Lab 2

Sampling

0. Preface

Now that you have learned MATLAB to a sufficient and necessary level, in this lab you will use MATLAB to explore sampling.

Your lab report should answer all questions in all sections.

1. Are you down with DFT? How about the FFT?

The discrete Fourier transform (DFT), denoted $X[k]$, for a finite-length sequence $x[n]$ of length N is defined over $0 \le n \le N-1$ and given by:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad 0 \le k \le N-1 . \tag{1}$$

It is the discretized analog of the (continuous) Fourier transform for a continuous time function $x_a(t)$. The sequence $x[n]$ is created by sampling signal $x_a(t)$ at a sampling rate (sampling frequency) $F_s = 1/T_s$: $x[n] = x_a(nT_s)$, for some range of n. The significance of the DFT is apparent when writing it in the following way:

$$X[k] = \sum_{n=0}^{N-1} x_a(nT_s) e^{-j2\pi \frac{kF_s}{N} nT_s}, \quad 0 \le k \le N-1 . \tag{2}$$

It can be seen in equation (2) that the samples of $x[n]$ are associated with *N* samples (*n=0,…,N-1*) of *N* sampled complex sinusoids (*k=0,…,N-1*) that have different angular frequencies. The value of $X[k]$ is the (complex) amplitude of the k^{th} "frequency bin"; the corresponding frequency of this bin expressed in radians per second is $\omega_k = 2\pi \frac{kF_s}{N}$, where the sampling rate is expressed in Hz. In particular, $X[0]$ is the zero-frequency, or D.C. (direct current) term; $X[1]$ is the frequency at $\omega_1 = 2\pi \frac{F_s}{N}$ Hz, etc. For real signals, $X[k]$ is “meaningful” only for $0 \le k \le N/2+1$.

> 1.0 Explain what should be understood by the word “meaningful”, why sample values with index *k* > *N*/2 +1 are not meaningful, and why the right hand side shows *N*/2 + 1 instead of just *N*/2.

All other k signify negative frequencies, and their values are the conjugates of those of the positive frequencies.

You will now perform and feel immensely proud of doing your first DFT, using a fast Fourier transform (FFT)—which is an *implementation* of the DFT. A FFT does the DFT, in other words. (Note that FFT is the name for an entire *group* of algorithms, not just a single unique method.)

1.1 Sample a sinusoidal signal with amplitude 1 and frequency of 128 Hz at a sampling rate of 2,048 Hz indexed by $0 \le n \le N-1$, where $N = 32$. This means sampling the following:

$$x[n] = \sin(2\pi 128n/2048), \tag{3}$$

at the points n=[0:31]. Plot the result using `stem`. Include your code and plot in your report.

1.2 Now find the DFT of the sequence you created in 1.1, using the MATLAB routine `fft()` (just call `fft()` with the sequence as an argument). Plot the magnitude spectrum of the result, `abs(X)`, versus the corresponding actual frequency (in Hertz and not k). Plot only the frequencies between 0 and Fs/2. Your plot should look like Figure 1. This means using `stem` instead of `plot`. Include your code.

