EECS 16A Designing Information Devices and Systems I
Fall 2015 Anant Sahai, Ali Niknejad Homework 14

This homework is due December 7, 2015, at Noon.

1. **Homework process and study group**

   Who else did you work with on this homework? List names and student ID's. (In case of hw party, you can also just describe the group.) How did you work on this homework?

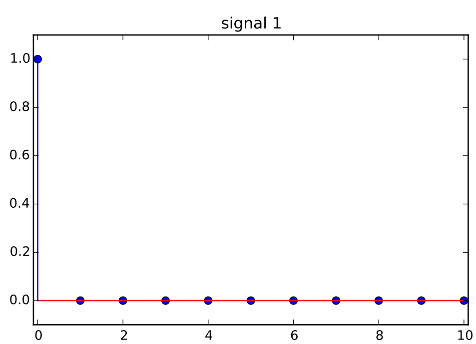
2. **Mechanical DFT**

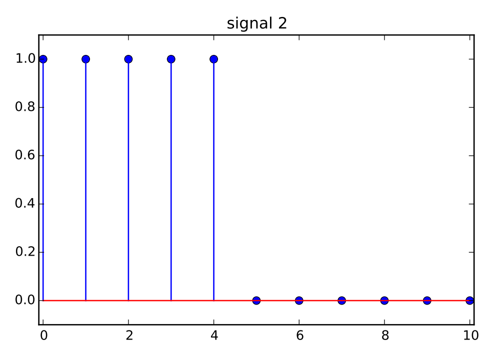
   The DFT (Discrete Fourier Transform) is both the name of the orthonormal basis of complex exponentials, as well as for the linear transformation that maps vectors from the standard basis to that basis (frequency domain). When we say "find the DFT" in a signal processing context, this is shorthand for either "find the coordinates in the DFT basis" or "find the eigenvalues of the circulant matrix for which the given signal is the first column" — where the right choice is made by context. In many cases, it doesn't matter which choice is made since the two are related by a factor of $\sqrt{n}$ anyway.

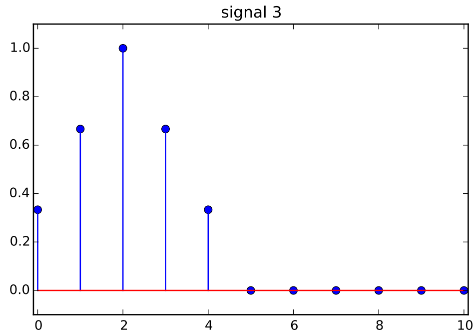
   Because the DFT basis is fundamentally complex, usually, the representation of a signal in that basis will have both real and imaginary components. It is traditional to view those coordinates in polar form — in terms of their magnitude and angle (which is usually called "phase"). This problem will help you see why this is traditional by exploring what happens to in the frequency domain if you perform a circular shift on a signal in time domain.

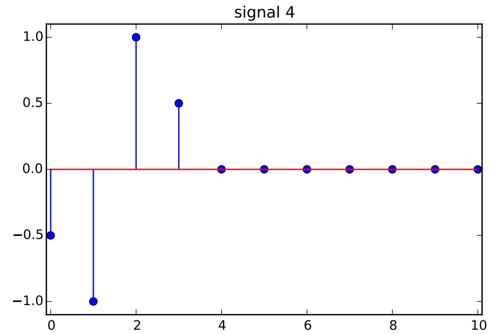
   Find the DFTs relating to following 4 signals that have length 10. For each signal, find the DFT of the signal itself (circular shift of 0), its circular shift by 3 and circular shift by 6. Plot the amplitude and phase of their DFTs separately for each of these shifts. You can use the accompanied IPython Notebook.

   For your reference, to circularly shift a vector $\vec{v}$ by 3 you can use `np.roll(v,3)`.









