EEE 391
Basics of Signals and Systems
Spring 2022–2023
MATLAB Mini Project 1
due: 22 March 2023, Wednesday by 23:00 on Moodle

In this assignment, you will gain experience with audio signals. First, you will convert the given signal representations from Cartesian to polar coordinates. By using the frequency, magnitude, and phase information, you will be able to plot your signals in the time and frequency domains. Then, you will play your audio signals and hear what they sound like. In the second part of the assignment, you will experience a phenomenon called beat frequency.

Part 1: Periodic and Aperiodic Audio Signals

Complex exponential numbers can be represented in two different formats which are Cartesian and polar coordinates. While the Cartesian coordinates (x, y) represent real and imaginary components, the polar coordinates (r, θ) represent the magnitude and phase angle of the complex number. Conversion between polar and Cartesian coordinates is possible through the following set of equations:

$$r = \sqrt{x^2 + y^2} \qquad x = r\cos\theta \tag{1}$$

$$\theta = \mathrm{arctan2}(y, x) \qquad y = r\sin\theta \tag{2}$$

The different frequency components f_k of a periodic signal are harmonically related, that is, $f_k = kf_o$ where k is an integer and f_o is the fundamental frequency. Such a periodic signal can be expressed in terms of complex exponential signals as follows:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi k f_o t} \tag{3}$$

where a_k are the Fourier series coefficients. You will recognize this as the synthesis formula of the Fourier series representation of periodic signals.

The following table provides the frequency components and complex amplitudes of two different audio signals where the D_i values $(i = 1, \ldots, 8)$ correspond to the eight digits of your ID number from left to right as $D_1D_2D_3D_4D_5D_6D_7D_8$.

audio signal 1		audio signal 2	
frequency (Hz)	complex amplitude	frequency (Hz)	complex amplitude
$\pm D_1 00$	$D_5 00 \pm jD_6 00$	$\pm D_1 00$	$D_5 00 \pm jD_6 00$
$\pm D_2 00 \times 2$	$-D_7 00 \pm jD_8 00$	$\pm D_2 00 \times \sqrt{5}$	$-D_7 00 \pm jD_8 00$
$\pm D_3 00$	$D_1 00 \mp jD_2 00$	$\pm D_3 00$	$D_1 00 \mp jD_2 00$
$\pm D_4 00 \times 4$	$D_3 00 \mp jD_4 00$	$\pm D_4 00 \times \sqrt{15}$	$D_3 00 \mp D_4 00$

(a) First, express the given complex amplitudes in polar form. In other words, calculate the magnitude and phase angle values of the complex amplitudes. Express both signals first as a sum of complex exponential signals, and then, as a sum of sinusoids. Note that, depending on your ID number, some or all of the relevant digits in a certain row of the table may be zero; this is acceptable.

(b) Evaluate each audio signal at discrete instants of time. These are called *samples* of the signal. To obtain uniformly spaced samples in time, define an independent variable $t_s = nT_s$ that changes as $0 : T_s : t_{\text{end}}$, where n is a nonnegative integer, T_s is the sampling period, and $f_s = \frac{1}{T_s}$ is the sampling frequency. Note that the units of t_s are seconds. Obtain the samples of your signals using $x_1[n] = x_1(nT_s)$ and $x_2[n] = x_2(nT_s)$ over a duration of $0 \le nT_s \le t_{\text{end}}$ where $t_{\text{end}} = 0.1$ s. Choose the f_s value 100 times the largest frequency component that you have in the two signals. Use the same f_s value for both audio signals. Store the two sequences $x_1[n]$ and $x_2[n]$ in arrays called **x1** and **x2**, respectively. These will be plotted later in part (e).

(c) Turn on the speakers of your computer. Examine the `sound(.)` and `soundsc(.,.)` commands of MATLAB by typing `help sound` and `help soundsc` in the MATLAB command window. Notice that `soundsc(.,.)` requires the sampling frequency f_s at which the signal samples were created as its second argument. To be able to hear the sounds for a sufficiently long period of time, for this part only, extend the duration of your sequences to $t_{\text{end}} = 1$ s. Play and listen to the two audio signals by using the `soundsc(.,.)` function using `soundsc(x1,`f_s`)` and `soundsc(x2,`f_s`)` commands. Listen to the two sounds.

(d) Type `help plot` in the MATLAB command window to see what you can do with the `plot` command which is one of the vital commands of MATLAB. Also study the MATLAB commands `xlabel, ylabel, title, xlim, ylim`, and `grid`. These commands are essential for producing professional looking plots in MATLAB. Make sure that you can properly use these commands.

