

# ECE2312 – Discrete-Time Signal and System Analysis

## Project 1

Due: 5.00 pm on Thursday February 16, 2023

### 1. Project Goals

The objective of this course project is to explore how to artificially generated sounds using sinusoidal functions such as  $\sin(x)$  and  $\cos(x)$ , how to include these sounds with other sound files, and how to remove these sounds using simple filtering techniques. This project will be built on top of the previous project.

### 2. Sine Tone Generation

Fourier Transform of a sine signal with frequency  $\omega_o$  is given by the following expression:

$$\mathcal{F}\{\sin(\omega_o n)\} = \frac{\pi}{j} \sum_{l=-\infty}^{+\infty} \{\delta(\omega - \omega_o - 2\pi l) - \delta(\omega + \omega_o - 2\pi l)\}. \quad (1)$$

Consequently, in any spectrogram we should be able to identify the presence of a sine signal or tone by a horizontal line across time at one specific frequency corresponding to  $\omega_o$ .



Using the sin function, produce a sine tone at a frequency of 5000 Hz using the sampling rate  $F_s$  employed for the recording of your speech files in Project 1. Make sure the time duration of sine tone is equal to that of your previously recorded speech file: “The quick brown fox jumps over the lazy dog.”

- Play your sine tone using the **sound** function and comment on what you hear.
- Save the resulting sine tone to a WAV file using the filename “team[[yourteamnumber]]-sinetone.wav” and include in your submission to CANVAS.
- Plot the spectrogram of the sine tone to include in your report.

### 3. Chirp Signal Generation

Once you have implemented a sine tone with a fixed frequency  $\omega_o$ , the next step is to generate a sine tone with a progressively increasing frequency over time. Such a sine tone is referred to as a “chirp.”



Using the sin function, produce a chirp signal based on a sine tone with a frequency that varies linearly from 0 Hz to 8000 Hz across the entire time duration of the signal. Please use a sampling rate  $F_s$  that was employed for the recording of your speech files in Project 1. Make sure the time duration of chirp signal is equal to that of your previously recorded speech file: “The quick brown fox jumps over the lazy dog”.

- Play your sine tone using the **sound** function and comment on what you hear.
- Save the resulting sine tone to a WAV file using the filename “team[[yourteamnumber]]-chirp.wav” and include in your submission to CANVAS.
- Plot the spectrogram of the sine tone to include in your report.

#### 4. Some Fun with Sine Tone

One of the all-time classic films of all time is Close Encounters of the Third Kind (CETK), directed by Steven Spielberg and starring Richard Dreyfuss<sup>1</sup>. At the center of this film, which involves first contact between humans and extraterrestrials, is a collection of five sine tones that are used as a common starting point of communications. A link to a YouTube video playing these five tones in sequence can be found at the following location: <https://www.youtube.com/watch?v=kpsEqINeMS4>



Using the sin function, produce an approximate rendition of this same sine tone pattern from CETK within the 0 Hz – 8000 Hz frequency range. Please use a sampling rate  $F_s$  that was employed for the recording of your speech files in Project 1.

- Play your rendition of this sine tone pattern using the sound function and comment on what you hear.
- Save the resulting sine tone pattern to a WAV file using the filename “team[[yourteamnumber]]-cetk.wav” and include in your submission to CANVAS .
- Plot the spectrogram of the sine tone pattern to include in your report.

---

<sup>1</sup> <https://www.imdb.com/title/tt0075860/>

## 5. Combining Sound Files

With the sine tone and chirp signals generated, let us explore how this signal can be added to one of our previous generated speech signals from Project 1.



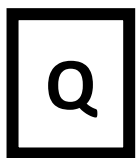
Load the previously recorded speech file from Project 1, namely, “The quick brown fox jumps over the lazy dog,” and add the 5000 Hz sine tone to it. Note that both signals should be the same lengths and should both be row vectors or column vectors.

