CPE 367 – Experiment 1 v7   (60 points)
Sampled Signals and Spectra – Dr F DePiero

Overview and Motivation

This experiment provides opportunities to examine sampled signals in both the time and frequency domains. You will write Python programs to generate a variety of signals, saved as WAV files. (A Python class for WAV file I/O is provided.)

A variety of MatLab functions are provided that load WAV files and provide plotting in both the time and frequency domains. Tools and techniques in this experiment will be used throughout the quarter. Resources and references are included below.

Learning Objectives

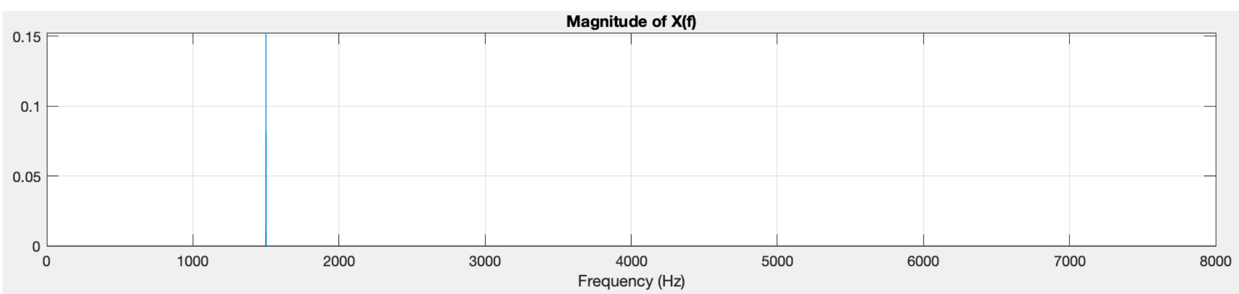
- Interpret spectra of sinusoidal signals, using MatLab
- Generate common waveforms in Python, to meet given specifications
- Compute a digital frequency, given an analog frequency and sample rate $\omega_0 = 2\,\pi\frac{f_0}{f_s}$
- Examine the effect of time-domain operations on signal spectra

Prerequisite Learning Objectives

- Introductory knowledge of MatLab and programming experience in Python
- Identify amplitude, frequency and phase shift of CT sinusoidal signals

Background

The spectrum below shows frequencies ranging from 0 Hz (DC) up to 8000 Hz. This spectrum was of a single cosine function at 1500 Hz and includes a spike at the frequency of oscillation. ***The magnitude of the spectrum is ½ the amplitude of the sinusoid***.



This plot was generated by MatLab and the signal was loaded from a WAV file. Signals in WAV files consist of a list of numerical values, however, MatLab displays the spectrum as if it originated from a continuous-time signal. To generate the discrete samples of a sinusoidal signal, let $t = \frac{n}{f_s}$ where fs is the sample rate. This yields a discrete-time sequence indexed by n.

$$x[n] = A\cos\left(2\,\pi\frac{f_0}{f_s}\;\;n\right) = \;A\cos(w_0\;\;n)$$

And here we see the digital frequency, $\omega_0 = 2\,\pi\frac{f_0}{f_s}$ which has units of radians/sample. A phase shift can be included as needed.

Procedures, Questions and Deliverables

1) Interpreting Spectra with Signals Generated in Python

Write a Python program to generate a 2 second WAV file, to generate a cos() at f = 2000 Hz. The sample rate should be 16kHz. The WAV file should have one channel (mono, not stereo). Set the amplitude to be approximately 75% of the maximum, for a 16 bit, signed number. Use a 45-degree phase shift.

See the example in cpe367_signal_gen.py which imports the cpe367_wav.py class.
Run your modified Python program
% python3 cpe367_signal_gen.py
After generating the wav file in Python, move it to the MatLab folder for subsequent analysis

- **1.1) How many samples in total? _________ (2)**
- **1.2) How many bytes are expected for the WAV file? (FYI the WAV header adds 44 bytes) WAV files are simply a sequence of the samples and do not include any compression. This signal is mono, not stereo. What is the expected file size in bytes? ________ (2)**
- **1.3) What is your chosen amplitude? __________ (2)**
- **1.4) What is the digital frequency in rads/sample? _________ (e.g. pi/20, or similar) (2)**

To verify your result, run the following script from the MatLab command line

>> spectrum_signal_dft your_signal.wav

This will display the magnitude and phase of the spectrum, as well as a time-domain plot of the sinusoid. Make sure to place your signal file in the MatLab folder. Also add any needed files, folders and subfolders to MatLab's path.

