

Recognizing Emotions from Audio Waves

Extended Essay in Mathematics

Research Question: To what extent can Fourier Transforms be used along with Machine Learning to decode emotions from speech recordings?

Contents

1. Introduction	3
1.1 Background Information	3
1.1.1 Fourier Transforms.....	3
1.1.2 Rationale	3
1.1.3 Finding meaning in audio waves.....	4
2. Fourier Transforms.....	4
2.1 Introduction to Fourier Transforms	4
2.2 The Fourier Series	6
2.2.1 The Trigonometric (Real) Series.....	6
2.2.2 The Exponential (Complex) Series	14
2.3 Deriving the Fourier Transform from Fourier Series	17
3. Retrieving Emotion from Audio	19
Sound Waves.....	19
Analyzing an Audio Sample	19
4. Conclusion.....	27
Potential Application.....	28
Further Scope	28
Bibliography	29

1. Introduction

1.1 Background Information

1.1.1 Fourier Transforms

Fourier transforms have the simple (yet mathematically complex¹) purpose of breaking down waveforms into an alternate form of varying sine and cosine functions. Essentially, it depicts that any waveform (a function of time) can be re-written as a sum of sinusoids². A common use of Fourier Transforms is in *Signal Processing*, processing any waveform (which could be light waves, your speech, or even stocks) to extract relevant information. For example, you could “filter out” unwanted parts of a signal by breaking it down, such as removing distracting background sounds or digital noise in a photograph. Despite this concept having a presumably narrow focus, the Fourier Transform abounds in applications in seemingly unrelated areas of Math and Physics, like the Uncertainty Principle³ or even the Riemann Zeta Function⁴. For this essay, I wanted to take advantage of this ubiquity of Fourier Transforms.

1.1.2 Rationale

After watching a video⁵ that intuitively explained the idea of Fourier Transforms through examples of sound signals, I was intrigued by the extent of novel exploration that this concept offers. The rabbit hole of research that followed this curiosity led me to discover that we could recognize aspects of speech exactly like humans do, using Mathematics. This was astonishing to me: the intricate and sophisticated skill of recognizing audio could be pinned down to a mathematical function.

¹ <https://scholar.harvard.edu/files/schwartz/files/lecture8-fouriertransforms.pdf>

² <https://www.thefouriertransform.com/>

³ <https://math.uchicago.edu/~may/REU2013/REUPapers/Hill.pdf>

⁴ <https://hp.hisashikobayashi.com/wp-content/uploads/2015/12/Riemann-Hypothesis-No.4-shortened.pdf>

⁵ <https://www.youtube.com/watch?v=spUNpyF58BY>

1.1.3 Finding meaning in audio waves

Sound waves are just disturbances in a medium (like air) that form areas of compressions and rarefactions (alternating regions of high pressure and low pressure). Areas of high pressure and low pressure seem astronomically disconnected from something that could have meaning in them (like emotions or language), and the fact that this disconnection could be lapsed by Mathematics left me in admiration. These pressure variations could be translated to graphical representations of waves: in terms of the frequency (the number of rarefactions and compressions that occur *per unit time*), and amplitude (the magnitude of the maximum disturbance in a sound wave) —of which can be represented as a function of time, as these aspects of the sound wave may vary across a length of time in human speech. **If it can be represented as a function of time, we can deconstruct it to a sum of sinusoids.** These sinusoids could be then used to find patterns that, for example, show the presence of a particular emotion in them.

1.2 Research Question

To what extent can Fourier Transforms be used along with Machine Learning to decode emotions from speech recordings?

2. Fourier Transforms

2.1 A Mathematical Understanding Fourier Transforms

All waveforms are the sum of simple sine waves (sinusoids) with different frequencies and amplitudes.⁶

The purpose of a Fourier Transform is to split a waveform into its fundamental sinusoids.

For example, let's break down the wave function in Figure 1.

⁶ <https://www.thefouriertransform.com>

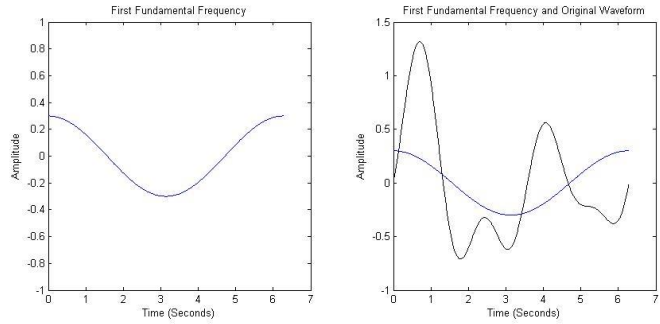
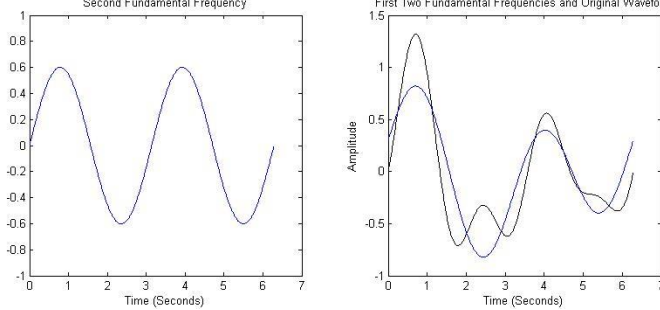
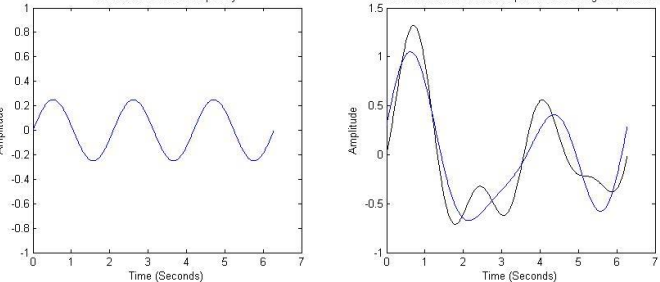
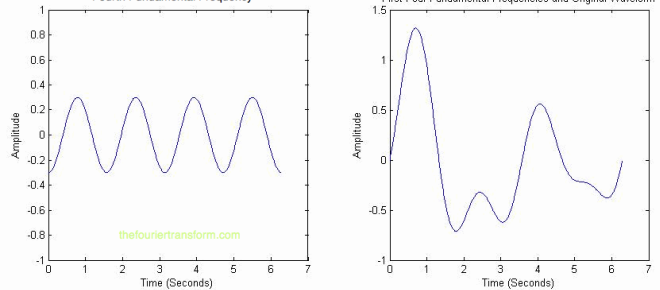
	<p>The First Fundamental Frequency shows the first sinusoid component of the original waveform function on the right.</p>
	<p>The blue function on the right shows the sum of the Second Fundamental Frequency (which is another sinusoid that comprises the original waveform) and the First Fundamental Frequency.</p>
	<p>The blue function on the right shows the sum of the First, Second, and Third Fundamental Frequency</p>
	<p>Finally, adding the third and fourth fundamental frequencies give us the original waveform . These fundamental frequencies are all sinusoids.</p>