- Play the resulting signal using the sound function and comment on what you hear.
- Save the resulting signal to a WAV file using the filename “team[[yourteamnumber]]-speechchirp.wav” and include in your submission to CANVAS.
- Plot the spectrogram of the resulting signal to include in your report.

## 6. Speech and Audio Filtering

In class, we will start to explore how the process of filtering can remove signal energy across specific portions of frequency while maintaining signal energy across other parts of frequency. In this section, we will construct a lowpass filter using the `firls` command and convolve it with the signals generated in the previous section using the `filter` command.

The `firls(N,F,A)` implements a least squares linear phase finite impulse response (FIR) filter given the parameters `N`, `F`, and `A`. The input variable `N` specifies the order of the filter, which is related to the number of coefficients that describe it. As for input variables `F` and `A`, these are vectors that describe the shape of the filter from 0 rad/s to  $\pi$  rad/s. Remember that discrete time systems with real coefficients produce symmetric frequency responses from  $\pi$  rad/s to  $\pi$  rad/s, and that these frequency responses are periodic in frequency every  $2\pi$  rad/s. Suppose one would like to design a lowpass filter that has approximately 50% passband and 50% stopband. In this case, we would define `F=[0 0.45 0.55 1]` and `A=[1 1 0 0]`, where at frequencies of 0,  $0.45\pi$ ,  $0.55\pi$ , and  $\pi$  we have magnitude values of 1, 1, 0, and 0, respectively.



Given the combined speech and sine tone signal from the previous section, design a lowpass filter with a cut-off frequency of 4000 Hz and apply it to this signal.

- Play the resulting signal using the sound function and comment on what you hear.
- Save the resulting signal to a WAV file using the filename “team[[yourteamnumber]]-filteredspeechsine.wav” and include in your submission to CANVAS.
- Plot the spectrogram of the resulting signal to include in your report.

## 7. Stereo Fun

Since human beings possess binaural sensory perception, in many instances we record and play audio information in stereo rather than mono format. The sounds being listened to are often very similar (our brains often expect this) with slight differences coming from the environment and how the sound reaches either ear. However, when stereo audio information is being listened to where one of the two audio channels possesses significantly different sounds, then our brains find this situation somewhat uncomfortable.



Create a stereo audio signal, with the left audio channel consisting of the previously recorded speech file, “The quick brown fox jumps over the lazy dog,” while the right audio channel contains the same speech signal with the 5000 Hz sine tone added to it.

- Play the resulting stereo signal using the sound function and comment on what you hear. It is recommended to listen this signal using headphones.
- Save the resulting stereo signal to a WAV file using the filename "team[[yourteamnumber]]-stereospeechsine.wav" and include in your submission to CANVAS.
- Plot the spectrograms for each audio channel of the stereo signal and include them side-by-side in your report.

## 8. Project Submission

Each student team should submit the following via the ECE2312 CANVAS website:

- A project report in PDF format that answers all the questions indicated in this project handout. Additionally, the following elements should be included:
  - Coverpage with course number and title listed, names of all student members of submitting team, date of submission, project number.
  - Descriptive captions for all figures contained within report submission.
  - Sufficiently detailed responses to all questions, providing insights about the answers provided. Responses should be written in complete sentences/paragraphs, should be grammatically correct, and written using professional vocabulary.
  - Proper pagination and formatting (e.g., 1-inch margins, single-spaced lines, 11-point serif fonts).
  - Proper pagination and formatting (e.g., 1-inch margins, single-spaced lines, 11-point serif fonts).
  - Proper pagination and formatting (e.g., 1-inch margins, single-spaced lines, 11-point serif fonts).
- All requested WAV files as indicated in this handout.

- Link to the Github/Gitlab containing all source code generated by the student team. This code should be in a condition that it can be executed by the teaching assistant to verify its functionality. Moreover, you will run and show this code on **February 16**.