{i_n j_n}]^T \tag{15}$$

$$\boldsymbol{\tau} = [\tau_{i_1 j_1} \quad \tau_{i_2 j_2} \quad ... \quad \tau_{i_n j_n}]^T \tag{16}$$

If n time delays are calculated, then $\boldsymbol{V}$ will have the dimensions $n \times 3$, and $\boldsymbol{\tau}$ will have the dimensions $n \times 1$. As shown in [22], equation (14) can therefore be extended to,

$$\boldsymbol{\tau} = \boldsymbol{V}\boldsymbol{k} \tag{17}$$

Since $\boldsymbol{V}$ is not necessarily invertible, a linear least-squares approximation [23] can be used:

$$\Rightarrow \boldsymbol{k} = (\boldsymbol{V}^T\boldsymbol{V})\boldsymbol{V}^T\boldsymbol{\tau} \tag{18}$$

Finally, the angles of arrival can be calculated [18] using:

$$\theta = \arctan2(\boldsymbol{k}_y, \boldsymbol{k}_x) \tag{19}$$

$$\phi = \arcsin\left(\frac{\boldsymbol{k}_z}{||\boldsymbol{k}||}\right) \tag{20}$$

2.2.3 3D Angle of Arrival in the Fourier Domain

An alternative method of calculating the propagation vector is described by Cobos et al [18] using the phase difference between the i^{th} and j^{th} sensors. The phase difference at a frequency α is,

$$\angle\left(\frac{X_j(\alpha)}{X_i(\alpha)}\right) \approx \frac{2\pi\alpha}{c}\boldsymbol{x}_{ij}^T\boldsymbol{k} \tag{21}$$

For multiple sensors, the vector $\boldsymbol{B}(\alpha)$ is contains the n phase differences used (similar to $\boldsymbol{\tau}$ in equation (15)), and thus

$$\boldsymbol{k} = \frac{c}{2\pi\alpha}\boldsymbol{V}^{-1}\boldsymbol{B}(\alpha) \tag{22}$$

As before, θ and ϕ can then be calculated using equations (19) and (20).

Given the scope of this project, only the first approach was considered.

3 Method

3.1 The Data Set

In order to test the algorithm comprehensively, a reliably labelled and suitably large data set was required. Though several sets were considered, the collection of recordings from the *TAU Spatial Sound Events 2019* was suitable for this project, and was used for all investigations. Specifically, the *Microphone Array (Development)* audio data was used, together with its associated metadata. This set was originally part of the DCASE 2019 Sound Event Localization and Detection (SELD) task [24], where participants were required to both localize and classify various sound events from an array of microphone recordings. Though the current project did not cover the classification of recordings, the class labels nevertheless provided an key insight into understanding the performance of the algorithm in a variety of conditions.

A full description of the data set can be found in [25]. Each audio file was stored in a 4-channel, lossless WAV format, sampled at 48kHz, with each channel representing the received signal at one of the microphones. In fact, these recordings were made using the *Eigenmike* – a 32-element spherical microphone array[3] – from the company *MH Acoustics LLC*. Four of the 32 signals were extracted. The signals extracted were from channels 6, 10, 26, 22 and were chosen because the positions of these microphones form a tetrahedron. The positions of the microphones used, in spherical coordinates, were:

$$M_1 = (45°, 35°, 4.2 \text{ cm}) \qquad M_3 = (135°, -35°, 4.2 \text{ cm})$$
$$M_2 = (-45°, -35°, 4.2 \text{ cm}) \qquad M_4 = (-135°, 35°, 4.2 \text{ cm})$$

where the first angle given is the azimuth angle, and the second is the elevation angle. The last measurement is the distance from the centre of the microphone to the origin [6].

Two loudspeakers in the far-field of the microphone array were used to play a 'maximum length sequence' (which can suitably approximate an impulse) at many known locations in space, at various azimuth and elevation angles. These 'impulses' were then recorded by the microphones. In total, 324 recordings were made at 1 metre away from the centre of the microphone array, and 180 recordings at 2 metres away. These recordings were done in 5 unique indoor environments, with a range of furniture compositions, room sizes, and roof materials. Surrounding ambient noise was recorded separately, and added in to all recordings afterwards. It was ensured that the magnitudes of the impulses were at least 30dB greater than the level of the surrounding ambient sound.

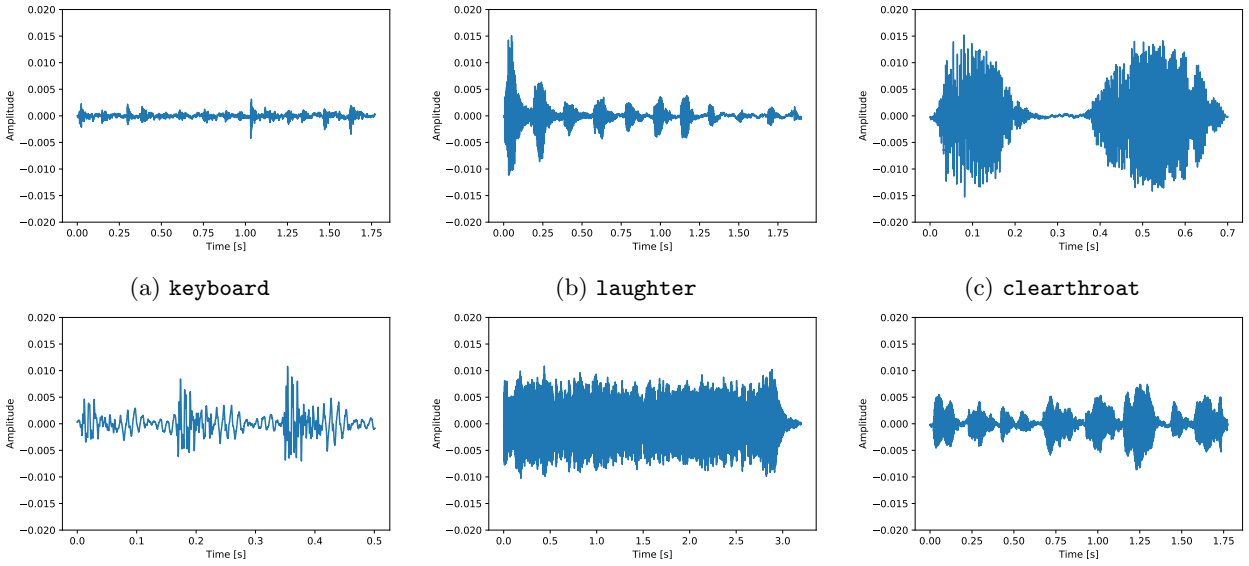
For the sake of the DCASE event the impulse recordings were then used to synthesize various *classes* of received signals, such that they could be 'classified' in the task. To do this, simple recordings of 11 different classes of sound events were used, as described in [26]. A particular sound event was convolved with a random impulse recording – thus simulating that sound event at the impulse location in space. This was done for the entire data set of impulses, with equal representation across the classes. Finally, collections of sound events were integrated temporally into 400 audio clips of 1 minute each. Around one third of the temporal integration allowed for no overlapping of events, and the remaining two thirds allowed for at-most two events to overlap at any given time.