Table 1 - Elementary Intuition behind a Fourier Transform

In the early 1800s, Joseph Fourier proved that any arbitrary signal (a function of time) can be universally broken down into fundamental frequencies that are sinusoids. The proof for this is beyond the bounds of this essay, but the derivation of the formula to find fundamental frequencies will be covered.

2.2 The Fourier Series

The Fourier Transform can be derived using the Fourier Series, which decomposes only **periodic** functions into sinusoids.

A function $f(t)$ is periodic if it follows this convention:

$$f(t) = f(t + n T)$$

Where,

$$n = 1, 2, 3 \dots$$

T is the time period at which the function repeats itself ⁷

2.2.1 The Trigonometric (Real) Series

Let $P_E(t)$ be a periodic **even** function, with period T . It would be simpler to first find the Fourier series formula for just even and odd functions, and then generalize it to all functions.

Even functions are functions that are symmetric about $t=0$, so an even function $f_e(t) = f_e(-t)$

Odd functions are functions that are antisymmetric about $t=0$, so an odd function $f_o(t) = -f_o(-t)$

Even Functions – The Fourier Cos Series

Any even function can be represented by the sum of only cosines, as the cos function is an even function and we know that all functions can be represented as a sum of sinusoids. Therefore, we will form a **synthesis** equation of representing an arbitrary even function $P_e(t)$ — we will *synthesize* the function by adding up cosines. So, $P_e(t)$ can be represented as

⁷ <https://mathworld.wolfram.com/PeriodicFunction.html>

$$P_e(t) = \sum_{n=0}^{\infty} a_n \cos(n\omega_0 t) \quad (1)$$

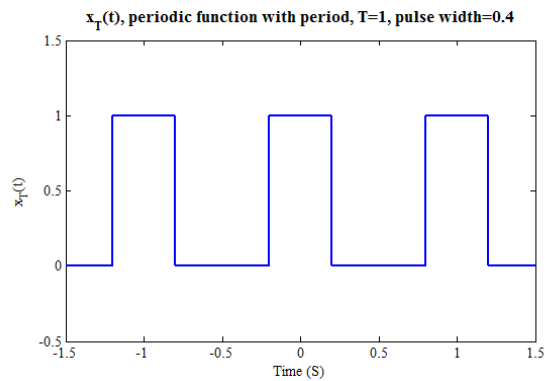
Where,

t is the time

ω_0 is the frequency, where the period $T = \frac{2\pi}{\omega_0}$

This equation essentially adds up cos waves of increasing frequency up to infinity to form the periodic function. Here is an example to illustrate how this *synthesis* process works.

Consider the following function $x_T(t)$, which has a period $T = 1$. Therefore, $\omega_0 = \frac{2\pi}{T} = 2\pi$.



Graph 1 - An example periodic function (frequency vs time)

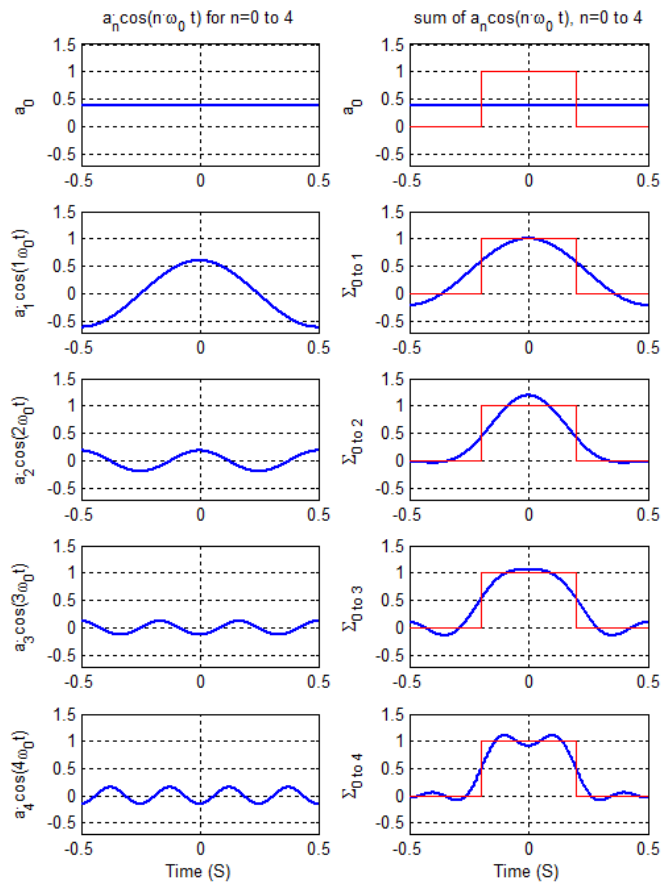


Figure 1 - Representing a periodic function as a sum of cosines

- The top graph in the left column is a_0 , a constant term. How the values of this coefficient a_n is calculated is shown later on.
 - The second graph shows $a_1(\cos(1\omega_0 t))$, where there is exactly one oscillation in the period $T = 1$ of $x_T(t)$. This is called the first harmonic.
 - The third graph shows 2 oscillations in the period $T=1$, the second harmonic.
 - This goes on and on for increasing values of n .
- The blue graph right column shows the cumulative sum as n increases; it's a representation of equation (1). AS n approaches infinity, you can see how the sum forms the function $x_T(t)$.

