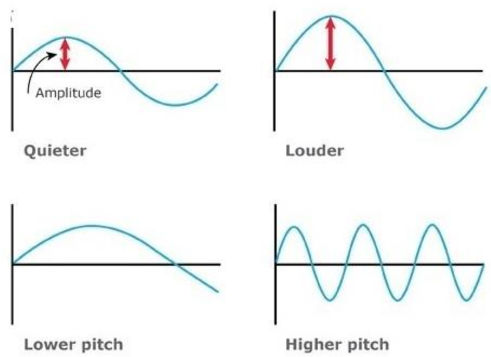
Investigation 2

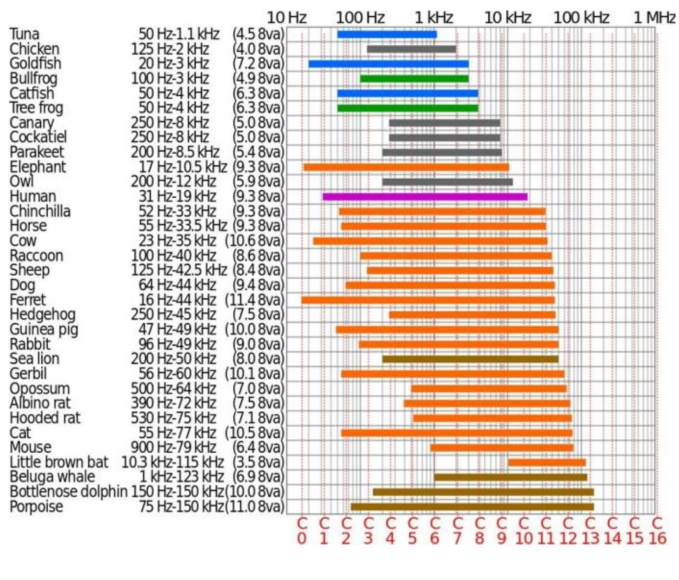
Audio Frequency

The concept of *audio frequency* is determined by the way in which sound waves oscillate whilst travelling to our ears and is measured by counting the total number of waves produced per second. Our ears vibrate in the same frequency that the sound is producing, thus turning the sound into electric frequencies that our brain is capable of processing.



The *audio frequency spectrum* represents the range of frequencies that the human ear can interpret and what we're able to hear in terms of music and pitch. Sound frequency is measured in the *Hertz* (Hz) unit. This audible frequency range, in the average person at birth, is from 20Hz to 20000Hz, or 20 kHz (kilo Hertz). The higher the frequency, the higher the pitch of the sound produced.

Factors like age and hearing damage can significantly reduce the hearing range of a person, making it difficult for people after certain incidents or age to detect certain sounds, often being those that are lower in the scale. Examples shown in the graph below.

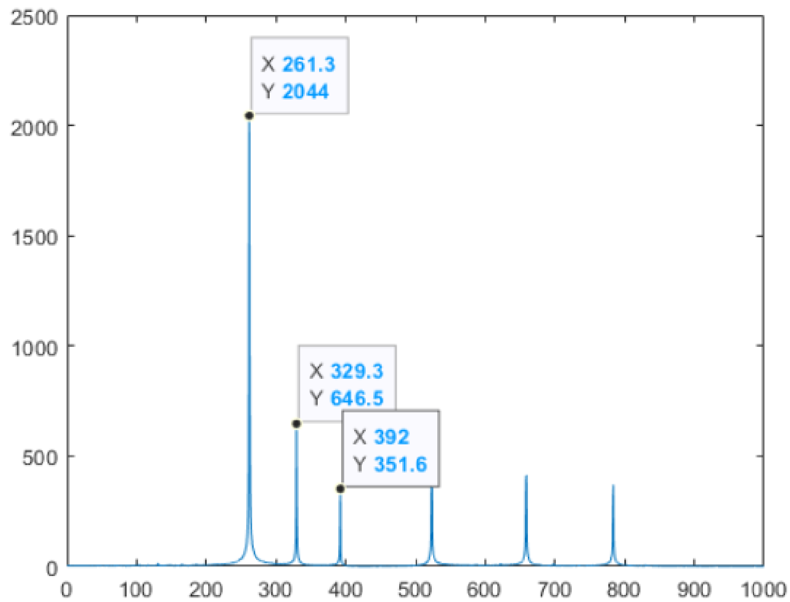


MATLAB ONRAMP

Audio signals are usually comprised of many different frequencies. For example, in music, the note 'middle C' has a fundamental frequency of 261.6 Hz, and most music consists of several notes (or frequencies) being played at the same time.

In this project, you will analyze the frequency content of an organ playing the C chord.

The C chord consists of the C (261.6 Hz), E (329.6 Hz), and G (392.0 Hz) notes. The highlighted points in this frequency plot correspond to each note.



Task 1

The C chord recording is stored in a file named Cchord.mat. This file contains two variables:

y: signal from recording

fs: sampling frequency

This task uses the numel function to return the number of elements in an array.

Load the file Cchord.mat.

Create a variable named n that contains the number of elements in y. Then use n to create an evenly-spaced vector t that starts at 0, ends at n-1, and has elements that are spaced by 1.

Task 1

```
load Cchord.mat
n = numel(y)
t = 0:n-1
```

```
n = 12288
t = 1×12288
     0     1     2     3     4     5     6 ...
```

Test Results: Correct!