For the testing of this project, all eleven classes were used from the data set. Six examples of different classes are shown in figure 3. It is important to note that the waveforms vary considerably in amplitude, length, and sparsity. All of these factors were predicted to play a role in the effectiveness of the simple angle of arrival algorithm.

3.2 Algorithm Implementation

The implementation of the algorithm was done in Python 3.7, with much of the core functionality designed using the `numpy` and `scipy` libraries. All of the code was run on an Apple MacBook Pro, Early-2015

[3] More information is available at: https://mhacoustics.com/products



(a) `keyboard` (b) `laughter` (c) `clearthroat`

(d) `knock` (e) `phone` (f) `speech`

Figure 3: Time domain waveform for examples of various event class

model, with a 2.7GHz dual-core Intel i5 processor, a 128GB solid-state drive, and 8GB of LPDDR3 on-board RAM.

Before the actual algorithm could be applied to the data, the individual chunks from the minute long recordings needed to be separated, and labelled accordingly. This entailed reading the metadata file alongside the audio file. Each of the 400 recordings had one corresponding metadata file, which was just an array of CSV entries, with each entry indicating a particular event within that recording. All chunks in a particular recording shared the same parameters of location and an 'overlap count' (either 1 or 2, where 1 indicated no overlap). For each chunk within a recording, the CSV columns indicated the temporal onset and offset times, the class of the sound, as well as the true elevation, azimuth, and radius values. Using the timestamps for each chunk, together with the known sample rate of 48kHz, the WAV file recordings could be read, sliced into individual events, and labelled appropriately. For each sound event, then, there were four arrays of values, one for each channel, coupled with that event's corresponding labels.

The first part of the algorithm entailed estimating the time delay between each pair of the four microphone signals using correlation. This was trivial in Python, particularly when using the functionality from the numerical library, `numpy`. The index of the peak output of the correlation function was calculated, and converted from samples to seconds by dividing by the sampling rate. The time delay results of each pair of microphones was then placed into a `taus` array.

The second part of the algorithm entailed setting up the arrays correctly, and invoking the appropriate linear algebra functions. The matrix `V` was populated with the sensor vectors, using the x, y, and z positions of the microphones, which were calculated from the spherical coordinates of the tetrahedral arrangement. Note that the order of the sensor vectors in `V` matched the order of the time delay pairs added to `taus`. With `V` and `taus` populated, `numpy`'s Linear Algebra library was used to calculate the least squares solution for the propagation vector – using equation (18). From this, the estimated azimuth and elevation angles could be calculated using simple trigonometric functions.

The simplified, general algorithm for calculating the angles of arrival in Python (pseudocode) is shown in listing 1.

Listing 1: General algorithm for calculating the angles of arrival for an event

```
# 4 channels of current event
channels = [[channel_0], [channel_1], [channel_2], [channel_3]]

# Calculate time delays and put into vector 'taus'
for i in range(4):
    for j in range(4):
        sample_delay = argmax(correlate(channels[i],channels[j]))
        time_delay   = sample_delay/fs
        taus.append(time_delay)

# Populate the matrix of sensor vectors, 'V'
for i in range(4):
    for j in range(4):
        sensor_vector = microphone[i] - microphone[j]
        V.append(sensor_vector)

# Apply linear least squares approximation
k = inverse( transpose(V) * V ) * transpose(V) * taus

# Calculate result
azimuth = arctan2(k[2], k[1])
elevation = arcsin(k[3]/norm(k))
```

3.3 Planned Experiments

Recall that the core of this project was to evaluate the performance of a simple approach to sound localization. To do this, a series of controlled experiments was run on the data, using an array of varying parameters. The entire data set was used for testing, which included 400 sound files, each composed of a multitude of event clips. In the end, over 60 000 individual event estimation tests were done, over a variety of test conditions.

A Python script was written to automate this process: each test was run with set parameters, and the results thereof were saved into a CSV file – containing all the information about that test, such as the location index of the original impulse recording, the overlap count, the event's class, and so on. Most importantly, included in the CSV output were the estimated- and true values for both the azimuth and elevation angles. All of this data was then taken into Microsoft Excel, where it was sorted, filtered, plotted, and analysed.

The experiments run for this project are summarised briefly below:

- Basic test: all data, no effects, no overlap
- Class Dependence test
- Radius Dependence test
- Location Dependence test
- True Angle Dependence test
- Overlap test
- Synthetic Audio Effects test

Each of these tests either grouped the data by some parameter, or applied some simulated non-ideal effect to the audio, and considered the performance of the algorithm over many test cases. For example, in the overlap test, the algorithm was run on an event played in isolation, and again run when it was played while overlapping with another event; the results of these two runs were then compared.

Moreover, throughout these experiments, a overarching issue was investigated: the impact of the event *class* on the accuracy of the results. This proved to be a critical insight into the performance of the algorithm. Thus, many of the results from the experiments were explicitly grouped by class membership in the output plots.

3.4 Simulated Non-ideal Conditions

Two of the experiments aimed to understand the impact of so-called 'non-ideal' conditions – somewhat more *realistic* scenarios – on the algorithm's performance. One of these was to look at how the results changed when two events were overlapping, compared to when they were not. Fortunately, the data set already provided cases of events with overlapping timestamps, as this formed part of the original DCASE challenge. Thus, when considering the impact of overlap, no additional effort was required to set up the tests.

The other non-ideal situation considered was the presence of *synthetic audio effects*, which attempted to mimic the real-world phenomena of echo, reverb, and overdrive. Each of these effects may naturally arise in certain physical scenarios, such as when attempting to localize in a large reverberant hall, or when there is excessive gain on the recording channel. To simulate the impact of such scenarios, the event data were altered in software using a simple audio-effects library. Granted, using synthetic effects is not a great picture of reality: a real-world recording of an echoey room will introduce a host of new complications, and this is similarly the case for the other effects. Nevertheless, this straightforward approach allowed an initial glance into the impact of such effects on the algorithm. There is certainly scope for more realistic investigations, based on the insights of this simple approach.

