ECE2111 laboratory 3: signals in frequency domain

Prepared by: Dr. James Saunderson

Aim

The aims of this lab are

1. to buid basic skills in computing and plotting discrete-time Fourier series representations of finite duration/periodic discrete-time signals
2. to use these techniques to determine which telephone button was pressed given a noisy DTMF signal
3. to implement a simple audio compression scheme by replacing short segments of a signal with Fourier series approximation, and to understand the artefacts such a scheme introduces

Introduction

This lab is about frequency-domain representations of signals. The first parts of the lab introduce you to basic methods to compute and plot the discrete-time Fourier series of a discrete-time periodic signal (Section 1). In Section 2 you will use these tools to investigate the sounds made by telephones when you dial telephone numbers, also known as DTMF (dual-tone multi-frequency) signalling. In Section 3 you will see how to use frequency domain representations as the basis of a simple lossy compression scheme, allowing us to store an audio signal with significantly reduced storage.

Scheduling

This lab runs over two weeks (weeks six and seven).

Prelab

There are two prelab questions in Section 1. By the end of week 5 (for the precise due date see Moodle), read through the lab document, find the prelab questions, and answer them.

Submit your answers to the prelab questions via the Moodle quiz called 'prelab 3' on the Moodle page. You are expected to do this individually.

Results document

You are required to organize your code and outputs in what we will call a "results document". This must be created following the guidelines in the accompanying file "Formatting requirements for lab results document". Submit this by the end of week 7 (for the precise due date see Moodle) via the 'results document' assignment link on Moodle. **Your submission must reflect your own work!**

End-of-lab quiz

There is a timed, end-of-lab quiz that must be completed by the end of week 7 (for the precise due date see Moodle). **Please do not start this quiz until you are ready.** This quiz tests your understanding of the lab material. **It must be completed individually.**

1 Basic frequency domain facts and skills

If x is a discrete-time signal that is periodic with period N_0, it has a discrete-time Fourier series decomposition. This is an alternative representation of the form

$$x[n] = \sum_{k=0}^{N_0-1} X_k e^{jk\omega_0 n} \quad \text{for all } n \tag{1}$$

where $\omega_0 = 2\pi/N_0$. The coefficients X_k are called the discrete-time Fourier series coefficients. They can be computed via the formula

$$X_k = \frac{1}{N_0} \sum_{n=0}^{N_0-1} x[n] e^{-jk\omega_0 n} \quad \text{for all } k. \tag{2}$$

Note that the sequence X_k is also periodic with period N_0. Furthermore, the sums in (1) and (2) can be taken over *any period*, not just the period $0, 1, \dots, N_0 - 1$.

Suppose x is a discrete-time signal with *finite duration*, so that $x[n] = 0$ for $n < A$ and $n > B$. We can represent x in MATLAB by a vector `x` of length $N = B - A + 1$. The *periodic extension* (with period N) of x is a periodic discrete-time signal defined by

$$x_{\text{periodic}}[n] = \sum_{m=-\infty}^{\infty} x[n - mN] \quad \text{for all } n.$$

Note that $x_{\text{periodic}}[n] = x[n]$ for $A \le n \le B$. Now x_{periodic} has a discrete-time Fourier series decomposition

$$x_{\text{periodic}}[n] = \sum_{k=0}^{n-1} X_k e^{jk\omega_0 n} \quad \text{for all } n$$

where $\omega_0 = 2\pi/N$, and where

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk\omega_0 n} = \frac{1}{B - A + 1} \sum_{n=A}^{B} x[n] e^{-jk\omega_0 n} \quad \text{for all } k.$$

In particular,

$$x[n] = \sum_{k=0}^{n-1} X_k e^{jk\omega_0 n} \quad \text{for } A \le n \le B,$$

giving us an alternative description of the finite duration signal x for $A \le n \le B$.