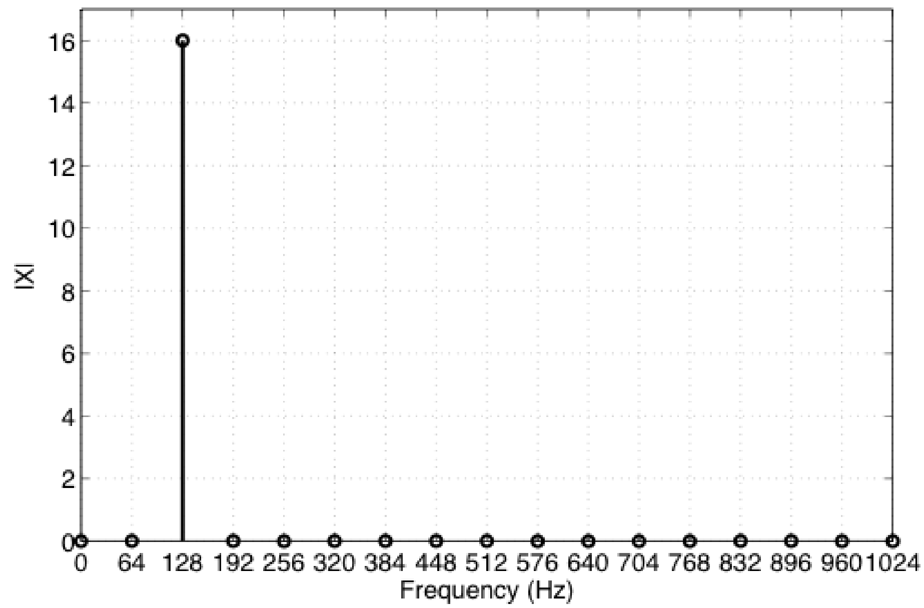

Figure 1: The magnitude spectrum of a real 128 Hz sinusoid sampled at 2048 Hz over $0 \le n \le 31$

1.3 What is the phase at 128 Hz? Find this using the `angle` routine on the relevant value of *X*[*k*]. Why does this answer make sense, i.e., explain your reasons how you can get to this result without using MATLAB [Hint: start from the corresponding continuous signal.] What does the sign of the phase tell us?

1.4 Now sample a sinusoid of 220 Hz at the same sampling rate given in 1.1. Create a 32-sample sequence and plot this versus *n* using `stem`. Include the plot in your report, but not your code.

1.5 Draw a picture of what you hypothesize will be the *magnitude* DFT of this signal. Include in your lab this hypothesized and expected picture. Explain why you drew what you did.

1.6 As in 1.2, find the DFT of this signal. Plot the magnitude of the result versus frequency. Include this plot in your report. How does this plot compare to that in 1.2? Was your hypothesized plot, which you created in 1.5, correct?

1.7 Verify that you get the exact same time-domain signal from this spectrum by taking an inverse DFT using the `ifft()` routine. Plot the resulting time-domain sequence. Is your result real or complex valued? Explain why.

Now what the heck is going on? The input is a "pure" sinusoid at 220 Hz, but the DFT of this new signal, evaluated in 1.6, shows that all frequencies that are possible with 32 uniformly spaced samples are present. Why?

There are two reasons for this, both of which are related. First, since the only frequencies that can be represented by a DFT of length-32 and sampling rate of 2,048 Hz are integer multiples of $f_k = \frac{2048}{32} = 64\ \text{Hz}$, a sinusoid with a frequency of 220 Hz is in between two frequency bins. Second, looking at the differences between the time-domain signal created in 1.1 and that created in 1.4, one can see in the latter that there is not an integer number of periods contained in the 32 samples. Thus, when this sequence is abruptly cut off, a discontinuity is created. This discontinuity is responsible for the extra frequencies when the DFT periodically extends the signal, at both ends, to +/- infinity. This effect, when a frequency "falls in a crack," is colorfully called "spectral leakage" (Figure 2).

Figure 2: A Leicester County worker listens for spectral leakage

Now we are going to perform a trick. If the frequencies of the DFT are integer multiples of $f_k = \frac{kF_s}{N}$, then one way to make the spacing of the bins closer together is to increase N. So instead of taking 32 samples of the sinusoid in equation (3), we can generate 512 samples, which gives a frequency spacing of 4 Hz. Theoretically, we could then get rid of spectral leakage for our sinusoids of 220 Hz because it falls right inside bin number 56. Let's demonstrate this now.

1.8 Create a 512-length sequence of the sinusoid in equation (3) with a frequency of 220 Hz, amplitude of 1, and a sampling rate of 2,048 Hz. As in 1.2, find the DFT of this signal and plot the magnitude of the result versus frequency. Include this plot in your report.

Problem solved? Well, what if we only had the 32-sample version created in 1.4 and there was no way to extend it or to get more data? We will return to this problem in a later lab called "windowing."

