360-420-DW Introduction to Computer Programming in Engineering and Science

Section 0001

Oriane Roy and Mariyana Slavova

May 17th 2019

Sound Decomposition (FFTs)

For our term project, we decided to focus on sound decomposition. More precisely, our goal was to create a program that would enable the user to obtain the harmonics associated with a specific sound. We chose this subject because both of us are musicians (Mariyana plays the saxophone and Oriane, the piano), and therefore we were particularly interested in bettering our understanding of the physics behind hearing. Decomposing a sound using Java can be obtained by applying what are known as Fast Fourier Transforms (FFTs) - the main concept behind our project. FFTs allow us to "untangle" the different components of a signal and to obtain a "Power vs Frequency" graph - graph on which we should expect sharp peaks at specific frequencies, ie the music note. Validating that our code works and that the specific frequencies are the ones wanted was done by testing it with a pure sound (one for which the frequency is already known).

A Bit about Hearing…

For our brain to perceive a noise, a sound wave has to travel through our ear canal *(see Figure 1)* where it hits the eardrum, which vibrates and transmits that vibration to the ossicles (the malleus, the incus and the stapes). These three tiny bones intensify the sound and send it into the cochlea where it is transformed into the electrical impulses that are sent to the brain.

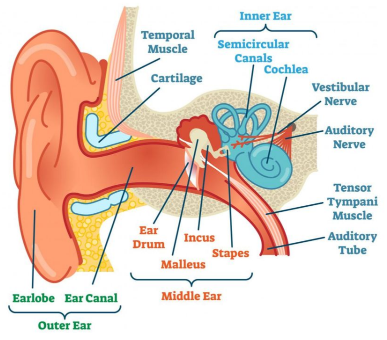

Figure 1: Ear diagram

Description of Model

The initial sound wave travelling through our ear canal does not correspond to a perfect sinusoidal function. Rather, it is composed of multiple individual sinusoidal functions that correspond to the sound wave when added together. For example, one can see on *Figure 2* that the sound wave (in black) is equal to the sum of the red wave, the blue wave, the green wave, the pink wave and the yellow wave added together. Every single one of those individual sine waves has its own frequency, and the goal of our algorithm was to determine what those frequencies were (for specific sounds of our choice). To obtain the individual frequencies of the sine functions that compose a sound wave, we need to apply a Fourier Transform in order to obtain a "Power vs Frequency" graph such as the one shown on *Figure 3*, where the peaks correspond to the specific frequencies we are looking for.

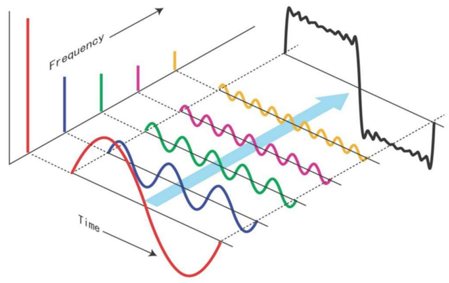

Figure 2: How a sound wave *(in black)* is composed of individual sinusoidal waves added together

To do so, we need to calculate the Fast Fourier Transform, which is simply a faster algorithm than the Discrete Fourier Transform. The Discrete Fourier Transform is actually represented by the formula

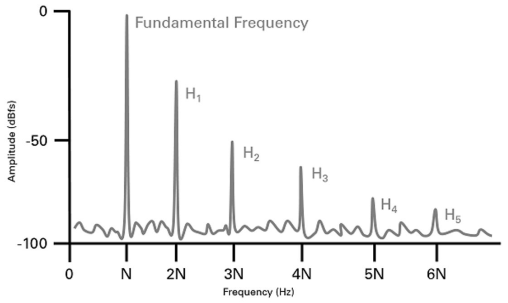

Figure 3: The different harmonics that compose a sound are associated with the frequency peaks on a power vs frequency graph

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \qquad k = 0, \dots, N-1, \tag{1}$$

and involves some complex numbers. That is because the fourier transforms rely on the idea of winding up a wave around itself *(see Figure 4)*, and complex numbers deal well with winding and rotation. (3Blue1Brown)

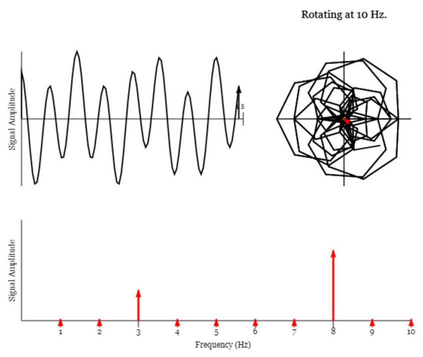

Figure 4: Wave as a function of time *(top left corner)*, winding up a wave around itself *(top right corner)* and the corresponding "Power vs Frequency graph *(bottom)*