We can compute the discrete-time Fourier series coefficients using the MATLAB command

```
X = (1/N)*fft(x);
```

Note that `X(k+1)` is the Fourier series coefficient X_k and that the Fourier series coefficients are *periodic*, satisfying $X_{k+N} = X_k$ for all k. Because of this periodicity, we often plot the discrete-time Fourier series coefficients over the period $X_{-\lfloor N/2 \rfloor}, \dots, X_{N-1-\lfloor N/2 \rfloor}$ which puts the DC coefficient X_0 roughly in the middle. Here the notation

$$\lfloor N/2 \rfloor = \begin{cases} N/2 & \text{if } N \text{ is even} \\ (N-1)/2 & \text{if } N \text{ is odd} \end{cases}$$

is the largest integer that is less than or equal to $N/2$. The magnitude of the Fourier series coefficients can be plotted in this way using the command

```
stem(-floor(N/2):(N-1-floor(N/2)), fftshift(abs(X)));
```

The phase of the Fourier series coefficients can be plotted in this way using the command

```
stem(-floor(N/2):(N-1-floor(N/2)), fftshift(angle(X)));
```

See the topic 5 notes and lecture slides for “Discrete-time Fourier series in MATLAB” for more information about how to plot the Fourier series coefficients in this way.

We can reconstruct the vector `x` from its discrete-time Fourier series coefficients `X` using the MATLAB command

```
x = ifft(N*X);
```

1.1 Prelab questions

Let $N = 8192$ and let

$$x[n] = \cos(2\pi(697/8192)n) + \cos(2\pi(1209/8192)n) \quad \text{for all } n.$$

Let X_k be the kth discrete-time Fourier series coefficient of x.

1. What is the fundamental period of x?
2. Find k_1 and k_2 satisfying $0 \leq k_1 < k_2 \leq N/2$ such that X_{k_1} and X_{k_2} are both non-zero.

1.2 Activities

In this part of the lab you will write a MATLAB script to compute and plot the spectrum of a number of discrete-time signals of finite duration.

1. Let x be a continuous-time signal of the form

$$x(t) = \cos(4\pi t).$$

   Let x_s be the discrete-time signal obtained by sampling x with sampling frequency 160 samples per second, and let

$$z[n] = x_s[n](u[n] - u[n-80]) \quad \text{for all } n.$$

   Make a stem plot of $z[n]$ vs n for $0 \leq n \leq 79$.

2. Let z_{periodic} be the periodic extension (with period 80) of z. Plot $z_{\text{periodic}}[n]$ vs n for $0 \leq n \leq 159$. Find a simple formula for $z_{\text{periodic}}[n]$ as a function of n.

3. Compute the discrete-time Fourier series coefficients of z_{periodic}. Plot the magnitude of the discrete-time Fourier series coefficients of z_{periodic}. Include one period of the coefficients in your plot, and center the plot at $k = 0$.

4. Now let w be the signal

$$w[n] = x_s[n](u[n] - u[n-70]) \quad \text{for all } n.$$

   Make a stem plot of $w[n]$ vs n for $0 \leq n \leq 69$.

5. Let w_{periodic} be the periodic extension (with period 70) of w. Plot $w_{\text{periodic}}[n]$ vs n for $0 \leq n \leq 139$. What is the difference between the plots of w_{periodic} and z_{periodic}?

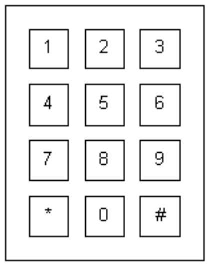
6. Compute the discrete-time Fourier series coefficients of w_{periodic}. Plot the magnitude of the discrete-time Fourier series coefficients of w_{periodic}. Include one period of the coefficients in your plot, and center the plot at $k = 0$. How does the plot differ from the plot of the magnitude of the discrete-time Fourier series coefficients of z_{periodic}? Why do you think this is the case?

Once you have finished this section:

- copy your code into your results document
- copy the plots from questions 1–6 into your results document.
- Answer questions 2 (the formula for z_{periodic}) and questions 5 and 6 in your results document.