1.9 Referring back to the plot you made in 1.7, what sample length N would you need in order to show a symmetric discrete spectrum with reference to 220 Hz (i.e., inclluding two maxima of equal height)? Explain your answer.

Through learning the FFT implementation of the DFT, you have taken your first steps toward –hopefully— understanding not only the frequency-domain representation of signals, but also the concept of sampling. When you evaluate some continuous function at uniformly spaced "samples," you are in effect sampling that signal.

2. Downsampling by decimation

You will now explore what it means to adminster medicine to animals. Oh, and also decimation.

2.1 Retrieve from the class website, and load, the speech sound file "speech_female.wav". This signal is sampled at a rate of 44.1 kHz. You are going to make a really pretty picture of its distribution of energy in time and frequency over the first epoch of length 1.4 seconds. During this time, the woman says: "To administer medicine to a." You can listen to this by using `sound`. (Make sure you pass the correct sampling rate, otherwise you will hear a zombie with an English accent.) Run the following program and include the resulting picture in your report. Add your own comments in the program, line-by-line, to describe what it is doing.

```
clear all

sndfile = 'speech_female.wav';

[x,Fs] = audioread(sndfile);

N = 512;
[S,F,T] = spectrogram(x(1:Fs*1.4),N,3*N/4,N*4,Fs);

f = figure('Position',[500 300 700 500],'MenuBar','none', ...
  'Units','Normalized');
set(f,'PaperPosition',[0.25 1.5 8 5]);
axes('FontSize',14);
colormap('jet');
imagesc(T,F./1000,20*log10(abs(S)));
axis xy;
set(gca,'YTick',[0:2000:Fs/2]./1000,'YTickLabel',[0:2000:Fs/2]./1000);

ylabel('Frequency (kHz)');
xlabel('Time (s)');

print(gcf,'-depsc2','p2i1.eps');
```

You might have heard about the "Short-Time Fourier Transform" (STFT). Even you haven't, the name speaks for itself: it computes a Fourier transform (which ideally runs from -∞ to -∞) restricted to some finite time interval, which is more practical and realistic for real-world signals. The figure you just created is the log magnitude of the STFT, and is called a "spectrogram" or "sonogram." Essentially it displays the distribution of energy in a signal as a function of time and frequency. This is a simple but nonrigorous way to show how the spectrum evolves in time. It is nonrigorous because time and frequency are a Fourier pair, so you cannot consider them as independent variables as you do (wrongly) in a spectrogram plot. However, if the spectrum changes slowly, this is a very good approximation. (If you go on to study advanced methods in DSP, you will learn about other methods, such as using the analytic representation of signals.)

2.2 Using the spectrogram you created in 2.1, in what frequency band is most of the energy located? Recalling the Nyquist-Shannon sampling theorem, at what minimum rate would you sample this signal such that it could be reconstructed well? Explain why is this now not an exact reconstruction?

2.3 Looking at the distribution of energy across frequencies in your spectrogram, and knowing that the woman says "To administer medicine to a", which sounds of which words correspond to the energy between 4 and 18 kHz?

Now we are going to *downsample* the waveform using *decimation*. Essentially, this means picking every other sample from a sampled signal (decimation by a factor of 2), or every third sample (decimation by a factor of 3), or every fourth sample (decimation by a factor of 4), or every *n*th sample (decimation by a factor of *n*, where *n* a positive integer). In the last case, the sampling rate becomes *Fs*/*n*, where *Fs* is the original sampling rate. For diagrams, see page 60/66 in the slides for unit 6/7 on signal classification.

2.4 Looking at the figure you produced in 2.1, what is the highest decimation factor you think that this signal can withstand whilst you can still understand the speech? Explain your reasoning. What would be the sampling rate at that factor? How would you expect the highest decimation rate to change for someone whos is unfamiliar with the English language, and why?

2.5 Do it. Include your code. Include all your observations.[1]

[1] Hint: Start with a random factor, say 42, and then have a friend or family member try to understand the spoken text. Decrease the factor until it is barely intelligibe, and then report its number.