To add the audio effects in Python, the `pysndfx` library[4] was used. This package is actually just a wrapper for the powerful Sound eXchange (sox) audio processing tool, which usually uses a command-line interface. The `pysndfx` interface allows one to apply a host of effects on conventional `numpy` arrays, which made the simulation process trivial. As mentioned already, three effects were used, and the parameters thereof are given below:

- **Echo**: Gain In = 0.8; Gain Out = 1; Delay = 500ms; Decay = 0.5
- **Reverb**: Reverberance = 100; Wet Gain = 2
- **Overdrive**: Gain = 50

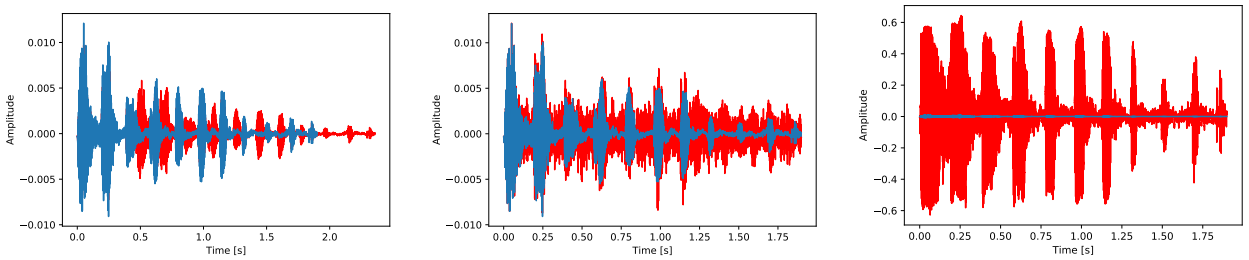
Consider figure 4 which shows the impact of the synthetic effects on the laughter sample from figure 3b.

3.5 Performance Metrics

To understand the performance of the time-delay algorithm, some sort of metric had to be chosen – something which would indicate the degree of success, or failure, across a range of data. Though many such metrics exist, each with particular qualities and drawbacks, this report used a simple error-based approach: the absolute error between the target angle, and the estimated angle.

Note, however, that the calculation of the error was not a naïve subtraction of the two values, as this would ignore the angular periodicity of the problem. For example, suppose the target angle was $-180°$, but it was estimated as $179°$. Simply subtracting these values would yield an absolute error of $359°$, whereas in fact the angles are only $1°$ apart. It is thus intuitive that the maximum possible error for *any* estimated angle is $180°$. To account for this fact, the difference between the two angles was confined to the principal range of

[4]More information is available at: https://pypi.org/project/pysndfx/



(a) Echo (b) Reverb (c) Overdrive

Figure 4: Time domain waveform showing the impact of the synthetic effects on a sample `laughter` event, where the *blue* signal is the original sound clip, and the *red* signal is the clip post-effect.

$[-180°, 180°)$, resulting in an absolute error in the range $[0°, 180°]$. This was done mathematically using the following process, for an arbitrary angle X:

$$|X|_{180°} = \begin{cases} X \bmod 360° & \text{if } (X \bmod 360°) \leq 180° \\ 360° - (X \bmod 360°) & \text{if } (X \bmod 360°) > 180° \end{cases} \tag{23}$$

The errors for the azimuth and elevation angles were then respectively defined as:

$$E_\theta = |\theta - \hat{\theta}|_{180°} \tag{24}$$

$$E_\phi = |\phi - \hat{\phi}|_{180°} \tag{25}$$

where $\hat{\theta}$ and $\hat{\phi}$ indicate the algorithm's output estimates, and θ and ϕ are the true angles of arrival from the sound source.

Since the algorithm was run on a *collection* of data, individual errors were mostly not considered in isolation. Instead, the errors were aggregated into a simple average value, indicating the overall performance achieved. Note that for each experiment done, the data was arranged into a variety of groupings – based on class, effect, location, etc. – and for each group, the average was calculated only over the members of that group. For example, when looking at the error for the `speech` events, only the `speech` event predictions were averaged, which is intuitive. All results presented in this report followed such a format, across a range of tests. For a given grouping, G, which contains N_G events, the absolute error was thus calculated as:

$$\bar{E}_\theta = \frac{1}{N_G} \sum_{i=1}^{N_G} |\theta_i - \hat{\theta}_i|_{180°} \tag{26}$$

$$\bar{E}_\phi = \frac{1}{N_G} \sum_{i=1}^{N_G} |\phi_i - \hat{\phi}_i|_{180°} \tag{27}$$

Notice that the errors of azimuth and elevation were not combined into a single metric. This was purposeful, as combining them would possibly lose interesting insights into the data. For all of the results, these angles were analysed separately.

Lastly, note that in some circumstances, instead of analysing the absolute error, the *change* in the error was considered – where a positive error change indicated a worse error than before. That is, when referring to the percentage error change from event A to event B:

$$\Delta\bar{E} = \left(\frac{\bar{E}_B - \bar{E}_A}{\bar{E}_A} \right) \tag{28}$$

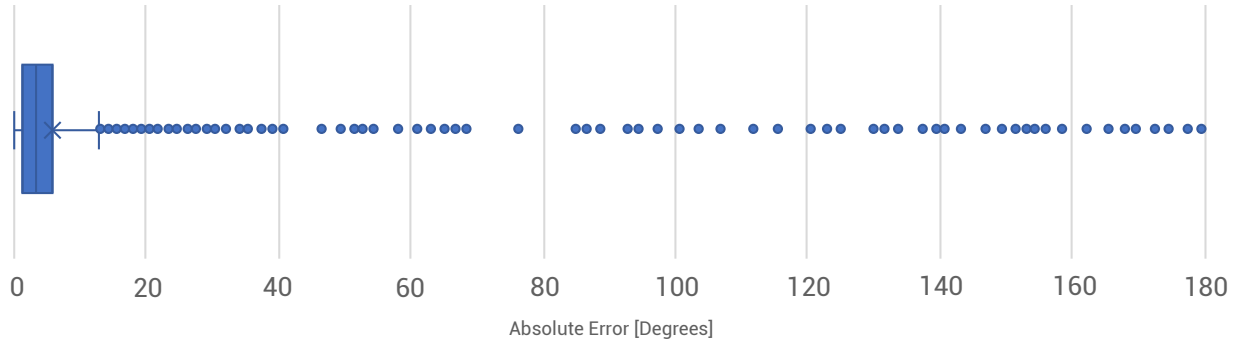
4 Results

The results of the experiments done are detailed in this section. Each experiment begins with a brief recap of its setup, followed by a presentation of the results obtained, and thereafter an analysis.

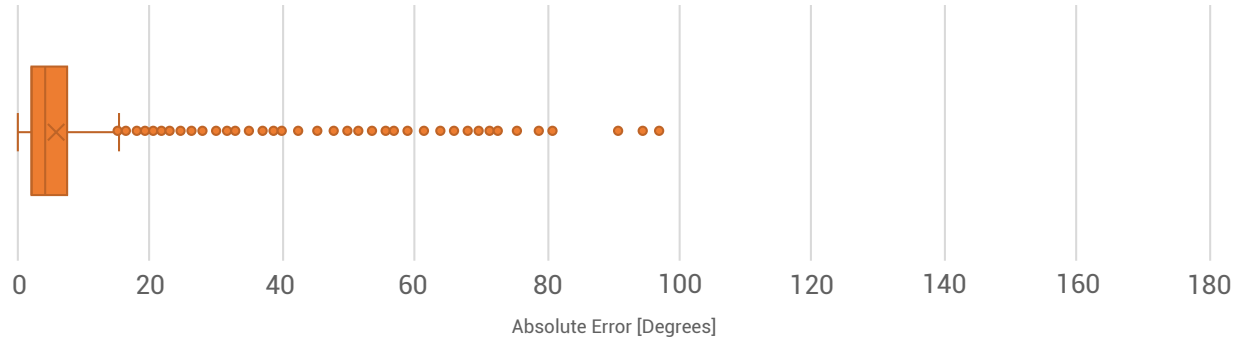
4.1 Basic test: all data, no effects, no overlap

This first test was to understand the 'raw performance' of the localization algorithm. In a sense, this experiment aimed to use conditions which were somewhat 'ideal'; whereas the addition of effects and overlap emulated non-ideal conditions. While there was already ambient noise added to all of the audio files in the original data set, it was fairly small. This test thus served as a baseline for future tests.