(e) Plot the **x1** and **x2** arrays that you acquired in part (b) using the `plot(x,y)` function with respect to t_s in the interval $0 \le t_s \le t_{\text{end}}$ where $t_{\text{end}} = 0.1$ s. Be careful with the units and the labels. Determine whether each of the two audio signals is periodic or not.

(f) Plot the frequency spectrum of each of the two audio signals. Compare the frequency values of the two signals. What makes one signal periodic or nonperiodic? What do magnitude, phase angle, and frequency values correspond to on the plots? Do you observe complex conjugate symmetry on the plots? Explain the underlying reasons for the existence or nonexistence of complex conjugate symmetry.

Part 2: Beat Frequency

In this part, you will experience an interesting audio effect or phenomenon called a *beat note* or *beat frequency*, which may sound like a warble.

(a) First, create a simple sinusoidal audio signal with a cyclic frequency of 300 Hz. Mathematical formula for this signal is given by:

$$x_1(t) = \sin(2\pi 300 t) \tag{4}$$

Play this audio signal by using the `soundsc(x(t),`f_s`)` function. Set the f_s value to 100 times the frequency of the signal. Plot this signal in the time domain. Discretize your signal in the same way as in part (b) of the first part of the assignment.

(b) Create a second sinusoid $x_2(t)$ by increasing the frequency of $x_1(t)$ by 10 Hz. Write the closed-form expression for this sinusoid and repeat part (a) for $x_2(t)$. Use the same f_s value as in part (a). Plot the two sinusoids on the same figure to see how similar they are. Then play them by using the `soundsc(x(t),`f_s`)` function to hear what they sound like. Do you hear a noticeable difference?

(c) Obtain the signal $x_3(t) = x_1(t) + x_2(t)$. Plot the three signals on the same figure as a function of time and observe the effect of summation. Do $x_1(t)$ and $x_2(t)$ strengthen or weaken each other? Use the same f_s value as in parts (a) and (b). Plot also the frequency spectra of the three signals. Play the three audio signals by using the `soundsc(x(t),`f_s`)` function. Does the $x_3(t)$ signal sound similar or different from $x_1(t)$ and $x_2(t)$?

(d) Peak values of $x_1(t)$ and $x_2(t)$ overlap at some points. The "beat frequency" corresponds to the number of times that the peak values overlap per unit time. How can you find the beat frequency of your signals? Do you see a relationship between the beat frequency and frequencies of $x_1(t)$ and $x_2(t)$?

The signal $x_3(t)$ fades in and out or warbles because the signal envelope is rising and falling. This is the phenomenon called "beating" of tones in music. Some musical instruments naturally produce beating notes. Musicians use this beating phenomenon as an aid in tuning two instruments to the same pitch. When two notes are close but not identical in frequency, the beating phenomenon is heard. As one pitch is changed to become closer to the other, the effect disappears, and the two instruments are then "in tune."

Note: In this assignment, you are *not* allowed to use symbolic operations in MATLAB. To modify the styles of the plots, to add labels, and to scale the plots, use only MATLAB commands; do *not* use the GUI of the figure windows. When your program is executed, the figures must appear exactly the same as you provide in your solution. You need to write your MATLAB codes not only correctly but efficiently as well.

Submit the results of your own work in the form of a well-documented report on Moodle. Borrowing full or partial code from your peers or elsewhere is not allowed and will be penalized. Please include all evidence (plots, MATLAB codes, MATLAB command window print-outs, etc.) as needed in your report. Append your MATLAB code at the end of your assignment, do not upload it separately. The axes of all plots should be scaled and labeled. Typing your report instead of handwriting some parts will be better. Please do not upload any photos/images of your report. Your complete report should be uploaded on Moodle as a single good-quality pdf file by the given deadline. Please try to upload several hours before the deadline to avoid last minute problems that may cause you to miss the deadline. Please DO NOT submit any hardcopies or files by e-mail. All assignments need to go through Turnitin.

IMPORTANT NOTE:
Please name the pdf file you submit on Moodle as follows using only lower-case English characters for your first name, middle name (if any), and lastname. Please use your full name as it appears on the Bilkent system.

MP1_firstname_middlename_lastname.pdf
filename example for Ayşenur Çiğdem Sürücü:
MP1_aysenur_cigdem_surucu.pdf