Finding the value of the coefficient a_n

To find a_n let's start with by multiplying both sides with $\cos(m\omega_0 t)$, where m is just an arbitrary integer variable. The exact reason behind the steps from here on to derive the Fourier Series equation will be done without justification, as the reason behind the choice of these steps is extremely complicated but understood remarkably well by Fourier.

$$P_e(t) \cos(m\omega_0 t) = \sum_{n=0}^{\infty} a_n \cos(n\omega_0 t) \cos(m\omega_0 t)$$

Now, let's integrate both sides over any one period interval (from 0 to T).

$$\int_0^T P_e(t) \cos(m\omega_0 t) dt = \int_0^T \sum_{n=0}^{\infty} a_n \cos(n\omega_0 t) \cos(m\omega_0 t) dt$$

We can switch the order of the integral and summation:

$$\int_0^T P_e(t) \cos(m\omega_0 t) dt = \sum_{n=0}^{\infty} a_n \int_0^T \cos(n\omega_0 t) \cos(m\omega_0 t) dt$$

Now, on the right-hand side we can use the identity $\cos(a) \cos(b) = \frac{1}{2} \cos(a+b) + \cos(a-b)$

$$\begin{aligned} \int_0^T P_e(t) \cos(m\omega_0 t) dt &= \sum_{n=0}^{\infty} a_n \int_0^T \frac{1}{2} (\cos((m+n)\omega_0 t) + \cos((m-n)\omega_0 t)) dt \\ &= \frac{1}{2} \sum_{n=0}^{\infty} a_n \int_0^T (\cos((m+n)\omega_0 t) + \cos((m-n)\omega_0 t)) dt \end{aligned}$$

Assume $m > 0$ (we will look at $m = 0$ later on). Since $(m+n)$ is a positive integer, $\cos((m+n)\omega_0 t)$ has an integer number of oscillations in one period T . If we find the area under the curve (integrate) in this period, it will be always be 0, as illustrated in the diagram below.

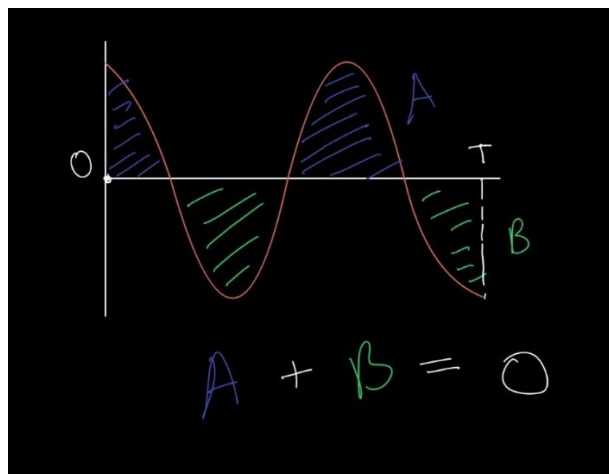


Image 1 - why the cos integral simplifies to 0

The graph shows a cos function — since area $B = -A$, they cancel out and the integral simplifies to 0.

This would be true for any integer number of oscillations. So, $\int_T^0 (\cos((m+n)\omega_0 t)) dt = 0$. Therefore, we have:

$$\begin{aligned}\int_0^T P_e(t) \cos(m\omega_0 t) &= \frac{1}{2} \sum_{n=0}^{\infty} a_n \int_0^T (\cos((m+n)\omega_0 t) + \cos((m-n)\omega_0 t)) dt \\ &= \frac{1}{2} \sum_{n=0}^{\infty} a_n \left(\int_0^T (\cos((m+n)\omega_0 t)) dt + \int_0^T (\cos((m-n)\omega_0 t)) dt \right) \\ &= \frac{1}{2} \sum_{n=0}^{\infty} a_n \left(\int_0^T (\cos((m-n)\omega_0 t)) dt \right)\end{aligned}$$

When $m \neq n$, $m - n$ is a non-zero integer.

When $m - n$ is positive, the term $\int_0^T (\cos((m-n)\omega_0 t)) dt = 0$ for the same reasons described above: the function has an integer number of oscillations. When $m - n$ is negative, the integral simplifies to 0 as cosine is an even function and $\cos(-\theta) = \cos(\theta)$. There will still be an integer number of oscillations.

However, when $m = n$, $\cos((m-n)\omega_0 t) = \cos(0) = 1$. So,

$$\int_0^T (\cos((m-n)\omega_0 t)) dt = \begin{cases} \int_0^T (\cos((m-n)\omega_0 t)) dt = 0, & m \neq n \\ \int_0^T 1 dt = T, & m = n \end{cases}$$

Let's look at our equation once again.

$$\int_0^T P_e(t) \cos(m\omega_0 t) = \frac{1}{2} \sum_{n=0}^{\infty} a_n \left(\int_0^T (\cos((m-n)\omega_0 t)) dt \right)$$

As n goes from 0 to infinity, every term of the summation would yield 0 except $m = n$ — where the summation term is $a_m T$. So, the summation would simply become:

$$\int_0^T P_e(t) \cos(m\omega_0 t) dt = \frac{1}{2} a_m T$$

$$\therefore a_m = \frac{2}{T} \int_0^T P_e(t) \cos(m\omega_0 t) dt$$

Here, since m is just any arbitrary variable for a positive integer, we can swap it with n .

$$a_n = \frac{2}{T} \int_0^T P_e(t) \cos(n\omega_0 t) dt \quad (2)$$

We still need to consider the case $m = 0$. For this case:

$$a_0 = \frac{2}{T} \int_0^T P_e(t) \cos(n\omega_0 t) dt$$

$$\int_0^T P_e(t) \cos(m\omega_0 t) dt = \sum_{n=0}^{\infty} a_n \int_0^T \cos(n\omega_0 t) \cos(m\omega_0 t) dt$$

$$\int_0^T P_e(t) \cos((0)\omega_0 t) dt = \sum_{n=0}^{\infty} a_n \int_0^T \cos(n\omega_0 t) \cos((0)\omega_0 t) dt$$

$$\rightarrow (\cos(0) = 1)$$

$$\int_0^T P_e(t) dt = \sum_{n=0}^{\infty} a_n \int_0^T \cos(n\omega_0 t) dt$$

Once again, the integral on the right-hand side simplifies to 0 as there is an integer number of oscillations except when $n = 0$. So, $n = 0$ is the only term that contributes to the summation. Therefore,

$$\int_0^T P_e(t) dt = a_0 \int_0^T \cos(0) \omega_0 t dt$$

$$\int_0^T P_e(t) dt = a_0 T$$

$$a_0 = \frac{1}{T} \int_0^T P_e(t) dt \quad (3)$$

Odd Functions – The Fourier Sine Series

Similar to even functions and equation (1), we can define any odd function $P_o(t)$ as a sum of sines with increasing frequency, as sine is an odd function.

$$P_o(t) = \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t) \quad (4)$$

We start the summation from $n=1$ as the $n=0$ term just becomes $b_0 \sin(0) = 0$.