All of the non-overlapping data elements were used for this experiment – across all classes, locations, angles, and at both at radii. No audio effects were added to the files. The *spreads* of the absolute error results from these tests are shown as box-and-whisker plots in figure 5.



(a) Results for azimuth angle, θ



(b) Results for elevation angle, ϕ

Figure 5: Boxplots showing the absolute error between the target- and estimated- angle values.

Moreover, the key descriptive statistics for these results are highlighted in table I.

The immediate observation from these results were the numerous – and considerably large – outliers. Both for the azimuth (figure 5a) and elevation (figure 5b) results, some of the absolute errors were clearly unacceptable. For example, the azimuth error of 180° for one of the events is the worst possible situation – the algorithm could not have been more wrong. Unfortunately, this already shows that the algorithm simply cannot be used in critical situations, where accuracy is paramount. This fact was not necessarily surprising – given the simplicity of the approach – but was nevertheless clear.

	E_θ	E_ϕ
Mean	5.823°	5.711°
Std Dev	14.54°	7.037°
Median	3.428°	4.160°

Table I: Descriptive statistics for the basic experiment's error results

	E_θ	E_ϕ
Mean	4.218°	4.999°
Std Dev	5.187°	4.230°
Median	3.270°	4.064°

Table II: Descriptive statistics for the modified basic experiment's error results, where the `phone` class was excluded

The results are not completely bleak, though. It is important to note that this experiment alone considered over 5000 events. Indeed, outliers did exist – and these could not just be ignored – but *majority* of the errors were far smaller than these outliers. One can see both in the boxplots, as well as in the statistics from table I, that the algorithm performed fairly well on the whole: with an average error of just under 6° for both angles, and a median even smaller than that.

Note that the elevation angle had a smaller spread than that of the azimuth (and a smaller maximum error). This is likely because a smaller range of incoming angles were considered: in the DCASE data set, only elevation angles of $[-40°, 40°)$ were recorded, whereas the azimuth angles were varied over a full revolution, $[-180°, 180°)$.

These early results seem to suggest that, instead of being a *completely* useless algorithm, its use-case should be constrained. Since it is a straightforward approach, it may be suitable for situations where costs – development effort, power usage, etc. – are to be minimized, but a handful of seriously bad results are acceptable.

4.2 Class Dependence test

Consider the basic algorithm performance, now grouped by the event *classes*, still using the entire data set of non-overlapping sounds. Figure 10 shows a plot of the average absolute error for each group.

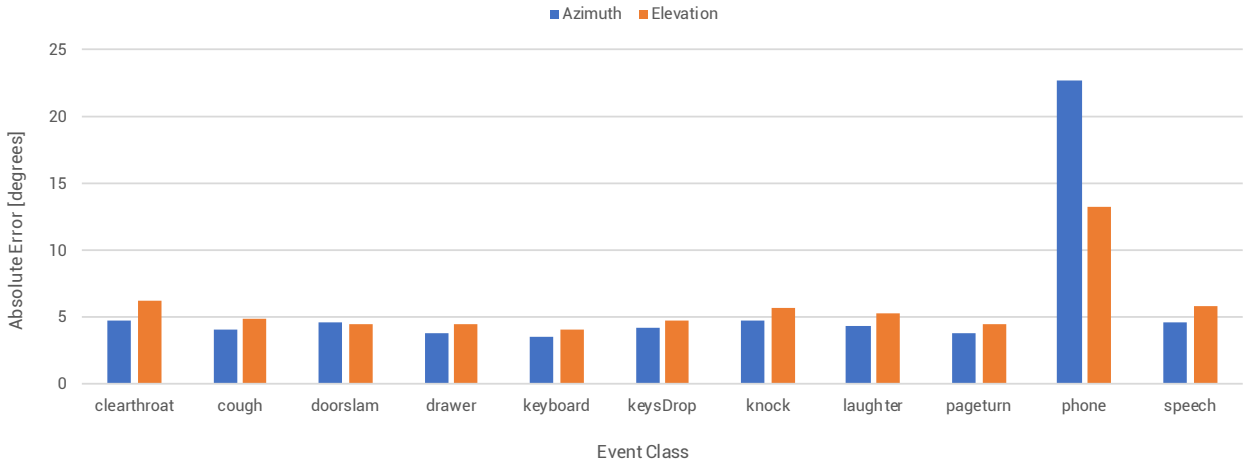


Figure 6: Absolute error for raw performance, split across the event class categories.

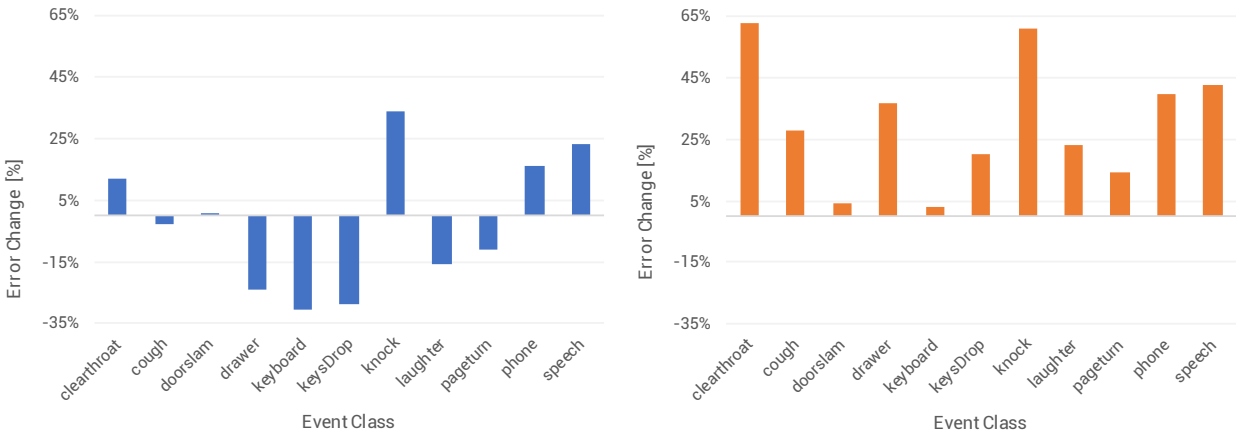
It is fascinating to note the massive distinction between the average error of the `phone` class, compared to the other classes. The variations amongst the other classes seem to be mostly negligible – varying only by a few degrees. The `phone` class, on the other hand, had noticeably larger errors than the rest of the group – a difference which could not be explained by mere chance. In fact, removing the `phone` class from the overall grouping improved the results drastically – reducing the mean, average, and median error. Consider table II

for these results. In particular, note the massive reduction in standard deviation when the `phone` class was removed, showing that it accounted for many of the previously shown outliers.