- ✓ Are the variables `fs` and `y` loaded from the MAT-file?
- ✓ Does the variable `n` exist?
- ✓ Is `n` assigned correctly?
- ✓ Does the variable `t` exist?
- ✓ Is `t` assigned correctly?

Task 2

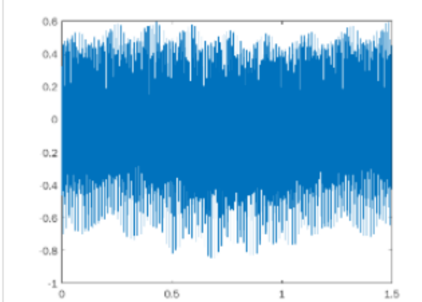
t now has the correct number of points, but it needs to represent the times when the audio signal was sampled. You can use the sampling frequency fs to convert the vector to time (in seconds).

Divide t by fs. Assign the output to the same variable t. Then plot y against t.

Task 2

```
t = t/fs
plot(t,y)
```

```
t = 1×12288
     0    0.0001    0.0002    0.0004 ...
```



Test Results: Correct!

- ✓ Is `t` updated correctly?
- ✓ Is `t` on the x-axis?
- ✓ Is `y` on the y-axis?

Task 3

In the plot, notice that y is periodic, but it's not a simple sine wave. It's made up of multiple sine waves with different frequencies.

A Fourier transform will return information about the frequency content of the signal. The location of the dominant frequencies will show what notes are contained in the chord.

You can use the fft function to compute the discrete Fourier transform of a vector.

fft(y)

The output values from fft are complex numbers. You can use the abs function to get the magnitude.

Create a variable named yfft that contains the absolute value of the discrete Fourier transform of y.

Task 3

```
yfft = abs(fft(y))
```

```
yfft = 1×12288
10^3 ×
    0.0028    0.0026    0.0026    0.0026 ...
```

Test Results: Correct!

✓ Does the variable yfft exist?

✓ Is yfft assigned correctly?

Task 4

In Tasks 1 and 2, you calculated the time vector t for the signal y. Similarly, you need to calculate the frequency vector f for your FFT vector yfft.

Create an evenly-spaced vector f that starts at 0, ends at n-1, and has elements that are spaced by 1.

Task 4

```
f = 0:n-1
```

```
yfft = 1×12288
10^3 ×
    0.0028    0.0026    0.0026    0.0026 ...

f = 1×12288
     0     1     2     3     4     5 ...
```

Test Results: Correct!

✓ Does the variable f exist?

✓ Is f assigned correctly?

Task 5

The vector f now contains n points. To convert these points to frequencies, you can multiply the entire vector by the sampling frequency (fs) and divide it by the number of points (n).

f will contain frequences from 0 to fs. The dominant frequencies are located at the beginning of f. You can use the xlim function to zoom in on the area of interest.

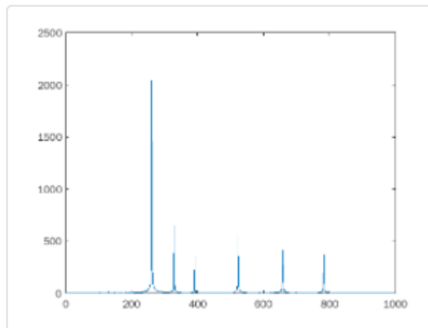
```
xlim([xmin xmax])
```

Multiply f with fs/n. Assign the output to the same variable f.

Plot yfft against f using the x-limits 0 and 1000.

Task 5

```
f = f*fs/n
plot(f,yfft)
xlim([0 1000])
```



Test Results: Correct!

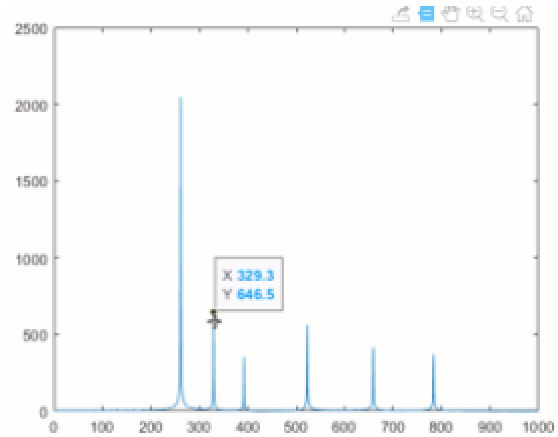
- ✓ Is `f` updated correctly?
- ✓ Is `f` on the x-axis?
- ✓ Is `yfft` on the y-axis?
- ✓ Are the x-limits correct?

Further Practice

Use the data cursor in the output pane to see the frequency locations.

The first three spikes are the notes comprising a middle C chord.

What are the other three spikes? When a chord is played, the signal contains fundamental frequencies and the associated harmonics. In this case, the harmonics are another octave of the same chord.



Using the frequencies in the table below, you can see that the 6 spikes in the plot correspond to the fundamental frequencies and the first harmonics of a middle C chord.

Note	Frequency
C_4	261.6
E_4	329.6
G_4	392.0
C_5	523.3
E_5	659.3
G_5	784.0

Bibliography

https://matlabacademy.mathworks.com/R2021a/portal.html?course=gettingstarted#chapter=10&lesson=2&section=1

https://www.headphonesty.com/2020/02/audio-frequency-spectrum-explained/

https://www.attune.com.au/2020/02/12/lets-talk-sound-frequencies/

https://www.sciencelearn.org.nz/resources/573-measuring-sound#:~:text=Frequency%20is%20measured%20in%20hertz,fixed%20point%20in%201%20second.