

# EEE - 391: Basics of Signals and Systems

## Matlab Assignment 1

---

Assigned: 20.10.2022 Thursday

Deadline: 19.11.2022-Saturday 23:55

In part A of the assignment you will perform basic operations regarding Fourier series and sampling. In part B, you are going to compute the Fourier series coefficients of a voice that you will record and re-synthesize the voice from the computed Fourier series coefficients. You are going to investigate the effect of some changes on the Fourier series coefficients on the re-synthesized voice, as well.

### Part A)

**A.1** Consider a signal of the form  $x_1(t) = \cos(2\pi f_0 t)$ . Here  $f_0$  is the frequency.

Take  $t = [0 : 1/8192 : 1]$  seconds for the following questions.

1. Take  $f_0 = 550$  Hz. Compute  $x_1(t)$  and store it in an array named  $x1$ . Plot  $x1$  versus  $t$  as if the signal is a continuous time signal. Turn on the speakers of your computer. Then type `soundsc(x1)`. Listen to the sound.
2. Do the same for  $f_0 = 430$  Hz, but do not produce any plot. What happens to the pitch of the sound as the frequency decreases?
3. Do the same for  $f_0 = 750$  Hz, but do not produce any plot. What happens to the pitch of the sound as the frequency increases?
4. Let digits of your Bilkent ID number be  $[d1, d2, \dots, d8]$

Assume  $f_1 = d5d6d7$  and  $f_2 = d2d3d4$ , both are three digit numbers ( $f_1$  may have less digits)

Assume a signal  $y(t) = 2 \cos(2\pi f_1 t) + 3 \sin(2\pi f_2 t)$

Let  $f_{\max}$  be the maximum frequency component contained in  $y(t)$ . Sample the signal with three sampling rates  $f_s = f_{\max}, 2f_{\max}, 4f_{\max}$

Plot  $y(t)$  and sampled signals.

5. Now, reconstruct the signal from its different samples from previous part. **We expect you to use interpolation methods you learned in class, rather than using built-in functions.** Plot reconstructed signals. Comment on the differences. Include your plots and code in report.

## A.2

A periodic signal  $x(t)$  has a period of  $T = 3$ . One period of this signal is given below

$$x(t) = \begin{cases} 3t & 0 \leq t < 1 \\ 5 - t & 1 \leq t < 2 \\ 3 & 2 \leq t < 3 \end{cases}$$

- 1) Plot three periods of  $x(t)$  for  $0 \leq t < 9$  on MATLAB.
- 2) Calculate Fourier series coefficients of  $x(t)$  analytically.
- 3) Let  $X_k$  be the Fourier series coefficient of the  $k$ th harmonic. Find a positive integer  $N$  such that approximately 95% of the energy of the Fourier series coefficients is confined within the interval  $k \in [-N, N]$ . For this purpose, calculate the total energy of the Fourier series coefficients analytically. That is, find  $N$  such that

$$0.95 \sim E_r = \frac{\sum_{k=-N}^N |X_k|^2}{\sum_{k=-\infty}^{\infty} |X_k|^2}$$

You can find  $N$  either analytically or using a Matlab code. (Hint: Use Parseval's relation to find the total energy, i.e. the denominator in above equation, of the Fourier series coefficients.)

- 4) For the value of  $N$  that you found in previous question, generate a stem plot of the magnitudes and the phases of  $X_k$ s for  $k \in [-N, N]$ . Put the labels and titles appropriately.

## Part B)

1. **Voice Recording:** In this part, you are going to generate an array in Matlab that includes to the samples of a periodic sound. For this purpose, you have two options:

**Option 1:** You can play a piano, a flute, or some other musical instrument which generates pure musical notes that are continuous in time (for example, playing a drum is inappropriate). And record this sound in Matlab (if you have a microphone on your computer, of course). For recording, use the following Matlab code:

```
recObj = audiorecorder;  
disp('Start recording.');
```

```
recordblocking(recObj, T);  
disp('End of Recording.');
```

where  $T$  determines the duration of the recording in seconds and you will decide this number in your recording.

You can listen to the sound that you recorded using

```
play(recObj);
```

Please make sure that the sound that you recorded and listened to are the same.

Then, extract the sound signal to an array using the following code

```
soundArray=getaudiodata(recObj);
```

This code generates an array **soundArray** that includes the samples of the sound signal. Please note that the sampling rate of this recording is 8 KHz, which is the default value of this sound recording command.

**Option 2:** If you do not have an instrument, you can utilize from virtual sources

found in the website. For example, the following websites provide some virtual instruments:

<https://virtualpiano.net/>

<http://www.apronus.com/music/flashpiano.htm>

<http://www.caseyrule.com/projects/piano/>

While playing these virtual instruments, if you have both microphone and speaker, you can record the sounds in Matlab by using the commands that are provided above.

You are highly encouraged to use a real musical instrument rather than synthetic sources of sound found on the internet i.e. Option 1 is preferable to Option 2. Not only will it be more fun, interesting, and educational, in our trial experiments, real instruments gave better and cleaner results. A human singer is also acceptable, if you or he/she can manage to produce a clean note. We still will allow the internet sources of sound if you do not have convenient access to a real musical instrument.