It is fair to conclude that for this particular event class – and perhaps for others that were not tested – the simple time-delay approach to localization performs poorly. More investigation would need to be done to make meaningful explanations of this fact – such as comparing the spectral components of the various event classes, and so on – but one theory is that the periodic nature of the ringing phone results in problems when finding the peak of the correlation function. Either way, it is an interesting exposure of a flaw in the algorithm.

4.3 Radius Dependence test

Consider next the effect of the radius of the sound source on the achieved estimation error. While the algorithm in this project could not reliably calculate the distance of the sound source to the microphone array – this was a known limitation – it was nevertheless interesting to understand how the radius affected the algorithm's ability to identify the incident angle of the sound. The data set had recordings of sound sources at both 1 and 2 metres away, and results up to this point have considered the distances together. Now, figure 7 shows the error change, $\Delta\bar{E}$, when moving the distance from 1 metre to 2 metres. Recall that a *positive* error change means that the algorithm's ability deteriorated – the error became larger.



(a) Results for azimuth angle, θ

(b) Results for elevation angle, ϕ

Figure 7: Accuracy change across event categories when moving source radius from 1 to 2 metres.

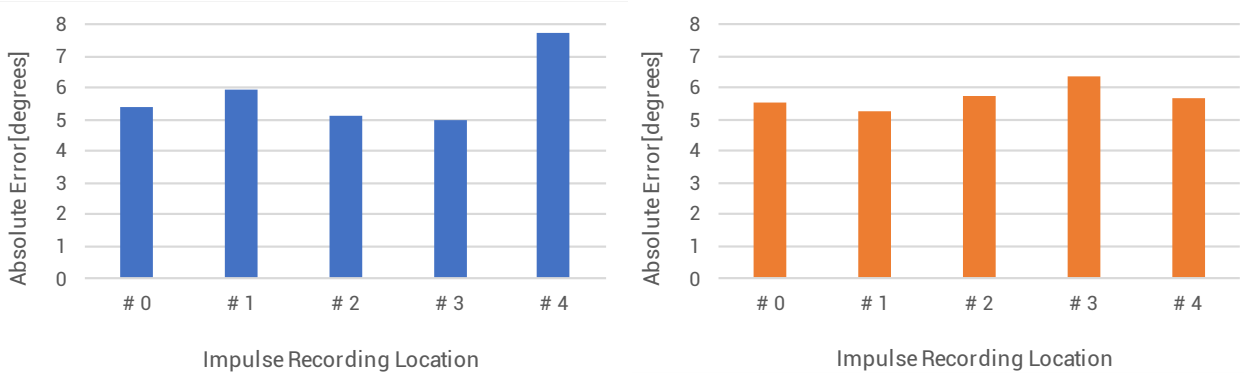
A couple of things are apparent about the radius dependence. Firstly, notice that the absolute error of the *elevation* angle estimation only became worse – with $\Delta\bar{E}_\phi > 0$ for all classes. In some cases, too, the error deteriorated badly: for example, for the `clearthroat` and the `knock` classes, the error increased by around 60%, which is a considerable amount. Yet for other classes, such as the `keyboard` and the `doorslam`, a minimal change was recorded. Secondly, notice that the *azimuth* error's results are mixed – in some cases, the algorithm performed *better* at a further radius, and in other cases, it performed worse.

It is difficult to know the exact meaning of these results. Naturally, at a larger radius from the microphone array, the sound wave from the source must propagate a further distance. As the sound propagates, assuming it is moving uniformly in all directions, its amplitude will ideally decay proportional to $(\frac{1}{R^2})$. This means the received waves from 2 metres away should have an amplitude at least 4 times smaller than those from 1 metre away. This certainly could explain the *poorer* performance of the algorithm, but fails to describe the *improved* performance, as was the case for some azimuth angles. It is clear, then, that deeper investigation is required into these phenomena.

4.4 Location Dependence test

Recall that the impulse recordings were done at five separate locations, each with unique acoustic characteristics and layouts. However, note that the data set simply referred to these locations with the indices 0 to 4, and those who recorded the data only provided bare-bones textual descriptions of the scenes – "large corridor, hard floors", for example. Moreover, several of the scenes were described with similar wording. It would thus have been difficult to infer defining characteristics from these descriptions, and any deductions made would likely ignore the true complexities of the environment. Thus, this experiment just sought to establish *if* there was a change in error across the locations, and not deduce explanations for observed differences, if any.

The experiment included all the non-overlapping data, across all classes and at both radii, but grouped the results by their original impulse recording location. Since there was no 'correct' recording location, using an error *change* metric was inappropriate; instead, the absolute error for each location was considered. Figure 8 shows the results of the tests.



(a) Results for azimuth angle, θ

(b) Results for elevation angle, ϕ

Figure 8: Absolute error compared across the five recording locations.

In this test, the results were fairly uninteresting. While there were differences between the errors across the locations, most noticeably so for the azimuth angles, they only amounted to a few degrees. Indeed, a few degrees did represent a somewhat significant error *change* between the locations, but one must remember that this experiment included data across *all* classes. As already seen, the class dependence was a major factor in the error results, and could have adversely affected these results. Moreover, recall that in the creation of the data set, impulse recordings from defined locations were convolved with *random* event samples from the 11 classes. This randomness may have introduced anomalous results, and could explain the differences between the locations. Having said that, this is not a certainty.

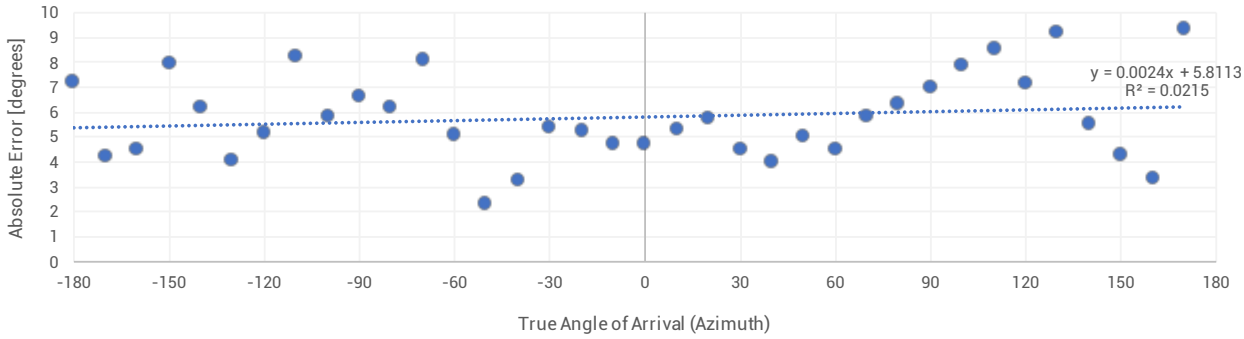
Once again, deeper investigation would have to be done to order to make more robust conclusions. On the surface, though, there is no marked difference between the calculated errors.

