İhsan Doğramacı Bilkent University
Electrical and Electronical Engineering Department

EEE391 Matlab Assignment 1

Assignment: 31/10/2018 Tuesday
Deadline: 26/11/2018 Monday (17:30)

You can leave your report to Sertaç Kılıçkaya (EE310) on 26/11/2018 between 15:40- 17:30. You can also submit your homework earlier than this deadline if you see one of the course TAs. Please do not leave your homework to a mailbox or a TA's desk and make sure that you submit your homework to one of the course TAs.

In addition to turning in your handwritten assignment, please e-mail the source code and sound file to eksioglu@ee.bilkent.edu.tr with the subject "EEE391 MA1 Submission" until deadline. These two files should be in a single ZIP file, named `YOURBILKENTID_MA1.zip`. Source code in the zipped folder should be named `YOURBILKENTID_MA1_SRC.m` and sound file should be named `YOURBILKENTID_MA1_SOUND.wav`. Don't forget to replace `YOURBILKENTID` with your actual Bilkent ID in the file names.

In this assignment, 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.

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:

- 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(16000, 8, 1);
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.

- 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. If you do not have access to a musical instrument, try the piano at the dining hall at some quiet hour. 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. You may find the page `https://pages.mtu.edu/~suits/notefreqs.html` which gives notes and the associated frequencies useful. By using these frequencies and the notes that you played, compute the fundamental frequency of the signal that you recorded. Make sure that the frequency you have found and the frequency in the data are consistent with each other. If the note you have played is not clear, calculate the frequency from the data.

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 $t \in [0T)$, 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 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 plots to your report.

5 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 16KHz. 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.

6 Voice Synthesis from Partial FSCs

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 $k = N_1, \ldots, N_M$ where $0 < N_1 < \ldots < N_M < N$, 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 $k \in [\frac{-(N_2-N_1)}{2}, \frac{+(N_2-N_1)}{2}]$ and, in this range, take the FSCs that you computed. In other words, compute a new signal from the interval, $k \in [\frac{-(N_2-N_1)}{2}, \frac{+(N_2-N_1)}{2}]$, 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 $i \in 3, 4, \ldots, M$ for $k \in [\frac{-(N_i-N_1)}{2}, \frac{+(N_i-N_1)}{2}]$. Include your codes and comments to your report.

7 Voice Synthesis from FSCs with Single Magnitude

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. Also include a plot covering the 10 periods of the waveform and comment on the waveform.

8 Voice Synthesis from FSCs with Zero Phase

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. Also include a plot covering the 10 periods of the waveform and comment on the waveform.