Following an extremely similar derivation as a_n , b_n can be represented by:

$$b_n = \frac{2}{T} \int_0^T P_o(t) \sin(n\omega_0 t) dt \quad (5)$$

General Functions

Given a function $P(t)$, we can split the function into its odd and even parts.

$$P_o(t) = \frac{1}{2}(x(t) - x(-t))$$

$$P_e(t) = \frac{1}{2}(x(t) + x(-t))$$

We can clearly see that both $P_o(t) = -P_o(t)$ and $P_e(t) = P_e(-t)$ (they follow odd and even function conditions). When added together, they create the original function:

$$P_e(t) + P_o(t) = P(t)$$

Using this and the Sine and Cosine Fourier series, we can create the Trigonometric equation for any function $P(t)$:

$$P(t) = \sum_{n=0}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t)$$

We can take out the $n=0$ term from the first summation — we will get $a_0 \cos(0) = a_0$.

$$P(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t)$$

The coefficients a_n and b_n are shown by these equations:

$$a_0 = \frac{1}{T} \int_0^T P(t) dt$$

$$a_n = \frac{2}{T} \int_0^T P(t) \cos(n\omega_0 t) dt, \quad n \neq 0$$

$$b_n = \frac{2}{T} \int_0^T P(t) \sin(n\omega_0 t) dt$$

2.2.2 The Exponential (Complex) Series

The Trigonometric form only applies to real numbers, so we will now extend the trigonometric form to a complex domain.

To achieve this, we will use the Euler's formula:

$$e^{j\omega_0 t} = \cos\omega_0 t + j\sin\omega_0 t$$

Where,

j is the imaginary number, $\sqrt{-1}$

ω_0 is the frequency term

t is the time

The Euler's formula can be easily visualized by a unit circle in the complex plane:

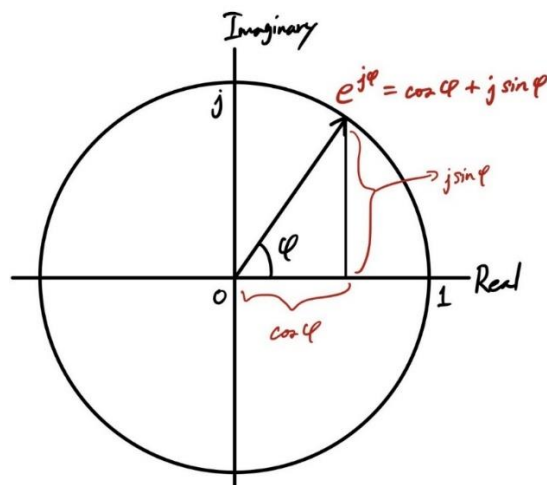


Figure 2 - The Unit Circle and Euler's Formula (Nakagome, Fourier Transform 101 — Part 2: Complex Fourier Series)

The Complex Fourier Series is as follows:

$$P(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \quad (6)$$

Where the coefficient c_n is defined as:

$$c_n = \frac{1}{T} \int_0^T P(t) e^{-jn\omega_0 t} dt \quad (7)$$

The derivation of this closely follows that of the trigonometric series, so I would not be deriving it from the start. **Instead, we will prove that this equation is equivalent to the trigonometric series equation.**

This form's notation is similar to the Fourier transform, so it will help us derive that.

We need to prove that:

$$a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

First, we can consider the case $n = 0$

$$a_0 = c_0 e^0$$

$$a_0 = c_0 \quad (8)$$

Next, we'll separate the summation into 3 ranges:

$$\begin{aligned} P(t) &= \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} = \sum_{n=-\infty}^{-1} c_n e^{jn\omega_0 t} + c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} \\ &= \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega_0 t} + c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} \quad (9) \end{aligned}$$

Now we will find the coefficient c_n in terms of a and b

$$\begin{aligned} c_n &= \frac{1}{T} \int_0^T P(t) e^{-jn\omega_0 t} dt = \frac{1}{T} \int_0^T P(t) (\cos(n\omega_0 t) - j \sin(n\omega_0 t)) dt \quad (\text{Euler's Formula}) \\ &= \frac{1}{T} \left(\int_0^T P(t) (\cos(n\omega_0 t)) dt - j \int_0^T P(t) (\sin(n\omega_0 t)) dt \right) \end{aligned}$$

$$= \frac{1}{2} [a_n - jb_n]$$

In the last step, a_n and b_n from the trigonometric Fourier Series was substituted.

Similarly, we have the coefficient c_{-n} as:

$$c_{-n} = \frac{1}{2} [a_n + jb_n] \quad (10)$$

From the Euler's formula, (8), (9), and (10) we have:

$$\begin{aligned} P(t) &= \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega_0 t} + c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} \\ &= \sum_{n=1}^{\infty} \frac{1}{2} ((a_n + jb_n)(\cos(n\omega_0 t) - j \sin(n\omega_0 t)) + c_0 \\ &\quad + \sum_{n=1}^{\infty} \frac{1}{2} ((a_n - jb_n)(\cos(n\omega_0 t) + j \sin(n\omega_0 t)) \end{aligned}$$

Expanding and simplifying:

$$\begin{aligned} &= \sum_{n=1}^{\infty} \frac{1}{2} (a_n \cos(n\omega_0 t) - ja_n \sin(n\omega_0 t) + jb_n \cos(n\omega_0 t) - j^2 b_n \sin(n\omega_0 t)) + c_0 \\ &\quad + \sum_{n=1}^{\infty} \frac{1}{2} (a_n \cos(n\omega_0 t) + ja_n \sin(n\omega_0 t) - jb_n \cos(n\omega_0 t) - j^2 b_n \sin(n\omega_0 t)) \end{aligned}$$