4.5 True Angle Dependence test

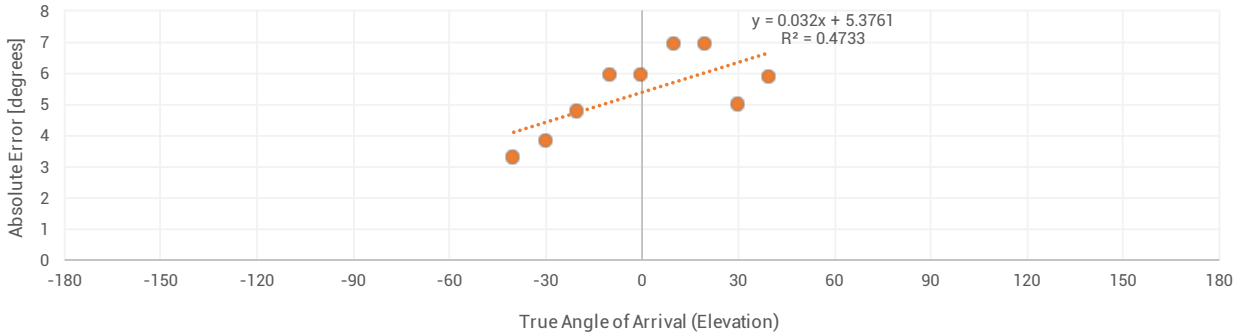
This experiment aimed to understand if the true angle of arrival from the sound source affected how accurate the algorithm was. Already it has been shown that there were some peculiar results indicating radial dependence – particularly for the elevation estimation. The angle dependence was expected to be different, however. Whereas the radius to the sound source directly affected the amplitude of the incoming wave, the angle was expected to be irrelevant due to the symmetric tetrahedral shape of the microphone array setup.

All the non-overlapping data was used for this test, and figure 9 shows the resulting absolute error metric,

plotted against the range of true angle values. Moreover, a straight line of best fit – using least squares regression – was calculated and superimposed for each of the angle plots. Recall that for the azimuth angle, the true angle values were recorded in steps of 10° over the full revolution of $[-180°, 180°)$; and for the elevation angle, the same step size was used, but over a reduced range of $[-40°, 40°)$.



(a) Results for azimuth angle, θ



(b) Results for elevation angle, ϕ

Figure 9: Absolute error considered against the true received angle value

Firstly, consider the plot in figure 9a, for the azimuth angle. This set of results had a large number of individual data points, as there were many distinct azimuth angles used in the original data set. According to the regression coefficients, there was no linear relationship between the true azimuth angle, and the algorithm's error, (with $R^2 = 0.02$). Furthermore, by simply judging the plot visually, one cannot seem to find a clearly defined *non-linear* relationship either. Instead, by the looks of it, the discrepancies in the errors across the range of angles were somewhat random. Once again, this randomness could perhaps be explained by the method in setting up the data set, with higher errors from certain event classes randomly matching with some angles more than others, thus skewing the data.

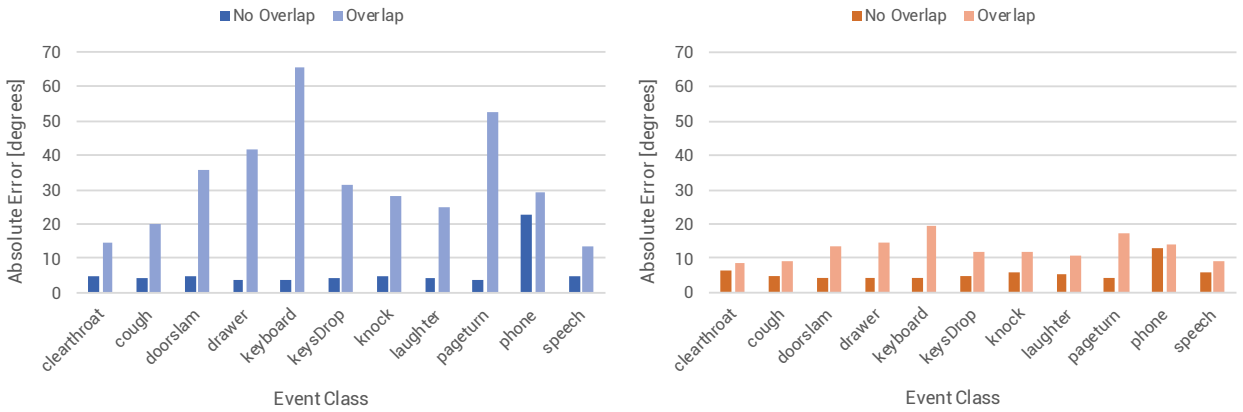
Secondly, consider figure 9b, for the elevation angle. Unfortunately, there were fewer unique elevation angles used in the original data set – compared to many unique azimuth angles over a larger range – and this resulted in a sparser plot. One may be tempted to infer the conclusion from the azimuth case – that there was no angle dependence – to the elevation case. However, as seen in most of the previous experiments, interesting differences exist in the algorithm's performance when determining the azimuth and elevation angles. And indeed, in the case of the few elevation angle points, there was actually a weak linear relationship evident in the data. This possibly indicates that as the true elevation angle increased from $-40°$ to $40°$, the error worsened. This relationship is markedly weak, however, and is based off only a handful of points. To make this claim more convincing, one would need to do more tests, using more distinct angles, and possibly over a larger range.

On the whole, then, there did not seem to be a noticeable relationship between the angle of arrival and the

corresponding absolute error. If such a relationship did exist, it was not particularly impactful.

4.6 Overlap test

The next completed experiment entailed temporally overlapping two sound sources. The data set used in this project already included overlapping sound chunks, and so testing this was straightforward. At most, two events were allowed to play simultaneously, and the overlapping event class was chosen at random. Consider figure 10, which shows the absolute error of each class, for both angles of arrival.



(a) Results for azimuth angle, θ

(b) Results for elevation angle, ϕ

Figure 10: Absolute error compared between no overlap, and random overlap, across the event categories.

The results of this test exposed an unequivocal flaw in the algorithm's approach: it could not reliably localize an audio source when the sound produced by the source was overlapping with a separate sound. Notice in the figure how, when the events overlapped, the average errors increased dramatically – for example, in the case of the `keyboard` class, the azimuth error increased tenfold, to over 65°. Furthermore, note how the severity of the problem varied across the event classes, with some classes affected by considerable amounts, and others not as much. The performance of the `speech` and `clearthroat` classes deteriorated the least, for both the elevation and azimuth angles.

These findings were not actually surprising when considering the theory used in the localization algorithm. Since the received time domain waveforms needed to be correlated for time-delay estimation, a loud overlapping signal coming from a different direction caused the weaker signal to be 'hidden' in time, unable to be detected by the cross-correlation process. Instead, the louder signal redirected the localization output, and thus caused large errors. This notion is supported by the results seen in figure 6, where the events with sparse, low-amplitude time-domain waveforms struggled against those with large-amplitude, dominant waveforms. For example consider the `speech` and `clearthroat` waveforms shown previously in figures 3f and 3c respectively, and compare them to the `keyboard` and `knock` waveforms from figures 3a and 3d, where the latter two signals performed far worse than the former two.