3. Beats by 16A

A pure tone signal can be represented as $A\cos(2\pi f t)$, where f is the frequency of the tone and A is the amplitude of the tone. If the signal we are dealing with is a sound then the amplitude corresponds to the volume. Most of us are natively unable to tell the difference between frequencies that are very close to each other, e.g. 552 and 546 Hz, if we were to listen to tones generated at these frequencies one after the other. (Trained musicians can be different.) However, if the sounds reach our ears simultaneously, what we hear is a sound at a frequency which is the average of the two frequencies. Furthermore, we hear a variation in the amplitude (volume) of the sound, and this variation corresponds to the difference between the two frequencies. This is referred to as the beat effect, and the frequency of the amplitude variation is the beat frequency. We will explain this effect shortly, but first play around with it in the Ipython Notebook.

So how does this work? First, take a look at the trig identity

$$\cos(2\pi f_1 t) + \cos(2\pi f_2 t) = 2\cos\left(2\pi \frac{(f_2 - f_1)}{2} t\right)\cos\left(2\pi \frac{(f_1 + f_2)}{2} t\right) \tag{1}$$

If f_1 and $f_2 > f_1$ are close together, then $\frac{f_2+f_1}{2}$ is much greater than $\frac{f_2-f_1}{2}$. Therefore, the expression $2\cos\left(2\pi\frac{(f_2-f_1)}{2}t\right)\cos\left(2\pi\frac{(f_1+f_2)}{2}t\right)$ can be thought of as a pure tone at frequency $\frac{f_2+f_1}{2}$ where the amplitude (volume) is itself changing at a much lower frequency of $\frac{f_2-f_1}{2}$.

(a) Derive the trig identity in equation (1). Hint: You should take advantage of one of the consequences of Euler's theorem, in particular $\cos(\theta) = \frac{e^{i\theta}+e^{-i\theta}}{2}$, where $i = \sqrt{-1}$.

(b) Consider two signals $x(t) = \cos(2\pi 101 t)$ and $y(t) = \cos(2\pi 99 t)$ that are simultaneously received at a receiver. Predict the frequency of the beats that you will hear. Does this agree with what you hear in the IPython notebook? Why or why not? (Hint: what is the correspondance actually between amplitude and perceived volume? Can there be negative volume?)

(c) It turns out the beat effect is what police radar guns use to calculate the speed of a car. Consider a radar gun that sends out a pulse $x(t) = \cos(2\pi\omega t)$. This is incident on a car that is moving towards the radar gun (the detector) with velocity v, and reflects back at the receiver. Recall that because of the Doppler effect the reflected wave will have a shift in frequency. If the original frequency of the wave was f, the the reflected wave will have frequency $f' \approx f\frac{c}{c-v}$, where c is the speed of light. What is the "beat frequency" that the radar gun will detect? How can you use this to estimate the speed of the car?

4. DFT and Compression

As we've discussed previously, sound is composed of a sum of complex exponentials (sinusoids) $\sum e^{i\omega t}$. You can think of sound as having two different representations: a time domain representation and a frequency domain representation. At each point in time, the complex exponentials that make up the music are evaluated to a number, which produces what you hear. You can literally feed in a vector of numbers into an audio player in iPython and get a sound out. You can also represent a sound by the frequencies that make it up. Knowing the frequencies, you can always generate the same sound signal.

Time domain is like a recipe, where you're told at each step what to put in the dish. Frequency domain is like a grocery list, where you're told in no particular order what ingredients make up the dish.

What if your signal was a pure tone, some $e^{i\omega t}$? In time domain, you would have to have a vector the length of the signal, for example length 100 for 100 seconds. In frequency domain, you could capture all the information of the signal in one number: ω. It turns out, most music is composed of dominant frequencies, with a lot of minor frequencies. Even if you were to get rid of those minor frequencies, the music signal would sound pretty much the same. This is useful for what is called "lossy" compression: rather than truncating the length of the song, you can keep the same song with roughly the same quality by transmitting fewer frequencies!

(a) In the file prob14.ipynb, play the song provided and plot the DFT (frequency domain representation.) Comment on what you see. What does the plot look like? Does it look like there are a lot of high frequencies? Low frequencies?