A sound can be explained as being one single signal with *n* number of points. Since a sound is a continuous signal, we need a way to accurately separate it into a finite number of samples. More precisely, according to the sampling theory, our number of samples should be at least twice the frequency of the highest frequency in the signal.The Nyquist frequency, which is *N/2*, *N* being the sampling frequency, is the minimum sampling frequency that we need if we want to digitize a sound as well as the theoretical maximum that can be determined by the FFT. If the sampling theory is respected, it is possible to reconstruct the original continuous sound by simply adding up the components we have found. (Bourke)

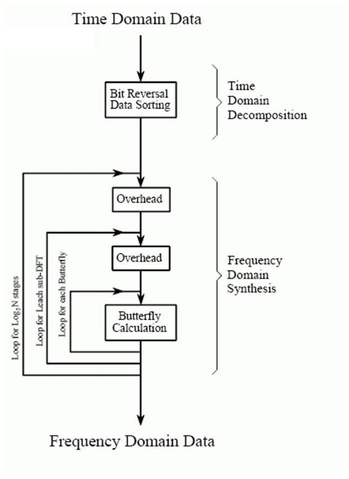

Figure 5: Flow diagram of the FFT

What we want to do is separate a single signal into *n* number of signals so that each signal only contains one point. The Fourier transform uses imaginary numbers, denoted by *i*. That is because Fourier transforms use complex notation where each point in the time and frequency domains are composed of a real part and of an imaginary part. Decomposition is made by separating the signal in two halves: its even and then its odd numbered samples. This step is repeated with each of the 2 signals that have been separated and so on, until there are *n* number of signals. This explains why it is optimal to have a number of samples that is an exponential with 2 as the base. With the Fast Fourier Transform, the computational time is proportional to $N\log_2(N)$ *(2)*. This method is called 'bit reversal' *(see figure 5)*. Then, we need to calculate the frequency step which is simply defined by 1/N. Finally, the end result, so the output of this function, is a set of numbers that represent the power of the amplitudes at each frequency step, which makes it possible for us to graph the sound wave according to the frequency. (Smith)

Description of Computational Method

First, we had to enable the user to select which sound file to use (note that we were working with WAV sounds). To do so, we used the JFileChooser class and called a method we created - getFile - that opens up our file directory which enables us to directly choose the file we want. We then need to decode the sound wave and put it into a buffer. Then, we calculate the time step and the Nyquist frequency which will be the x-axis for, respectfully, the time domain graph and the frequency domain graph. Since the program gives us the exact number of samples of the sound, we need to make sure that the number of samples is of base 2. To do so, we pad the signal, adding zeros until we reach the next highest power of 2 closest to the initial number of samples. We downloaded a library (Apache Commons Math 3.6.1) that calculates the complex numbers and that applies the FFT for us. By creating an FFT object in our program, we can forward transform the signal with the use of the complex numbers and transforms classes. Now, with the outputs of the real and imaginary numbers that these classes give us, we can calculate the power. All that is left is to graph the sound with the help of JMath plots.

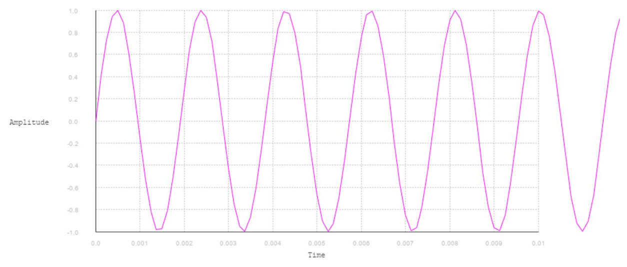

Figure 6: Amplitude vs Time for 523 Hz

Results

We made sure our code worked by running it with a pure tone (one composed of a single sine wave) of a known frequency: 523 Hz. As one can see from

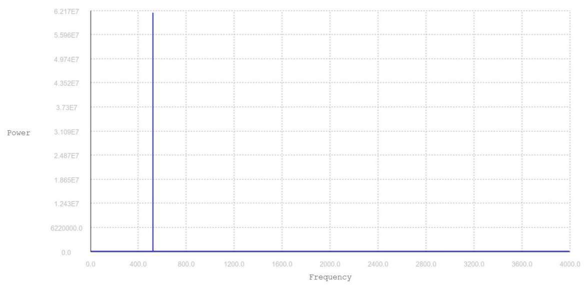

Figure 7: Power vs Frequency for 523 Hz

Figure 6, it corresponds to a perfect sine wave as expected. After having performed a FFT on it, we obtained the graph on *Figure 7*. There is an obvious peak at 523 Hz, which confirms that our algorithm works.

Now that our code was ready, we were able to start running it with the specific sounds we were interested in. We used the sound of a C5 being played on a trumpet and obtained the graphs shown in *Figure 8*. As one can see on the power vs frequency graph, there are specific peaks at certain frequencies: the first peak is at 523 Hz and the highest peak is at 1042 Hz.