We know that $j^2 = -1$ (imaginary numbers), and combining the summations, we get:

$$\begin{aligned} &= \sum_{n=1}^{\infty} \frac{1}{2} (a_n \cos(n\omega_0 t) \\ &\quad - ja_n \sin(n\omega_0 t) + jb_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t) + a_n \cos(n\omega_0 t) \\ &\quad + ja_n \sin(n\omega_0 t) - jb_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) + c_0 \\ &= \sum_{n=1}^{\infty} \frac{1}{2} (2 a_n \cos(n\omega_0 t) + 2 b_n \sin(n\omega_0 t)) + c_0 \end{aligned}$$

$$= \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) + c_0$$

$$\rightarrow c_0 = a_0$$

$$= a_0 + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t))$$

This $P(t)$ is exactly the same as the Trigonometric Fourier Series. Hence, we proved that the Exponential Fourier Series is equivalent, but in the Complex form. The coefficients a_n and b_n become one coefficient c_n .

2.3 Deriving the Fourier Transform from Fourier Series

To derive the Fourier Transform, we will use the Complex Fourier Series (as it is easier to deal with one coefficient, c , instead of two), and apply it to aperiodic signals. Here is a graphic to visualize this:

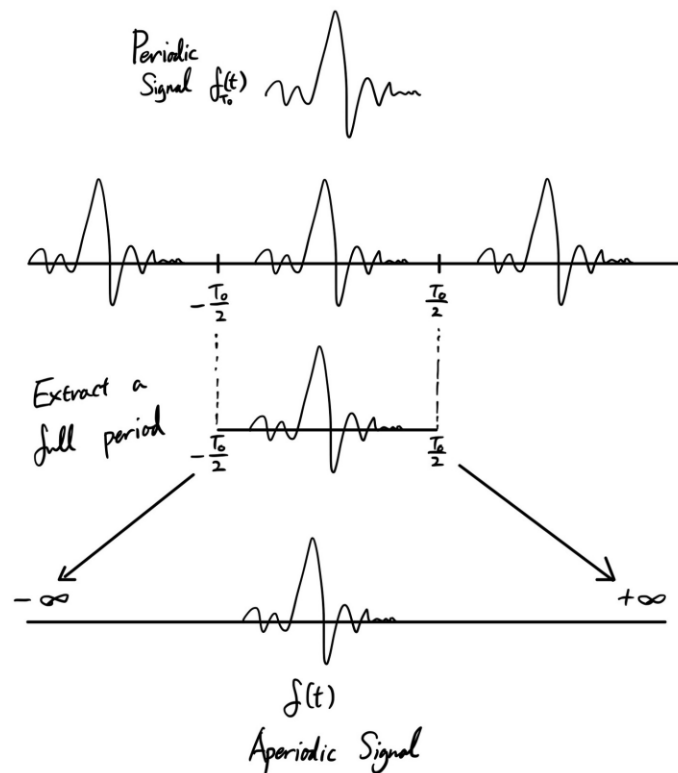


Figure 3 - Extending a Fourier Series to non-periodic functions (Nakagome, Fourier Transform 101 — Part 3: Fourier Transform)

We are taking out the wave function in one period and stretching it out to infinity. This would make a wave that is clearly not repeating itself as it's created from one period. We will apply this idea to derive the Fourier Transform equation. Extending only a single period allows us to include aperiodic functions as well.

Let's start with equation (7), but we'll write $P(t)$ as $f(t)$:

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-jn\omega_0 t} dt$$

Let's also take the period $(-\frac{T}{2}, \frac{T}{2})$ instead of $(0, T)$, because as mentioned before the exact period does not matter. We want to stretch the waveform from $-\infty$ to ∞ . So, we can rewrite it as:

$$Tc_n = \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt$$

Now, let's extend the limit of the period ($T \rightarrow \infty$) to infinity as shown in the image.

$$Tc_n = \lim_{T \rightarrow \infty} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt$$

As $T \rightarrow \infty$, the frequency term $\omega_0 = \frac{2\pi}{T}$, becomes extremely small and, as you can recall from the summation in the Exponential Fourier Series, n has a range of $\pm\infty$. Therefore, the term $n\omega_0$ can now take on any value, so we will rewrite it as $\omega = n\omega_0$. We then get:

$$Tc_n = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

We will also rewrite Tc_n as a function, and this will be a function of ω , which is the frequency.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (11)$$

This is it. The equation for the Fourier Transform. **Here we can see that a function of time $f(t)$ is converted into a function of frequency $F(\omega)$ — one of the main characteristics of the Fourier Transform.**

The frequency domain would better help us give information about emotion compared to a time-amplitude domain waveform, as emotion in our speech is largely based on variations in pitch.

3. Retrieving Emotion from Audio

Sound Waves

Sound is the compressions and rarefactions, regions of high and low pressure, propagated through a medium, such as air. These compressions and rarefactions travel as a wave. We create our sounds from the Glottal Pulse, which is the folds of the vocal cords as we are speaking. The Glottal Pulse moves its way through the Vocal Tract — a system of organs in our mouth and throat. The Vocal Tract includes the nasal cavity, tongue, teeth and more — so most of the meaning in the sound is created in the Vocal Tract. Our goal would be to get the data of the Glottal Pulse moving through the Vocal Tract in tangible format, so we can analyze it and break it down.

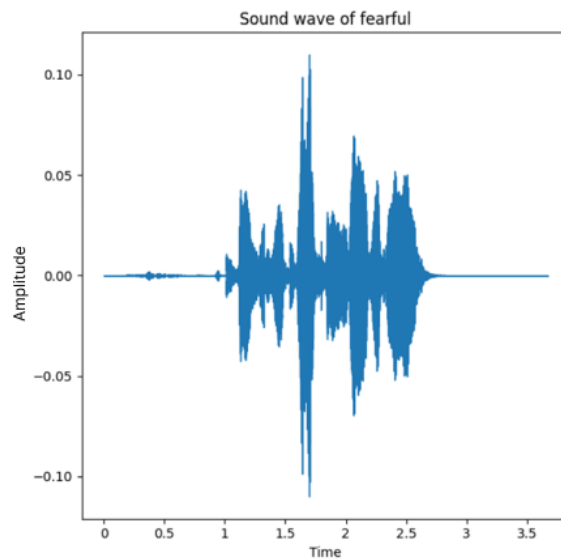
Analyzing an Audio Sample

Now, we will use a sample audio recording from the RAVDESS dataset⁸ (containing 12 male and 12 female voice actors), where a voice actor vocalizes two lexically-matched statements in different emotions. I will be using Python⁹ to process this audio recording. The processing includes Fourier Transforms along with other processing such as scaling which is explained further as I break down each step of the processing.