It is interesting to note that the elevation angle estimation, while indeed negatively affected by the overlapping sounds, was (on the surface) more robust than the azimuth angle estimation. As the worst case event class, the `keyboard` class average error rose only to 20°. It is crucial, however, not to draw the wrong conclusions here. The distinction (or at least part thereof) can be explained by two factors. Firstly, there were fewer unique elevation angles used in the data set, with only 9 possible angle values, compared to the 36 possible azimuth angles used. This unavoidably meant there was a higher chance that the overlapping sound would have the same elevation angle to the actual sound being localized. Thus, on aggregate, even when dominating sounds played over weak sounds, the average error between them would be smaller. Secondly, the range of

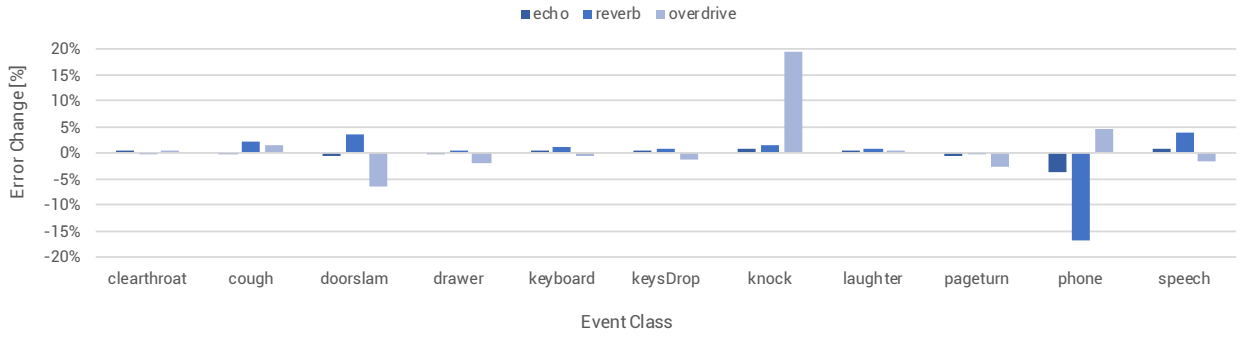
the elevation angles was limited to $[-40°, 40°)$. Thus, the smaller errors were not only due to the fewer *number* of possible angle values, but the fact that an overpowering sound's elevation angle could, at most, be 80° away from that of the correct sound's location. This should be compared to the azimuth angle, which could at most be 180° away.

Now, it is not *impossible* that for other reasons, the elevation angle is generally more accurate than the azimuth angle. Several of the previous experiments may even support this claim. However, a wider array of tests would need to be run, across a broader range of angles, in order to understand such an effect truly, and in order to make meaningful conclusions about it.

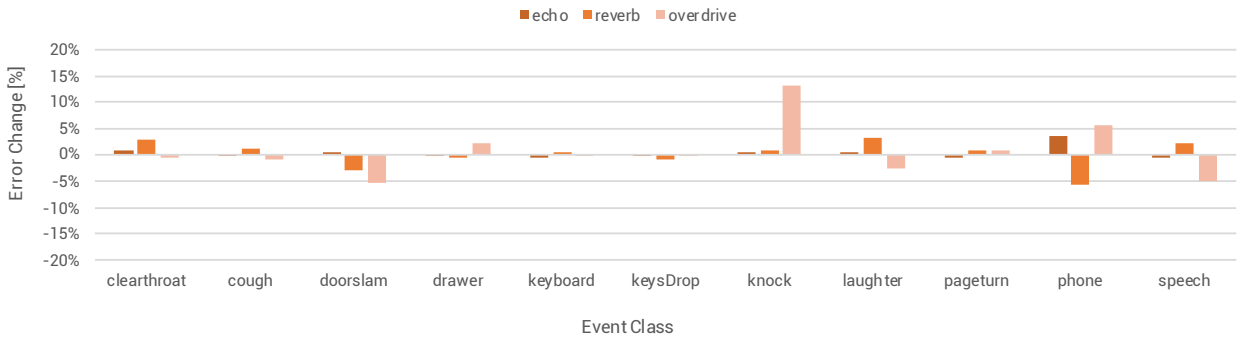
In essence though, for the azimuth and elevation angles alike, the overlap test harshly affected the accuracy of the estimated angle of arrival. This is a massively important note to make, because sound localization in reality will often occur in a multi-source environment. The algorithm would clearly struggle in such conditions, and would thus be unsuitable. However, once again, this does not necessarily make the approach completely pointless; rather, it depicts a constraint on the applications for which such a simple algorithm would be appropriate.

4.7 Synthetic Audio Effects test

Finally, the last experiment involved applying synthetic audio effects to the received recordings to emulate various physical phenomena. Specifically, effects of echo, reverb, and overdrive were applied separately to all the non-overlapping events in the data set. Figure 11 shows the results of these tests, grouped by class membership. The plots are shown with respect to the error *change*, $\Delta\bar{E}$, from the pre-effects, original sound event, to the post-effects event. Once again, a positive error change here indicates that the performance has deteriorated.



(a) Results for azimuth angle, θ



(b) Results for elevation angle, ϕ

Figure 11: Accuracy change across event categories after simulating three synthetic audio effects

Somewhat surprisingly, on the whole, the applied effects actually made little difference to the achieved errors. Granted, there were a few notable exceptions – such as the application of overdrive to the `knock` signal, which increased the average error by 20%. Moreover, in a bizarre way, for some classes the audio effects actually *improved* the average accuracy. The `phone` event, for example, had its average error reduced by 16% when reverb was applied to its recordings. Barring these anomalies, however, the effects – particularly the echo effect – made negligible differences to the algorithm's accuracy, for both the elevation and azimuth angles.

These findings certainly highlight an advantage of this algorithm, though the celebration thereof must be limited. Indeed, the results showed that even in the presence of non-ideal effects, albeit simulated, the correlation algorithm could still successfully calculate the correct time delay, and therefore the correct angle of arrival. This occurs, arguably, because the method does not generally rely on the quality or contents of the actual signals, but rather the similarity of the signals received at the four microphones. Thus, provided the effects are applied somewhat uniformly to all channels, the algorithm can still perform fairly well. This explains why the echo effect made little difference – the time delay of the decaying repetitions were applied equally to each channel. This allowed the algorithm to perform just as well.

However, the proviso of uniformity highlights a key issue: in reality, these non-ideal effects would likely not apply to each channel *uniformly*. In a reverberant room, for example, the relative positions of each microphone, the sound source, and the scene objects – walls, furniture, roof, etc. – would all play a massive role in affecting the recorded signals; and, importantly, each of the elements of the microphone array would thus receive a *different* reverberated signal. This is bound to cause much more severe error degradation and loss of accuracy. Similar arguments can be made for the real-world echo and overdrive effects. Hence, there is much scope for future work to explore the impact of real world effects, or, at least, synthetic effects applied non-uniformly to the microphone channels.