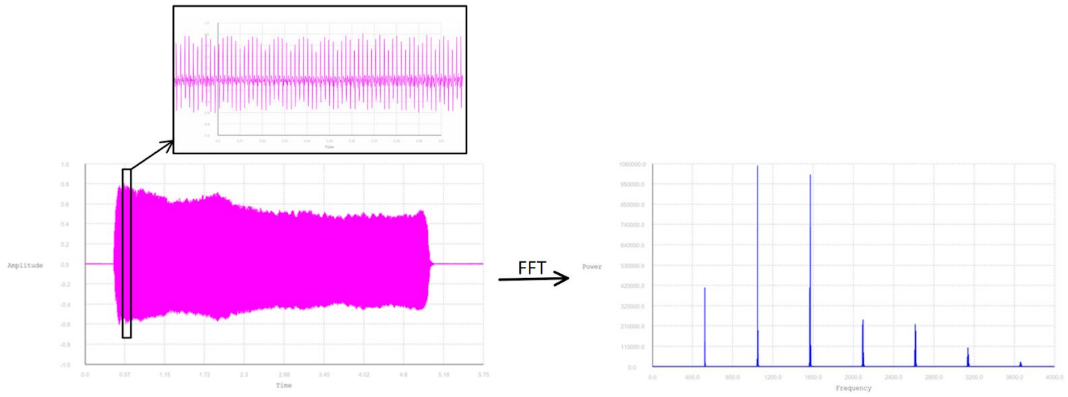

Figure 8: Amplitude vs Time for a Trumpet sound (C5) *(right)* and Power vs Frequency graph for the same sound *(left)*

We also tested it with a C2 being played on a cello and obtained the graph shown on *Figure 9*. The first peak was at a frequency of 65 Hz.

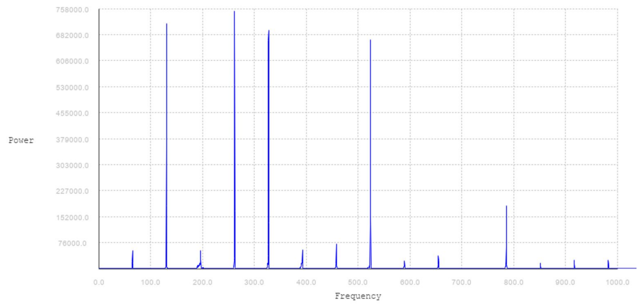

Figure 9: Power vs Frequency graph for cello sound (C2)

Discussion

When we hear a sound, what we recognize as the pitch of a musical tone is called the fundamental frequency. (Heeger) As explained above, a musical sound can be decomposed into individual sine waves with respective frequencies, and the fundamental frequency corresponds to the lowest one. Indeed, the *lowest frequency* is the one we perceive as being the *loudest*. For a

C5 played on a trumpet, the first frequency obtained was of 523 Hz, which indeed corresponds to a C5 according to the literature. For a C2 played on a cello, we obtained, as the first frequency, 65 Hz, which indeed is the frequency associated with a C2. All the other frequencies that we can see on those graphs and their varying powers are what gives music notes their timbre and how we can differentiate instruments from each other. What is interesting about studying harmonics is that it is possible to play a harmonic note on an instrument without the presence of its fundamental frequency. In wind instruments, it is done by 'overblowing' and in string instruments, it can be done by pressing lightly on the nodes of an open string. ("The Music Theory…")

Fourier transforms and sound decomposition have very relevant applications in "real life". In fact, they enable us to remove unwanted sounds from a recording by removing or modifying the frequencies obtained in the frequency domain graph. After, to obtain the desired sound, we must simply perform a reverse FFT.

References

3Blue1Brown. "But What Is the Fourier Transform? A Visual Introduction." YouTube, YouTube, 26 Jan. 2018, www.youtube.com/watch?v=spUNpyF58BY.

Bourke, Paul. "DFT & FFT." Paul Bourke, paulbourke.net/miscellaneous/dft/.

Connelly, Bill. "Visualizing How FFTs Work." Bill Connelly, www.billconnelly.net/?p=276.

"Fast Fourier Transform." Wikipedia, Wikimedia Foundation, 9 May 2019, en.wikipedia.org/wiki/Fast_Fourier_transform.

“Fundamental Frequency.” Wikipedia, Wikimedia Foundation, 20 Mar. 2019, en.wikipedia.org/wiki/Fundamental_frequency.

Heeger, David. “Harmonics Vs. Formants.” VoiceScienceWorks, www.voicescienceworks.org/harmonics-vs-formants.html.

Smith, Steven W. “The Scientist and Engineer's Guide ToDigital Signal Processing.” How the FFT Works, www.dspguide.com/ch12/2.htm.

“Tech Stuff - Frequency Ranges.” Zyrax.info, www.zytrax.com/tech/audio/audio.html.

Tektronix. “FFT Tutorial.” YouTube, YouTube, 17 May 2012, www.youtube.com/watch?v=zKKGA30bHG0.

“The Music Theory Behind Acoustics and Harmonics.” Dummies, https://www.dummies.com/art-center/music/the-music-theory-behind-acoustics-and-harmonics.