(b) Set 50, 80, 99 and 99.9 percent of the relatively unimportant (smaller in magnitude) frequencies to 0. What do you notice as you set more frequencies to 0? Experiment to see what fraction of small frequencies you can throw away before hearing a difference in the music.

(c) For 50, 80, 99 and 99.9% of song removed, plot the corresponding time and frequency domain representations of the song. What do you see? Does it look similar to the original song? When you can hear a difference in the music, what do you notice about the frequency/time plots?

5. OFDM

The current state-of-the-art systems for modern wireless communication tend to be based on an idea called "OFDM" which stands for Orthogonal Frequency Division Multiplexing. The core idea behind OFDM is to use the DFT Basis we found in class to carry information across a wireless channel that might have echoes in it. This problem is designed to introduce some of the core ideas of OFDM while also reinforcing your understanding of the DFT.

Throughout this problem, we have a wireless radio channel that has a direct signal with complex amplitude h_0 that is received with "no[1] delay", and a few echoes that are received after i time steps, with that echo being multiplied by complex[2] amplitudes h_i. Here, i ranges up to a small number $k-1$. (Think $k = 4$ to 40 or so.)

So, if the trasmitter transmitted the sequence $0,1,0,2,0,0,\ldots$, the receiver would receive $0,h_0,h_1,2h_0+h_2,2h_1+h_3,2h_2+h_4,\ldots,2h_{k-3}+h_{k-1},2h_{k-2},2h_{k-1},0,0,\ldots$. At some point, all the echoes would be over.

[1] In wireless systems used for communication, "no delay" is used to refer to a delay corresponding to light traveling on the direct path between the transmitter and receiver.

[2] Real-world echoes have complex amplitudes because of something called "phase" in electromagnetic waves. Echoes can shift the phase of the carrier waves in addition to attenuating the intensity of the reflection relative to a direct line-of-sight path.

(a) Let's start with understanding the most basic form of proto-OFDM. Suppose that we choose a song-length $n > k$ and we choose to transmit n messages α_ℓ for ℓ ranging from 0 to $n-1$ by transmitting the vector $\vec{x} = \sum_{\ell=0}^{n-1} \alpha_\ell \vec{f}_{\ell,n}$ where the vectors $\vec{f}$ are the DFT Basis vectors. Namely, $\vec{f}_{\ell,n}$ is the ℓ-th Fourier basis element for n-dimensional complex space $\mathbb{C}^n$ with t-th entry $f_{\ell,n}[t] = \frac{1}{\sqrt{n}} \exp\left(it\frac{\ell 2\pi}{n}\right)$ where $i = \sqrt{-1}$. In column notation, $\vec{f}_{\ell,n} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ e^{i\frac{\ell 2\pi}{n}} \\ e^{i2\frac{\ell 2\pi}{n}} \\ e^{i3\frac{\ell 2\pi}{n}} \\ \vdots \\ e^{i(n-1)\frac{\ell 2\pi}{n}} \end{bmatrix}$.

Then, the vector $\vec{x}$ is transmitted over and over again periodically by the transmitter. The receiver listens to one period of the vector, say from time n to $2n-1$ and calls that $\vec{y}$.

Express $\vec{y} = H\vec{x}$ where H is an appropriate matrix to represent the action of the echoes in the wireless channel.

(b) Continuing the previous part, how would you recover α_i from $\vec{y}$? Show that this can be done by taking an appropriate inner product. Show that the inner product gives you $\lambda_i \alpha_i$ where λ_i is the appropriate eigenvalue of the matrix H.

(c) You decide that it is a bit silly to keep transmiting the vector $\vec{x}$ over and over again given that you are just going to listen to what happens from time n to $2n-1$. Argue why it is safe to just transmit the vector $\vec{x}$ twice — once starting at time 0 and then again starting at time n.