2. **Fundamental Period Calculation:** In this part, you are going to calculate the period of the signal that you record analytically. For this purpose, look at the webpage <https://pages.mtu.edu/~suits/notefreqs.html>. In this page, the notes and the associated frequencies are given. By using these frequencies and the notes that you played, compute the fundamental frequency of the signal that you recorded.

Then, look at the plot of **soundArray** and extract a portion where the note is recorded and generate another array, **partOfSoundArray**, from this portion. (There might be some silent parts at the beginning and end of **soundArray**. Also, the musical note will decay as time goes on. Therefore, the portion that you will extract to **partOfSoundArray** should be a part where there is no silence and also the musical note does not decay significantly. **partOfSoundArray** should include at least 100 period of the fundamental period.)

Then, justify your analytical calculation by looking at the plot of **partOfSoundArray**. Include your plots and comments to your report. Put correct labels and titles to your plot, as well.

3. **Fourier Series Analysis:** Now, you are going to compute the Fourier series coefficients (FSC) of the signal in Matlab. First, generate a new array, **firstPeriodOfPartOfSoundArray**, using **partOfSoundArray** that corresponds to first period of the sound file. That is, you will extract the portion of **partOfSoundArray** that corresponds to the time interval  $t \in [0, T)$ , where  $T$  is the fundamental period that you found above. Then, you will compute the FSCs in Matlab. Please note that, the Fourier series computation requires an integration, which is a continuous operation. In order to perform this operation in Matlab, you can use an approximate numerical technique. For this purpose, use **trapz** command of Matlab, where you can find the details in <https://www.mathworks.com/help/matlab/ref/trapz.html>.

Although the FSCs of a periodic signal are infinitely many, here, you will compute a finite number of them. Let  $k$  be the index of the FSCs where  $k \in [-N, N]$ . You will choose  $N$  such that, all the nonzero FSCs are guaranteed to be computed in your Matlab code. Please note that, due to numerical errors in the computation of the FSCs, you may not get identically zero FSCs in the locations where you expect to be zero. Instead, you may see some very small coefficients. You can ignore these nonzero but small coefficients and assume that they are zero.

Include your code and the plot of the magnitudes of the FSCs as a function of  $k$ .

#### 4. Comparison of Fourier Series for Different Instruments:

Now, you will record the exact same note from two different instruments (i.e,  $A_4$ , which is 440 Hz from a piano and a flute) in the way described in Part 1. Then, you will repeat part 2-3 for these signals and obtain the plot of the magnitudes of the FSCs as a function of  $k$ . Are the Fourier series coefficients you obtained same? If they are different, comment on the reason of difference. Include your code and plots.

5. **Fourier Series Synthesis:** Now you will write a Matlab code that generates one period of the periodic signal from the FSCs that you found in the previous part. Then, generate a plot of the re-synthesized signal and make sure that it is similar to one period of the original sound array. Next, generate another array by periodically extending the array that you created in this part. The length of the new array should be equal to **partOfSoundArray**. Include your codes and plot to your report.
6. **Voice File Generation:** In this part, you are going to generate a sound file from the array that you created in the previous part. For this purpose, use **audiowrite** command, where the details of this command can be found in <https://www.mathworks.com/help/matlab/ref/audiowrite.html>. During the generation of the sound file, do not forget to adjust the sampling rate of the recorded signal to 8 KHz. Then, listen to the sound file that you generated and make sure that you hear the same signal as **partOfSoundArray**. Include your code to your report.
7. **Synthesis of the Voice from Partial Fourier Series Coefficients:** In this part, you are going to synthesize another sound file using different subsets of the FSCs that you computed. Let  $M$  be the number of the nonzero harmonics that you saw in your plot and  $k = N_1, \dots, N_M$ , where  $0 < N_1 < \dots < N_M < N$ , be the positive index of the FSCs that corresponds to the frequency of those notes. Then, generate a new array by assuming that the FSCs are all zero except for  $k \in [-(N_2 - N_1)/2, (N_2 - N_1)/2]$  and, in this range, take the FSCs that you computed. In other words, compute a new signal from the interval,  $k \in [-(N_2 - N_1)/2, (N_2 - N_1)/2]$ , from the FSCs of the original signal.

Next, generate a sound file from the generated array using the procedure described in the previous parts. Then, listen to the sound that you generated and make your comments on the differences of the newly generated sound file and the **partOfSoundArray**.

Next, do the same for all  $i \in \{3, 4, \dots, M\}$  for  $k \in [-(N_i - N_1)/2, (N_i - N_1)/2]$ . Include your codes and comments to your report.

8. **Synthesis of the Voice from Fourier Series Coefficients whose Magnitudes are equated to one:** In this part, you will generate another sound signal using the modified versions of the FSCs. First, generate a new set of FSCs such that the phases of the initial FSCs are the same but the magnitudes are all equated to one. Then, as described in the previous parts, generate a new sound signal from these FSCs and listen to that sound. Do you hear a different sound? Include your code and comments to your report.
9. **Synthesis of the Voice from Fourier Series Coefficients whose Phases are equated to zero:** Now, you will perform a similar task to the previous item, where, instead of the magnitudes, you will modify the phases of the FSCs. So, generate a new sound signal from the FSCs such that their magnitudes are the same as the initial FSCs but their phases are all equated to zero. Then, generate a new sound signal from these FSCs and listen to that sound. Do you hear a different sound? Include your code and comments to your report.