5 Conclusion

From the results, it is clear that that the algorithm performs well on specific cases and performs poorly on others. Therefore, the use of the algorithm would heavily depend on the use case. From figure 3, one can see that the shape of the waveform for the `phone` event is very different from the other events. Further research could be done to investigate if the shape of the waveform affects the accuracy of the angle of arrival algorithms. Furthermore, possible improvements to the algorithm may include up-sampling the data. It should be noted that the distance from the sound source to the microphone array does affect the accuracy of the algorithm. This change in accuracy is not negligible, as a small error in the angle of arrival over a large distance can cause a large error in the actual position of an object.

It was found that the surroundings of the sound source do not have a large effect on the accuracy of the algorithm, although again, further research could provide more information about how the environment affects the algorithm.

From figure 9 one can conclude that there is no clear relationship between the true angle and the error for the *azimuth* angle. However, as the sound events in the data set had a narrow range of elevation angles, more research must be done, with a more comprehensive data set, to determine if there is a relationship between the true *elevation* angle and the corresponding output error.

As expected, when more than one sound event occurs at the same time the results from the algorithm are not usable. This is most likely because the cross-correlation does not produce a peak at the right time due to the two signals interacting with each other. Therefore an important conclusion is that this algorithm can not be used for direction of arrival detection in cases when many sounds are overlapping. In these cases, a more complex method, such as a neural network, may be more applicable.

The addition of echo, reverb and overdrive to the signals, surprisingly had a very small effect on the accuracy of the algorithm, barring a few anomalies. However, these 'non-ideal' effects were synthetic and not actually recorded in real signals. Thus, it is possible that the algorithm's performance may deteriorate when the conditions are not similar to those in the data set. There is certainly a lot of space for further research into the effects of true, real-world 'non-ideal' effects on the performance of the algorithm.

In summary, the simple algorithm proved to work in fairly simple contexts. Further investigation is required to make meaningful conclusions about some of the tests – such as the radius dependence – and this may include sourcing a larger, more diverse data set. Some of the results obtained in this project were unequivocal, though. For example, the algorithm clearly fails in the context of overlapping signals, and this constrains the use-cases for which it could be reliably applied. Moreover, the class of the sound source was shown to have a possible impact on the resulting error, and this further constrains its usefulness. Ultimately, using such a straightforward approach cannot account for many of the complexities of real-world applications; however, it should not be completely overlooked in modern investigations.

A Resource Repository

All the code and resources used in the project can be found here[5].

[5] https://drive.google.com/drive/folders/12PrnlK9D84uCc3SOAHmLlOYRV3BqwcPc?usp=sharing

References

[1] Y. Zhang, Z. Wang, W. Wang, Z. Guo, and J. Wang, "Solo: 2d localization with single sound source and single microphone," in *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, 2017, pp. 787–790.

[2] M. Usman, F. Keyrouz, and K. Diepold, "Real time humanoid sound source localization and tracking in a highly reverberant environment," in *2008 9th International Conference on Signal Processing*. IEEE, 2008, pp. 2661–2664.

[3] J. Yli-Hietanen, K. Kalliojarvi, and J. Astola, "Low-complexity angle of arrival estimation of wideband signals using small arrays," in *Proceedings of 8th Workshop on Statistical Signal and Array Processing*, 1996, pp. 109–112.

[4] A. Pourmohammad and S. M. Ahadi, "Tde-ild-hrtf-based 3d entire-space sound source localization using only three microphones and source counting," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–6.

[5] N. Sakamoto, W. Kobayashi, T. Onoye, and I. Shirakawa, "Dsp implementation of 3d sound localization algorithm for monaural sound source," in *ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No. 01EX483)*, vol. 2. IEEE, 2001, pp. 1061–1064.

[6] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[7] F. Keyrouz and K. Diepold, "An enhanced binaural 3d sound localization algorithm," in *2006 IEEE International Symposium on Signal Processing and Information Technology*, 2006, pp. 662–665.

[8] X. Chen, C. Hu, B. Zhong, H. Xu, and X. Feng, "Improved algorithm of interaural time difference based on correlation method," in *2016 IEEE International Conference on Mechatronics and Automation*, 2016, pp. 2296–2301.

[9] J. W. H. Schnupp and C. E. Carr, "On hearing with more than one ear: lessons from evolution," *Nature Neuroscience*, vol. 12, no. 6, pp. 692–697, may 2009.

[10] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[11] R. Bracewell, "Pentagram notation for cross correlation. the fourier transform and its applications," *New York: McGraw-Hill*, vol. 46, p. 243, 1965.

[12] J. Deane. [Online]. Available: http://personal.ee.surrey.ac.uk/personal/st/J.Deane/Teach/eee2035/autoc_peak.pdf

[13] M. Azaria and D. Hertz, "Time delay estimation by generalized cross correlation methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 280–285, 1984.

[14] M. Rhudy, B. Bucci, J. Vipperman, J. Allanach, and B. Abraham, "Microphone array analysis methods using cross-correlations," in *Volume 15: Sound, Vibration and Design*. ASMEDC, jan 2009.

[15] E. W. Weisstein, "Cross-correlation theorem." [Online]. Available: https://mathworld.wolfram.com/Cross-CorrelationTheorem.html

[16] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast fourier transform?" *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.

[17] H. T. Friis, C. B. Feldman, and W. M. Sharpless, "The determination of the direction of arrival of short radio waves," *Proceedings of the Institute of Radio Engineers*, vol. 22, no. 1, pp. 47–78, 1934.

[18] M. Cobos, L. Jose, and S. Sascha, "A sparsity-based approach to 3d binaural sound synthesis using time-frequency array processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, 02 2010.

[19] A. Navarro, W. Cruz, C. Amu, and Y. Castano, "Broadcast emitters localization using power difference of arrival," in *2018 IEEE MTT-S Latin America Microwave Conference (LAMC 2018)*, 2018, pp. 1–3.

[20] S. T. Birchfield and R. Gangishetty, "Acoustic localization by interaural level difference," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 4, 2005, pp. iv/1109–iv/1112 Vol. 4.

[21] V. Kunin, W. Jia, M. Turqueti, J. Saniie, and E. Oruklu, "3d direction of arrival estimation and localization using ultrasonic sensors in an anechoic chamber," in *IEEE International Ultrasonics Symposium Proceedings*, 2011.

[22] J. Yli-Hietanen and T. Saarelainen, "Analysis of robust time-delay based angle-of-arrival estimation methods," in *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, vol. 1, 2002, pp. 239–242 vol.1.

[23] D. Margalit and J. Rabinoff. [Online]. Available: https://textbooks.math.gatech.edu/ila/least-squares.html

[24] Dcase, "Sound event localization and detection." [Online]. Available: http://dcase.community/challenge2019/task-sound-event-localization-and-detection

[25] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," *arXiv preprint arXiv:1905.08546*, 2019.

[26] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, Feb 2018.