EE 102A - Assignment 8

Nicholas Dwork

Problem 1. The vector DFT$\{x\}$ is imaginary and odd. What value is the value of $x[0]$?

Problem 2. Consider a function $f : [-L/2, L/2] \to \mathbb{C}$. The function f is isomorphic to its periodic extension $f_{p.e.}$. As we've discussed, there is no information gained or lost by working with the periodic extension rather than the original function. We can compute the Fourier Series of f and we can compute the Fourier Transform of $f_{p.e.}$. Are the two related? If so, how?

Problem 3. *Nuclear Magnetic Resonance (NMR) - with help from David Zeng*
Nuclear Magnetic Resonance is a physical process where atoms absorb magnetic energy and release it in a characteristic way (that depends on the atomic structure of the molecule containing the atoms). The figure below shows a Varian Unity Inova 900 MHz, 21.1 Tesla NMR spectrometer.

Each atom absorbs energy at a specific frequency, called the Larmor frequency of the atom. That frequency may be altered slightly due to the nearby atoms in a molecule that affect the local magnetic field; this is called chemical shift. Once the energy is absorbed, the atoms begin emitting the absorbed energy in the form of an electromagnetic wave at that atom's particular frequency (Larmor frequency + chemical shift). By observing the frequencies emitted, we can use NMR to identify the molecules in a sample.
The process of NMR characterization is to emit a magnetic wave with a spectrum encompassing all molecules of interest. Some of this energy gets absorbed by the atom and re-emitted. An antenna is used to capture the signal emitted by the sample.

Download the data at the following address:

`www.stanford.edu/~ndwork/teaching/1706ee102a/hmwk/data/NMRSpec.mat`

The variable `st` contains an NMR signal; the data sampling rate is 2000 samples/s. Plot the magnitude and phase of the NMR spectrum (versus frequency). Can you distinguish the peaks corresponding to Creatine (around 209 Hz) and Choline (around 185 Hz)?

Problem 4. *You're Pushing My Buttons! - with help from David Zeng*

(Not that) long ago, telephones were made out of analog hardware components. Initially, people would talk into the phone and ask the operator to connect them to the party they wanted to talk to. Eventually, a system of buttons was used to designate the receiving party. The figure below shows a phone from the 1980s that might have been used for this purpose.

	1209	1366	1477
697	1	ABC 2	DEF 3
770	GHI 4	JKL 5	MNO 6
852	PQRS 7	TUV 8	WXY 9
941	*	0	#

But how could the switchboard tell which buttons were pushed? The solution was that each button would emit a different sound. So telephones could have been designed with 12 oscillators in the phone, issuing a different tone for each button. To save on hardware, though, the choice was made that each column would power the same oscillator, and each row would power a different oscillator. The frequencies corresponding to each oscillator (in Hz) are shown in the image above. For example, if the number 4 were pushed on the phone, the signal emitted would be

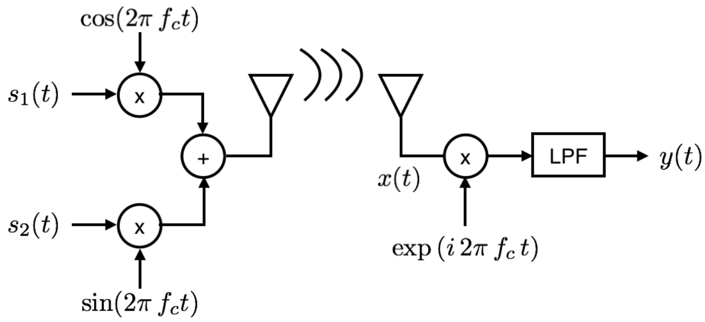
$$s(t) = A\cos(2\pi\, 770\, t) + A\cos(2\pi\, 1209\, t),$$

where $A \in \mathbb{R}$ is some unknown amplitude and t is time in seconds. So phones were made with 7 oscillators rather than 12.
For legacy reasons, we still transmit those frequencies when we push buttons on our phones. If you ever get to a menu where they say something like "Push 1 to schedule an appointment, push 2 to hear details about your account, push 0 to speak to an operator", they are waiting to hear the appropriate tones to understand which button you've pushed.
And thus we arrive at the problem. Download the following file of data.