⁸ <https://smartlaboratory.org/ravdess/>

⁹ <https://www.python.org/>

First, we have the original audio waveform of one of the audio recordings in a **fearful voice** from the dataset:



Graph 2 - Sound wave of the fearful voice, Amplitude vs Time (seconds)

Here, the y-axis is the amplitude, and the x-axis is the time in seconds. The blue wave shows the waveform in the time domain.

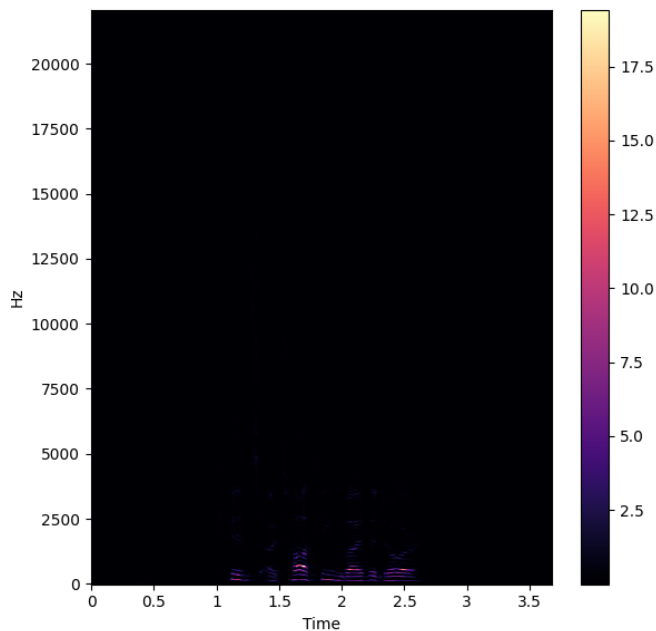
Now we will apply a special type of Fourier Transform to this waveform called the Short-Time Fourier Transform (STFT). This applies the Fourier Transform multiple times across small, overlapping time intervals. This way, we can also save the time information in every time interval window (while applying Fourier transforms), to achieve a 3-dimensional *spectrogram*.

Short-time Fourier Transform Function

```
D = np.abs(librosa.stft(data))  
librosa.display.specshow(D, sr=sampling_rate, x_axis='time', y_axis='linear');  
plt.colorbar()  
plt.show()
```

Image 2- A snippet of the code used, applying STFT to the audio wave data

I am applying this Fourier Transform using a library called Librosa¹⁰, which has an inbuilt function called “stft” (Short-Time Fourier Transform), which applies the Fourier transform. This function is given $f(t)$ (a function of time), and returns $F(\omega)$ (a function of frequency and amplitude) alongside an array of *time* values (from to the short-time windows). Matplotlib¹¹ was used plot these values as a spectrum, as shown in the code snippet above.

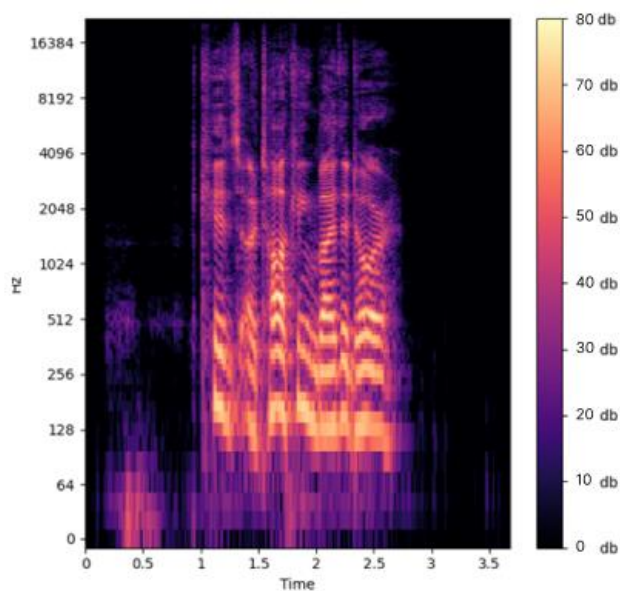


Graph 3 - A color-plot of the spectrogram retrieved from applying STFT

¹⁰ <https://librosa.org>

¹¹ <https://matplotlib.org/>

The color-plot above represents the audio waveform in the frequency-time domain, where the frequency is in Hertz (Hz) and Time is in seconds (s). Along with this, the amplitude shown by the color at each point. This plot does not give us too much information – you can only see specs of color – because the human hearing and speaking frequency range is much smaller. To fix this, we will apply a log-scale to our amplitude data, converting it into a unit known as *decibels (Db)*. This essentially “zoom” into the graph so that we can the representation is more detailed.



Graph 4 - A log-scaled spectrogram, with the amplitude in Decibels (Db)

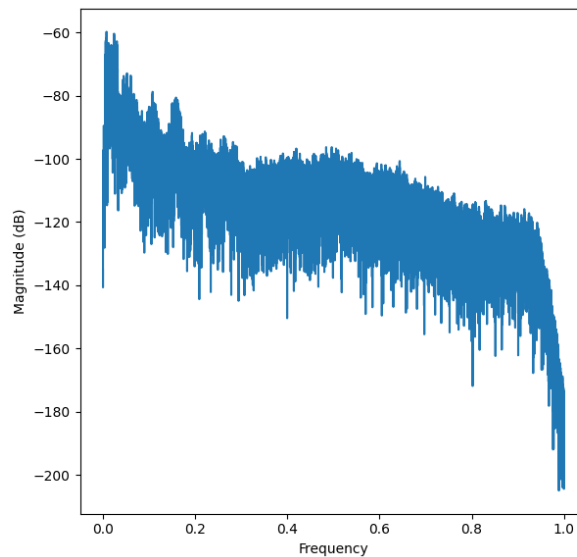
```
37 DB = librosa.amplitude_to_db(D, ref=np.max)
38 librosa.display.specshow(DB, sr=sampling_rate, x_axis='time', y_axis='log');
39 plt.colorbar(format='%+2.0f db')
40 plt.show()
41
42 plt.magnitude_spectrum(data, scale='dB')
43 plt.show()
```