(d) Up till this point, we have been assuming that we already knew H at the receiver. Suppose that we didn't have access to this. **Argue why the following strategy could be made to work.** We first transmit $\vec{z} = \sum_{\ell=0}^{n-1} \vec{f}_{\ell,n}$ twice (starting at times 0 and n) and then transmit $\vec{x}$ twice after that (starting at times $2n$ and $3n$). We then listen twice. First to get $\vec{r}$ from time n to $2n-1$ and then to get $\vec{y}$ from time $3n$ to $4n-1$. We use $\vec{r}$ to recover the matrix H (**How can we do this?**) and then use $\vec{y}$ to get back all the α_i that were transmitted. (**How can we do this?**)

(e) You think some more and decide that you want to take advantage of the fact that the duration of the echos k is significantly smaller than n. So, you decide to just transmit the last $k-1$ entries of $\vec{x}$ immediately before transmitting $\vec{x}$. This is called a "cyclic prefix." In other words, you transmit $x[n-k+1], \ldots, x[n-1]$ first, starting at time 0 and then at time $k-1$, start transmitting $\vec{x}$ itself until time $n+k-1$. The receiver records from time $k-1$ to $n+k-1$ and calls the result $\vec{y}$. **Argue why $\vec{y} = H\vec{x}$ where H is the same matrix that you had in the first part of this problem.** Why is it that as far as the receiver is concerned, the transmitter might as well have been repeating $\vec{x}$ forever?

(f) You decide that you don't want to do a second transmission to figure out what H is and want to integrate figuring out H with figuring out the messages α_i. You do a simple calculation: $\vec{y}$ has n dimensions. There are k dimensions of uncertainty in H. So we can reasonably hope to extract no more than $n-k$ complex numbers as a message from the transmitter if we are also going to be figuring out H. Let us try the simplest possible strategy. We set the first k coefficients α_i to be 1. These are called "pilot[3] tones." The other $n-k$ coefficients are set to carry message content. At the receiver, the first thing to do is to recover the first k eigenvalues of H. **Describe how you would do this and why this works?**

[3]Why are they called "pilot tones?" The "tone" part is easy — these are complex exponentials and hence pure frequencies or tones. The "pilot" might feel more confusing. This is not related to an airline pilot, but more to the "pilot light" in your water heater or gas stove. These are small flames that let you start bigger ones. Or "pilot episode" to mean the first prototype episode used to start a TV series.

(g) With the first k eigenvalues $\lambda_0, \dots, \lambda_{k-1}$ in hand, we would like to figure out the unknowns $h_0, h_1, \dots, h_{k-1}$. **Write a system of linear equations solving which will give you the unknowns. Use the attached IPython notebook to see that these equations are solveable.**

(h) Once you have all the $h_0, h_1, \dots, h_{k-1}$, how would you recover the messages themselves? Use the attached IPython notebook to verify that your procedure works.

(i) (Bonus — Out of Scope) Now, you realize that actually, you could've chosen any k tones to be your pilot tones — you didn't have to choose the first few. So what makes a choice better or worse? To explore that, we can use simulations that include a little bit of noise. By noise, we just mean received signals that are extraneous and did not come from echoes of the transmitted signal of interest. The other thing we can do is look at the eigenvalues of the inverse matrix of the matrix defining the equations above. The bigger these eigenvalues are, the worse the effect of noise will be. What is the best selection of k tones that you can find?

(j) (Bonus — In Scope) Suppose that in order to further combat noise, you decide to do least-squares recovery on all the λ_i based on your observations of the pilot tones. Here, you can assume that you assign $2k$ pilot tones. Set up the problem and show how you would solve for the best estimates for $\lambda_0, \lambda_1, \dots, \lambda_{n-1}$ given your $2k$ measurements of the pilot tones. *(HINT: the DFT basis is orthonormal. So it preserves both angles and lengths. How can you exploit this?)*

6. Your Own Problem Write your own problem related to this week's material and solve it. You may still work in groups to brainstorm problems, but each student should submit a unique problem. What is the problem? How to formulate it? How to solve it? What is the solution?