1 Tips

1.1 The Discrete Fourier Transform and the FFT Algorithm

The discrete Fourier transform is defined by:

$$S(k_f) = \sum_{n=0}^{N-1} s(n_t) e^{-2i\pi k_f n_t / N} \tag{1}$$

Where,

$s(n_t)$ is the sound signal (your data)

$S(k_f)$ is the Fourier transform of the signal

n_t is an integer number associated with time (i.e. n_t the n^{th} time step)

k_f is an integer number associated with the frequency

The discrete Fourier transform algorithm uses these indices (n_t) regardless of the time associated with them. You must therefore associate by yourself the right $S(k_f)$ with the right frequency (i.e. the right index (k_f)). You will simply call the Fourier function like so:

$$\boldsymbol{S} = Fourier[\boldsymbol{s}] \tag{2}$$

The discrete Fourier transform algorithms will compute the Fourier transform from $k_f = 0$ to $k_f = N - 1$, this will give you values for frequencies from $-f_s/2$ to $f_s/2$. In the context of this lab we are only interested in positive frequencies. Also, the Fourier transform is a complex number; we are only interested in the modulus of this number.

Therefore, you need to plot these two vectors:

$$\boldsymbol{FT} = \left| \boldsymbol{S}(1, 2, ..., \frac{N}{2} + 1) \right| \tag{3}$$

$$\boldsymbol{f} = \frac{f_s}{N} \left\{ 0, 1, 2, 3, ..., \frac{N}{2} \right\} \tag{4}$$

Where,

N is the number of time steps recorded

$f_s = \frac{1}{t_{i+1} - t_{i-1}} = \frac{N-1}{T}$ is the sampling rate

T is the total recording time

As you can see from equation 4, the difference in frequency between two points in the Fourier transform is $\frac{f_s}{N}$, this is the *resolution* of the Fourier transform. Hence, we can adjust the way we acquire data to obtain the resolution needed.

We know the different peaks in frequencies will be distant by roughly 100Hz, if we want to be precise at about 5% of the difference we need a resolution of about 5Hz. The number of data points (N) recorded by the oscilloscope is constant at 45,000. We can only adjust the time period (T) on which these points are recorded. It needs to be adjusted to:

$$\text{res} = 5\text{Hz} = \frac{f_s}{N} = \frac{\frac{N-1}{T}}{N}$$

so,

$$T = \frac{N-1}{\text{res} \cdot N} \approx \frac{1}{res} = 0.2\text{s}$$

We want to be able to average multiple Fourier transform with such a resolution, hence we will record for a total time of mT where m is the number of Fourier transform we want to average. For $m = 100$, we need to record for a total time of about 20 seconds.

1.2 Useful information

Discrete Fourier Transform

Moving Average

2 Procedure

In the laboratory:

1. Record raw sound data using the oscilloscope for at least 20 seconds or 1 second per division (see why we chose 10 seconds in section 1.1).
2. Save data in .CSV format on a flash drive

On the computer:

1. Import .CSV file to Mathematica, Matlab, Python or any program of your choice
2. (a) Perform a discrete Fourier transform on the complete set of data. You should use a pre-programed FFT function. Be sure to carefully read the documentation on the function before using it and read the tips in section 1.
   (b) Plot the result. What do you observe?
   (c) Estimate the signal to noise ratio.
3. (a) Separate your data into 100 equal parts maintaining the original order.
   (b) Perform the discrete Fourier transform on each part of the data. If you separated the data evenly in the previous step, you will be able to calculate the average of these transforms.
   (c) Plot the result. What do you observe?
   (d) Estimate the signal to noise ratio? How is it different from part 1?
4. You should be able to see clear peaks in your data. Measure the position (in frequency) of these peaks. Once again Mathematica, Matlab and Python have pre-implemented functions to find peaks.
5. What is the full width of the peaks at half of their intensity (FWHM)? How can this be used as a way to estimate the uncertainty on your measurement?