2 DTMF signalling

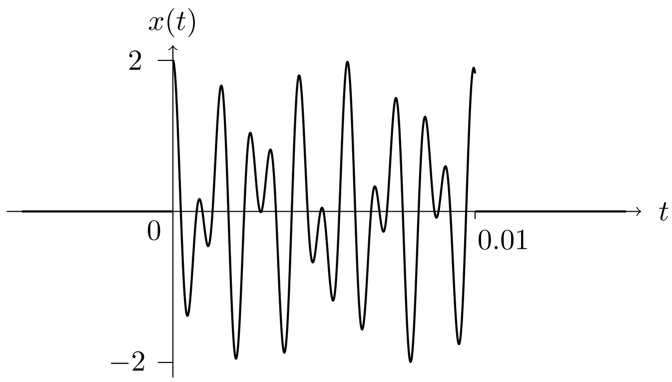
If you get out your phone, turn the sound on, and dial a number, the phone will play sounds corresponding to the button presses. These sounds are encoded using DTMF (dual-tone multi-frequency) signalling.



The basic idea of DTMF signalling is that when a button on a telephone is pressed, a signal that is the sum of 'tones' at two different frequencies is produced. For example, when you press the key 1 from time t_1 to time t_2, the sound produced (ideally) has the form

$$(\cos(\omega_{r1}t) + \cos(\omega_{c1}t))\,[u(t - t_1) - u(t - t_2)].$$

A plot of a signal of this form (with $\omega_{r1} = 2\pi \times 697$ and $\omega_{c1} = 2\pi \times 1209$ and $t_1 = 0\,\mathrm{s}$ and $t_2 = 0.01\,\mathrm{s}$ is shown below.



The frequencies are labeled ω_{r1} and ω_{c1} because 1 is in the first row and the first column of the telephone dial-pad. In general, in dual-tone multi-frequency signalling, the key in row i (for $i = 1, 2, 3, 4$) and column j (for $j = 1, 2, 3$) corresponds to a tone of the form

$$(\cos(\omega_{ri}t) + \cos(\omega_{cj}t))\,[u(t - t_1) - u(t - t_2)].$$

In the lab we will work with sampled versions of these signals, with sampling frequency $F_s = 8192$ samples/second and sampling period $T_s = 1/F_s$ seconds. In this case, when we press key 1 from time n_1 to n_2 the sound produced gives the discrete-time signal

$$x_1[n] = (\cos(\omega_{r1}T_s n) + \cos(\omega_{c1}T_s n))\,(u[n - n_1] - u[n - n_2])\,.$$

2.1 Activities

In this part of the lab you will be given DTMF signals and asked to determine which telephone button was pressed for each signal. You will do this first for signals with no noise, and then for signals with noise added to them. This will illustrate that the frequency domain methods we are using are quite robust to noise.

1. Download the folder `lab3files` from the ECE2111 Moodle site. Load the file `dtmfclean.mat`. Signals `xdtmf1`, `xdtmf2`, `xdtmf3`, `xdtmf4`, `xdtmf7`, `xdtmfa` will appear in your workspace. The signal `xdtmf1` is a length 8192 signal obtained by sampling (at sampling frequency $F_s = 8192$ samples/second) the DTMF signal corresponding to pressing the button '1' for 1 second. The other signals are obtained in a similar way. (Here `xdtmfa` corresponds to the astersk key `*`.)

2. For each signal `xdtmf1`,...,`xdtmfa`:
   (a) Listen to the signal using `soundsc`.
   (b) Plot the magnitude of the discrete-time Fourier series coefficients of the signal. Use a stem plot. Arrange the horizontal axis so that the zero frequency component is shown in the center of the plot. Label the axes, and add a title to the plot.
   (c) Find the indices of the discrete-time Fourier series coefficients of the signal that have large magnitude.

3. Determine the frequencies ω_{ri} for $i = 1, 2, 3, 4$ and ω_{cj} for $j = 1, 2, 3$.

4. Now load the file `dtmfnoisy.mat`. A signal `ydtmf` will appear in your workspace. What is its duration?

5. Listen to the signal using `soundsc`. Can you hear the noise?

6. Plot the magnitude of the discrete-time Fourier series coefficients of `ydtmf`. Use a stem plot. Arrange the horizontal axis so that the zero frequency component is shown in the center of the plot. Label the axes, and add a title to the plot.

7. Find the indices of the discrete-time Fourier series coefficient of `ydtmf` that have large magnitude.

8. Which button press does `ydtmf` correspond to? (Pay careful attention to the duration of the signal when interpreting the frequency components.)