Image 3- A snippet of the code used to apply log-scaling to the previous spectrum

We can see that there is much more value in this information as we can see the decibels of different frequencies over time. The spectrograms themselves have been proven to be incredibly useful to classify

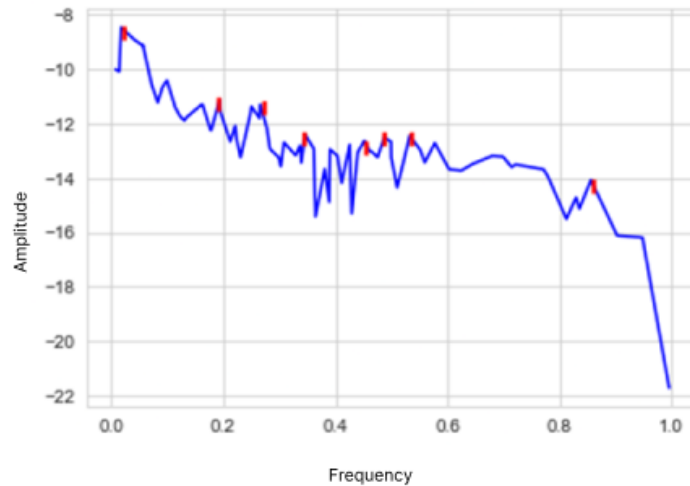
emotions from due to its heavily descriptive nature, but in this essay, we will dive deeper into the wave so that the emotion data is clear.

Now, let's consider the spectrum of our sound wave in the frequency-amplitude domain – the *spectrum*.



*Graph 5- Log-Amplitude scaled graph of Frequency vs Amplitude (Decibels) of the **same fearful emotion recording***

This is achieved by applying a normal Fourier Transform to our data, and then a log-scale transformation into decibels. This would be a good representation of the Glottal Pulse moving through the vocal tract, as we have the speech wave information. Now, consider a smoothened (moving average) version of this graph:



Graph 6- The Spectral Envelope of the *same fearful emotion recording*

This is called the *spectral envelope*. This allows us to look at the peaks of the frequency-amplitude plot, known as *Formants*. These peaks (shown by the red strokes in the image above) would be an accurate representation of the Vocal tract, as **it clearly displays the variations in the Glottal Pulse** (by organs such as the tongue).

In the next part, we train a machine learning model to understand how these formants look like in different emotions. Then, the model can be applied to sound waves in general and we will try to see if it can understand emotion by just by looking at the Formants.

Feature Extraction Using Machine Learning

We have recognized our vocal tract, but we need a way to extract them from the speech. This is where Mel-frequency cepstral coefficients (MFCCs) come in. From our log scaled spectrum, we will apply Mel-Scale, a logarithmic-based scale that converts frequencies (Hertz) into perceptually relevant unit known as Mels--taking into account the human ears ability to recognize changes in frequencies only more than 500 Hz, essentially creating a spectrum of a spectrum. Finally, a Discrete Cosine Transform will be applied. Discrete Cosine Transforms are simplified Discrete Fourier Transforms, that express signals as a sum of

only cosines. The advantage is that these are faster in processing and give real value coefficients, making them easier to handle.

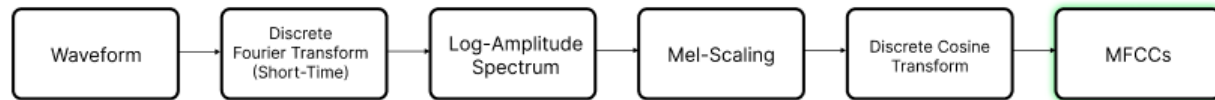
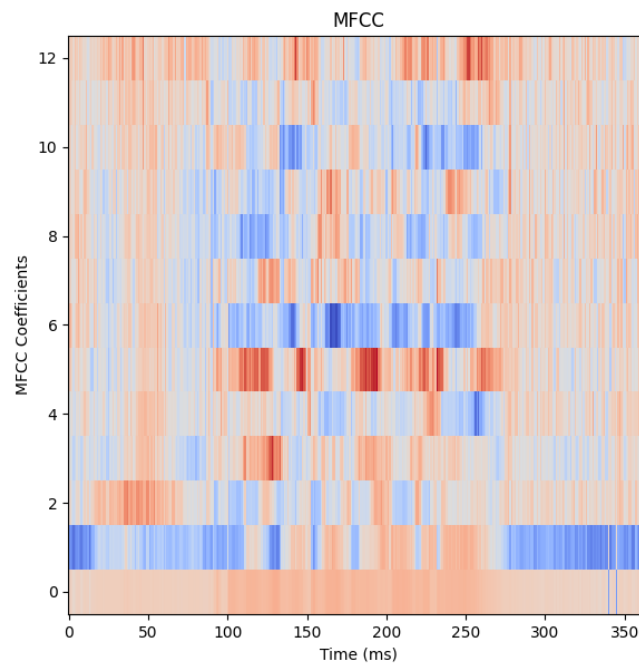


Figure 4 - A flowchart of the signal-processing done before training the machine learning model

From the discrete Cosine Transformation, we will receive different values of coefficients for each base cosine function, and these coefficients are called Mel-Frequency Cepstral Coefficients. MFCCs are helpful in separating vocal tract, and these MFCCs were given to a Multi-Layer Perceptron (MLP) machine learning model. The MFCCs of an audio signal can be represented as such:



*Graph 7- A Spectrogram showing the MFCCs in the same **fearful** audio waves*

This was achieved using the Mel-scaling and discrete Cosine transform from our previously shown spectrum. This graph can be thought of as a matrix, where the columns in the graph area represent the

chunks of audio (or the audio sampling rate), and the rows represent the different coefficients of the cosine functions, in this case we have 12 coefficients, where the warmth/coolness represents the value of the coefficients. The particular value of this is not shown in the graph, but rather stored as arrays. These MFCCs were computer for a set of samples of different emotions (neutral, happy, angry and disgust) (and different voice actors) in the dataset, which were given to a Multi-Layer Perceptron (MLP) Model, a type of an Artificial Neural Network.

```
[+] Number of training samples: 504 504
[+] Number of testing samples: 168 168
[*] Training the model...
Accuracy: 74.40%
```

