# Term Project

ENGR 362: Digital Signal Processing I
School of Engineering
The University of British Columbia — Okanagan
Document Version: v1
Date: October 29, 2019

## 1 Objectives

The objectives of this project are to give students experience with:

1. using an analog to digital converter to capture an audio signal;
2. analyzing the audio signal in the frequency domain using the DFT;
3. converting a frequency domain response into a power spectrum;
4. using the inverse DFT to estimate the autocorrelation function from the power spectrum;
5. implementing an FIR bandpass filter using the window design technique;
6. determining a system response using the filter.

## 2 Tuning Fork

### 2.1 Capture Audio

Capture at least one 10 second audio clip of an oscillating tuning fork. Make sure you note the frequency of the tuning fork(s) which you are working with.

1. Capture a 10 second discrete time signal for a tuning fork tone (tuning forks will be given to you in calss). Use the `audiorecorder` object in Matlab to capture the audio. The audio recorder object has parameters that set the sample rate $F_s$, the number of bits, number of channels (mono or stero) and the input device identifier. A configuration using a sample rate of 8 kHz, 16 bits, and one channel can be used. The input device default is 0 which is associated with the built in microphone on a laptop. An external microphone can also be used and will have a different input device number. On a Windows machine the `audiodevinfo` command can be used to query the hardware and device ID's. Example code for recording audio is shown in the demo code `record_audio_v1.m`.
2. Save the audio clip to a `.mat` file for post processing. Also save the sample rate and any other variables that are associated with the capture of the audio clip. Several audio data sets can be saved to the same file using the command:
   `save('filename','var1','var2', ... ,'-append')`.
   Make sure the variables have unique names if you are appending to the file to ensure previously saved data is not overwritten.

### 2.2 Analyze Audio

Create a new script or function to analyze the tuning fork audio data.

1. Make a calibrated frequency scale to use for plotting the DFT. Use the sample frequency and number of points $N$ (length of the time vector) to create this vector.

2. What is the fundamental frequency of the tuning fork based on your frequency domain analysis? How close is this frequency relative the frequency which was stamped on the tuning fork?

## 3 Noisy Tuning Fork

A ten second audio clip for a tuning fork is saved in the file `noisy_tuning_fork.mat` (you can also create it by adding noise using MATLAB command `randn`) . Design a bandpass filter using the window technique to increase the signal to noise ratio of the signal and attenuate the noise. Play the original audio track and then compare this with the filtered audio track. It should sound much better.

1. Convert the discrete time signal into the frequency domain. Plot the power spectrum in dB with respect to a calibrated frequency axis. Determine the frequency of the tone which is just visible above the noise floor.
2. Use the window technique to design a bandpass filter the noisy time signal. Plot the filtered spectrum in the frequency domain and listen to the filter tone. Is the signal to noise ratio of the filter signal much better.
3. The time signal is much longer than the impulse response of the filter. The convolution of the filter and the input signal can be done in either the time or frequency domain. Because of the disparity in vector lengths, use the Matlab function `filtfilt`.

## 4 Grading

- Grading the project will consist of one in-class evaluations and a written report. The dates of in class evaluations are as follows.
  During in class evaluation, students will demonstrate their results on their own personal laptops or on a lab PC. The demonstration will be followed with a question and answer session. TA will spend about five minutes with each student to review and assess their work.
- Submit a final report by 10:30 AM on **Nov. 27, 2019**.

# 5 Written Report

Hand in a written report with the following figures as well as answers to the questions below.

1. Plot the magnitude, phase, and power spectrum (in dB) of a tuning fork signal.
   (a) Use a calibrated frequency axis for the plots.
   (b) Place a marker on the power spectrum to show the frequency of the fundamental tone and compare this with the frequency stamped on the tuning fork.
   (c) Make sure the phase spectrum is properly labeled (degrees, radians).
   (d) Comment on the symmetry of the magnitude and phase spectrum.
2. Document your filter design for improving the signal to noise ratio of the noisy tuning fork. State what window you used, the lowpass cutoff frequency $f_c$, and the centre frequency of the bandpass filter $f_0$. Also give the frequencies in Hertz.
3. Show the magnitude and phase spectrum of the lowpass filter or filters prototype that you used to generate the bandpass filter.
4. Show the magnitude and phase spectrum of the bandpass filter used for filtering the noisy tuning fork spectrum. Adjust the x-axis scale so the passband, transition band and stop band are easily distinguished.
5. Describe the method that you used to filter the noisy tuning fork signal. Did you implement a convolution, difference equation, or work in the frequency domain?
6. Generate two plots which show the unfiltered tuning fork power spectrum and the filtered tuning fork spectrum. Show the y axis in dB and the x-axis in frequency (Hz or kHz).