Once you have finished this section:

- copy your code into your results document
- copy the plots from questions 2(b) and 6 into your results document.
- copy the frequencies from 3 into your results document.
- answer the questions in 4 and 7 and 8 in your results document.

3 Audio compression

In this part of the lab you will approximate a long audio signal, sampled at a high sampling rate, with a signal that has a much more concise frequency domain representation. In doing so you will perform a sort of lossy (i.e., involving approximation) compression. The basic idea is that audio signals are often approximately periodic (with fairly simple frequency domain representations) over short time periods.

3.1 Activities

Write a MATLAB script to carry out the following tasks.

1. Load the signal `trumpet.wav` using the command

   ```
   [y,Fst] = audioread('trumpet.wav');
   ```

   as in lab 2.

2. At the sampling rate `Fst`, how many samples correspond to 30 ms of audio? Let `M` denote this number of samples. Let `P = length(y)/M`.

3. Use the MATLAB command `reshape` to rearrange your signal `y` into the form of a `M` by `P` matrix called `yseg`. The ith column of the matrix corresponds to the audio from the ith 30 ms segment.

4. Create a `M` by `P` matrix `Yseg` such that the ith column of `Yseg` corresponds to the discrete-time Fourier series coefficients of the ith column of `yseg`.

5. Compute `maxval`, the maximum absolute value of a Fourier series coefficient appearing in `Yseg`.

6. Let `threshold=0.01` for instance. Make a new array of Fourier series coefficients by keeping only those coefficients from `Yseg` with magnitude larger than `threshold*maxval`. One way to do this is to use a command of the form

   ```
   Ysegtrunc = Yseg.*(abs(Yseg) > threshold*maxval);
   ```

   Explain how this command works.

7. Use the command `Ysegtrunc = sparse(Ysegtrunc);` to convert to a sparse storage format in MATLAB. By typing `whos Ysegtrunc` find out how much memory this variable requires. Compare this with the memory required to store `y`.

8. Now we convert our frequency-domian representation back into a time-domain representation (essentially we are uncompressing our audio segment). First we convert `Ysegtrunc` back from sparse data format to normal data format with `Ysegtrunc = full(Ysegtrunc);`.

   (a) Convert each column of `Ysegtrunc` back into a time-domain signal using `ifft` separately on each column (see earlier in the lab).

   (b) Reshape the signal back into a single vector of length `M*P` using `reshape`. Call the resulting signal `ytrunc`.

   (c) Use `soundsc` to play `ytrunc` and compare it with `y`. What differences do you hear?

   (d) There are clearly undesirable artefacts in the reproduced signal. Choose a time-interval on which `ytrunc` and `y` are both non-zero, that crosses a 30ms segment boundary. Over this time-interval, plot both signals. What do you notice? Plot the difference between `ytrunc` and `y`. What do you notice? What do you think is the basic problem with our compression method?

Optional: How could we improve this method so that the artefacts identified in question 8(d) are less pronounced?

Once you have finished this section:

- copy your code into your results document
- copy the plot from question 8(d) into your results document.
- answer questions in 2, 6 (explanation of the command), 7 (memory requirements), 8(d) discussion of artefacts in your results document.

Once you have completed the lab tasks and understand them, you are ready for the end-of-lab Moodle quiz. This quiz is closely based on the lab tasks. It must be completed **individually**.You may use MATLAB. Unlike the prelab, you only have **one attempt at each question**.

Assessment

This lab is marked out of 12. Your mark is based on the following:

- **Prelab:** Correct responses to the two prelab questions, submitted via the prelab Moodle quiz (3 marks, 1 for question one, 2 for question two)
- **Results document:** (4 marks, 3 marks for content and 1 mark for presentation)
  Marks per section: Each of the 3 sections is marked out of 1 (3 marks):
  - 0 marks if not attempted
  - 0.5 marks if a reasonable attempt is made, but has clear flaws
  - 1 mark if no errors or possibly very minor errors

  Presentation: (1 mark)
  - Results document adheres to the formatting requirements (1 mark)
  - Results document mostly adheres to formatting requirements, but not completely (0.5 mark)
  - Results document rarely adheres to formatting requirements (0 marks)
- **End-of-lab quiz:** Correct responses to the end-of-lab Moodle questions (5 marks, 1 per question).