# Lab 4: Flow Control in MATLAB

ENGR 151

Released: September 23, 2024

Due: September 29 2024, 11:59 p.m.

## 1 Music Animation (Selection, Iteration, and Functions)

In these exercises, you get to play around with sound! Successive exercises pick up on where the previous one left off, so approach these problems in order.

## 2 Sound Basics

First, let's go over some basics. Sound is vibrating air. These vibrations occur at different frequencies. When the frequency is very regular, we hear this as music. Music notes can be modeled by sine waves with certain amplitudes and frequencies. The amplitude is how loud the sound is and the frequency changes the pitch of the note. A low note has a low frequency, and a high note has a high frequency. For example, the graph below shows a low note being played followed by a higher note. In music, certain frequencies are given names. For example: a frequency of 440 Hz is the note 'A' (in octave 4). More on this later.

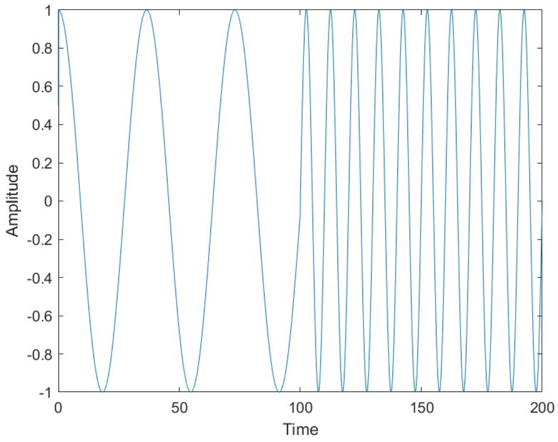

Figure 1: Two notes of different frequencies.

## 3 Exercise 1: First Musical Steps

MATLAB has a function called `sound`, which will play a note if given a sine wave as an argument. Let us start by playing one note. To create a sine wave, we will generate many points using the `sin` function. For this lab, you must use a resolution of 1/8192. (8192 is referred to as the "sampling frequency"). For example, you can use the code:

```
duration = 1; % Duration of the note (seconds)
frequency = 440; % Frequency of the note in Hz, this is an 'A'
samp_freq = 1/8192; % Sampling frequency
time_arr = 0:samp_freq:duration; % Create one second of points
wave = sin(2*pi*frequency*time_arr); % Create sin wave of desired frequency and duration
```

Now if you call `sound(wave)` it will play the note A for 1 second! Canvas autograder does not play sounds, you must run this in your local MATLAB instance to hear it. Make a function called `play_note` that generates a musical note for 1 second, plays it on your speaker, and plots the wave! Make sure to use the `sound` function to play the note! Comment out or remove the `sound` function when submitting to the canvas autograder.

- Input: `frequency`, the frequency of the note to be played in Hz.
- Output: `wave`, the final array of the sound wave.

## 4 Exercise 2: Musical Note Generator

The goal of this exercise is to write a function which takes in an array of one-character strings and plays each of them in sequence. If you wanted to play a couple of notes in sequence right now, you might try something like this:

```
sound(wave1);
sound(wave2);
sound(wave3);
```

Go ahead and make yourself a couple waves and try it! You will notice that the notes do not play in sequence, but rather all at once (depending on which notes you chose, this might actually sound nice). To get them to play one after the other, we must use the `pause` function. The `pause` function takes in one argument the number of seconds to halt the program. So, if you want a note to play for one second without any other notes interrupting it, you must first construct the wave of note to be one second long, play it with the sound function, and then pause the program for one second. In this exercise, you should make all the notes play for one second, followed by a one second silence. Use the table below for the relationships between the note and its corresponding frequency.

| Note Name | Note Frequency |
|---|---|
| C | 261.63 |
| D | 293.66 |
| E | 329.63 |
| F | 349.23 |
| G | 392.00 |
| A | 440.00 |
| B | 493.88 |

Write a function `play_notes` that takes in an array of the notes as its argument and plays those notes in succession (each one second in duration, with one second of silence between each). You should be using a loop for this! When you are testing your code and want to produce your own array of notes, be sure to use double quotes (like "A") to encase the letters, not single quotes (like 'A'). Your function should output one wave with all of the notes waves in it, as well as all of the pauses in between each note (these would just be sections of zero amplitude). Remember to comment out or remove the `sound` function when submitting to the Canvas autograder.

## 5 Submission

This lab assignment is due on September 29th. Please turn in both `play_note` and `play_notes` to the respective assignments on Canvas.