`www.stanford.edu/~ndwork/teaching/1706ee102a/hmwk/data/DialNumber.mat`

You can load this data into Matlab using the `load` command. The variable `dialTone` contains the sampled signal where someone is entering a 10 digit sequence, and the variable `Fs` contains the sampling rate. Identify the sequence of buttons pushed.

Problem 5. *Quadrature Amplitude Modulation*
Suppose you want to send a band-limited complex signal; this is equivalent to sending two real messages (s_1 and s_2) each with the same band-limit (corresponding to the real and imaginary parts of the signal). This can be done with Quadrature Amplitude Modulation, depicted in the figure below.



The low-pass filter has a cutoff frequency equal to the band-limit of the two functions. (Assume an ideal low-pass filter throughout this problem.)
a) Find an expression for y.
b) Show that you can recover s_1 and s_2 from y.
c) Suppose that the receiver isn't in phase with the transmitter (i.e. instead of multiplying by $\exp(i2\pi f_c t)$, the receiver multiplies by $\exp(i\,2\pi f_c(t-\Delta))$ for some unknown Δ. Can you still recover s_1 and s_2? If not, does the difference matter?

Problem 6. *Radar*
RADAR is a system used to measure the distance to objects; it stands for RAdio Detection And Ranging. A directional antenna is used to send an electromagnetic wave of a known form in a given direction. This burst of electromagnetic energy is called the "pulse." The radar then uses the same antenna to listen to the electromagnetic spectrum in the vicinity. If there's an object in the direction that the wave was emitted, the energy will reflect off of that object and come back at the antenna. The data is processed to see if any portion of the waveform return looks similar to the emitted waveform; if it looks similar enough, then a detection is registered at that distance. You will create an algorithm to perform radar detection in this problem.

Download the data from the following link. The variable `template` is the waveform emitted by the antenna. The variable `sig` is the signal received by the antenna after the initial emission; it is known as the "radar return." The data was sampled at 5 GHz.

`www.stanford.edu/~ndwork/teaching/1706ee102a/hmwk/data/radarReturn.mat`

So how do we perform this detection? We need to search for the template in the radar return; our problem is noise; there's a whole bunch of stuff there that isn't our return, so we've got to find our signal in the data.

One way to do this might be to hypothesize that the return is at index 1 in the radar return; with this hypothesis, you perform a point-wise multiplication of the template (of length N) with the first N elements of the radar return and add up all the values attained. Then you hypothesize that the data is at the second index, and perform a similar operation. If ever there's a point where this value is high, then you might think that the location indicated is the return of the emitted signal. But what is this operation exactly? It's a cross-correlation! It can be expressed as

$$(t \star r)[n] = \sum_{m=0}^{M} \bar{t}[m]\, r[n+m],$$

where t is the template, $\bar{t}$ is the conjugate of t, and r is the radar return.

And we're well on our way there. It looks like the cross-correlation will be instrumental in our process of detection. There are two problems remaining: (1) if there's a spurrious high value in the return (if there just happens to be a spot with a lot of noise) then our algorithm will assign a lot of weight to that location even though it's just noise, and (2) we don't know what values correspond to a good match. We would prefer an algorithm that doesn't get confused with high values of data, and where the thresholds are on an absolute scale.

For that, we turn to the Pearson Correlation Coefficient (PCC). PCC is defined for two populations x and y, each with N elements, as follows:

$$PCC = \frac{\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\mu_x)^2}\sqrt{\sum_{i=1}^{N}(y_i-\mu_y)^2}},$$

where μ_x, μ_y are the mean values of x and y.

The PCC has a maximum value and a minimum value. And so the threshold we choose for a detection is independent of the values in the return itself. And there lies our answer: rather than computing the dot product of the template and the return at each index of the return (called cross-correlation), compute the PCC at each index of the return (called Normalized Cross Correlation). Set an appropriate threshold, and make detections.

a) What are the maximum and minimum values of the Pearson Correlation Coefficient. When does it attain its maximum? When does it attain its minimum?
b) Use the FFT to implement a fast normalized cross correlation and detect the radar return. Use subplot to plot the original signal as well as the result of the normalized cross correlation.
c) How far away was the object detected?
d) Explain this detection algorithm in your own words. Why does it work? Any ideas on how you might improve the algorithm?

Note: searching for a template in a signal in this way is called "Matched Filtering." The algorithm shown is for the special case when the noise is white.