Image 4 - Overall accuracy 74.40% was achieved with the MLP model used

504 of the voice samples' MFCCs were given to train the model, and then 168 different samples were tested for prediction. **74.4% of these model's predictions matched the true emotion of the voice.**

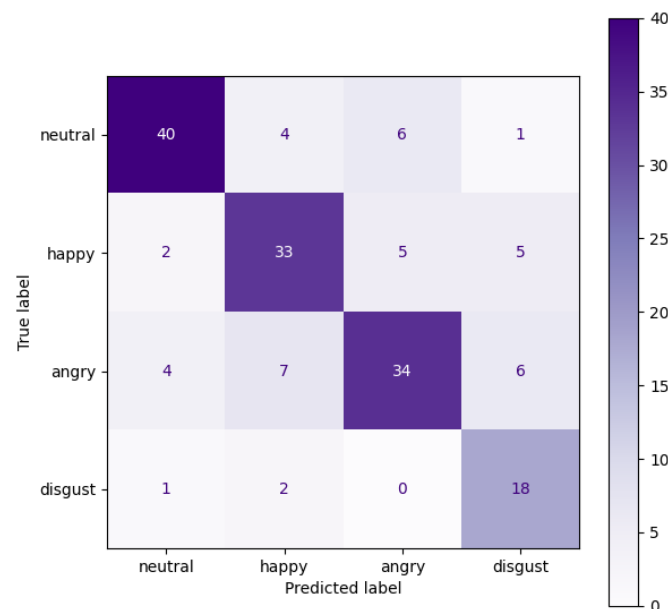


Figure 5 - A Confusion Matrix showing the accuracy and distribution of the model's prediction

The Confusion Matrix above shows the distribution of the model's prediction, by plotting the true emotion of the voice sample (y-axis) against the predicted emotion (x-axis). For example, the top left square in this matrix shows us that for 40 samples, a neutral voice was predicted correctly as neutral. In this way, all 168 tested audio samples were plotted in the matrix. The high color-bar values help us visualize that when the predicted label is very often matched correctly with the actual label.

Thus, this shows that the model was fairly successful in predicting the voice, as seen in the matrix it usually matched the true label. There are certain inaccuracies, such as with the emotion of disgust, but they can be fine-tuned with better models, as discussed later in Further Scope.

4. Conclusion

4.1 Summary

To summarize, Fourier Transforms *can* be used along with Machine Learning to decode emotions from speech recordings. Through mathematical analysis, we broke down audio waves using Fourier Transforms (a detailed derivation of which was shown) and some further processing depicted in the flowchart in Figure 3.

4.2 Assumptions and Limitations

Only 4 emotions (neutral, happy, angry, fear) were evaluated in this Extended Essay, but this could be extended to more emotions (such as disgust or sadness). The assumption is that these emotions also have considerably distinct formants that the machine learning model can accurately distinguish between. The model has to be more precise so that it can distinguish variations in the MFCCs that might be not so distinct. To achieve such a precise model, perhaps more feature extraction methods (that show the

identity of the sound) could be used with along with MFCCs. These feature extraction methods, like evaluated in this paper¹², weren't focused in this Extended Essay, however could be further considered.

4.3 Further Scope

The accuracy of this model is still questionable, and might need to be improved in order to have real world application. More voice samples and datasets could be included to increase the accuracy of the model, and a different machine learning model may be used that may be better suited for this approach to increase the accuracy. Moreover, adding a wider range of emotions to recognize from would be helpful.

The potential applications of these emotion recognition systems are vast, like analyzing phone calls as feedback for customer service.

A paper showed that such speech emotion recognition systems can be applied to children with autism spectrum disorder, for others to understand prosody of their voices when it is difficult for children with ASD to verbally communicate their emotion. The emotion in the voices of children with ASD can be recognized by the algorithm, and then can be displayed to people trying to understand their emotion. Additionally, speech recognition systems can be presented in the forms of quizzes, where children with ASD are trained to recognize emotion better by trying to guess the emotion of a particular voice recording. Recognizing and conveying emotions, being such an integral part, may even improve their overall quality of life.

¹² <https://www.researchgate.net/publication/337919098>

Bibliography

Baram, Tal. *Classifying emotions using audio recordings and Python*. n.d.

<<https://towardsdatascience.com/classifying-emotions-using-audio-recordings-and-python-434e748a95eb>>.

blackpenredpen. *Fourier Series Coefficients*. n.d. <<https://www.youtube.com/watch?v=iSw2xFhMRN0>>.

Cheever, Erik. *Derivation of Fourier Series*. n.d.

<<https://lpsa.swarthmore.edu/Fourier/Series/DerFS.html>>.

—. *Introduction to the Fourier Transform*. n.d.

<<https://lpsa.swarthmore.edu/Fourier/Xforms/FXformIntro.html>>.

Nakagome, Sho. *Fourier Transform 101 — Part 1: Real Fourier Series*. n.d. <<https://medium.com/sho-jp/fourier-transform-101-part-1-b69ea3cb4837>>.

—. *Fourier Transform 101 — Part 2: Complex Fourier Series*. n.d. <<https://medium.com/sho-jp/fourier-transform-101-part-2-complex-fourier-series-934a885b3921>>.

—. *Fourier Transform 101 — Part 3: Fourier Transform*. n.d. <<https://medium.com/sho-jp/fourier-transform-101-part-3-fourier-transform-6def0bd2ca9b>>.

—. *Fourier Transform 101 — Part 4: Discrete Fourier Transform*. n.d. <<https://medium.com/sho-jp/fourier-transform-101-part-4-discrete-fourier-transform-8fc3fbb763f3>>.

The Fourier Transform. 5 April 2022. <thefouriertransform.com>.

Velarado, Valerio. *Mel-Frequency Cepstral Coefficients Explained Easily*. n.d.

<https://www.youtube.com/watch?v=4_SH2nfbQZ8&t=1357s>.