- **1.5) Include a copy of your MatLab plot (4)**

Note that when MatLab reads a 16-bit WAV file, it automatically converts the data to floats with a +/-1 maximum intensity. Also note that you can hover the mouse over the MatLab plot window and it will display associated plot values. You can also zoom into the MatLab plot. Given the time-domain signal and based on theoretical expectations, what should the following values be?

- **1.6) <u>Expected</u> magnitude at f = 2000 Hz? _________ Any error seen in plot? %?_____ (2)**
- **1.7) <u>Expected</u> phase at f = 2000 Hz? (degrees) _________ Any error seen in plot? %?___ (2)**
- **1.8) Include a copy of your modified Python routine 'gen_wav', with comments (4)**

2) Impact of Delay on Frequency Spectra

Delay is a very common effect, so it is useful to understand the impact of it on spectra. Set the delay, D, to be D = 0.25 mSec, and re-examine the spectrum. What is the equivalent delay in samples, d?

- **2.1) What is the equivalent delay, d, (as a # of samples)? _________ (2)**

To implement the delay in your Python program, you can simply modify the sample index

$$m = n - d$$

and use the new variable m in the argument to cos()
Now, let's relate the delay, d, to an equivalent phase shift, θ. Call the original signal $x_1[n]$

$$x_1[n] \; = \; x_1\left(t = \frac{n}{f_s}\right) \; = \; A\cos\left(\frac{2\pi f_0}{f_s}\, n\right)$$

The delayed signal is $x_2[n]$

$$x_2[n] \; = \; x_1[n-d] \; = \; A\cos\left(\frac{2\pi f_0}{f_s}\,(n-d)\right)$$

What is the associated phase shift, θ?

$$x_2[n] \; = \; A\cos\left(\frac{2\pi f_0}{f_s}\, n \, - \, \theta\right)$$

- **2.2) What is the equivalent phase shift, theta, (in radians)? _________ (2)**

Run your program twice, once with the delay and once without. (Drop the phase shift of part 1). Compare the spectrum in these two cases.

- **2.3) Describe similarities and differences between the delayed and original version (2)**

Lastly, drop the delay (d) from your program and replace the amplitude A with -A. Regenerate the signal and compare

- **2.4) Describe similarities and differences between the delayed version and the one with -A. Explain why these similarities or differences arise. (2)**

3) Impact of Amplitude Modulation on Frequency Spectra

Amplitude modulation refers to the process of taking the product of a given signal with a sinusoid. Here we will multiply two cos() functions together and examine the resulting changes to the frequency spectrum.

Write a Python program to generate a 2 second signal, x(t), and save it in a mono 16-bit WAV file, with a sample rate of 16 kHz. Generate the signal x(t) by taking the product of two cosines.

$$x(t) = \; A\cos(2\,\pi\, f_1 t + \; \theta_1) \quad B\cos(2\,\pi\, f_2 t + \; \theta_2)$$

Let f_1 = 2000 Hz and f_2 = 500 Hz. Assign A and B such that x(t) signal has an amplitude of approximately 100% of the maximum. Assign the phase of your sinusoids such that the higher frequency component of x(t) has a phase of +45 degrees and the lower component has a phase of -45. These higher and lower components are associated with the frequency range 0-8000 Hz.

See the example in cpe367_lab.py. Also note that math.pi and math.cos are available for you.

- **3.1) What mathematical identity allows you to predict the spectrum of x(t)? (For example, in the DSP 1st textbook on page 14) (2)**
- **3.2) What are your values for A and B? ____________ (more than one answer possible) (2)**
- **3.3) What are your values for θ_1 and θ_2? ____________ (2)**
- **3.4) If the order of the cosine functions is swapped then would x(t) change? ___(Yes/No) (2)**
- **3.5) Include a copy of the MatLab plots (4)**
- **3.6) Include a copy of your modified Python routine 'gen_wav', with comments (4)**

4) Impact of Absolute Value on Frequency Spectra

Write a Python program to generate a signal, x(t), that is 1/16 of a second long and save it in a mono 16-bit WAV file, with a sample rate of 16 kHz. Generate the signal x(t) by taking the absolute value of a cosine.

$$x(t) = |\ A\cos(2\,\pi\,f_1 t)\ \ |$$

Let f_1 = 160 Hz. Assign A such that x(t) signal has an amplitude at approximately 100% of the maximum possible for a 16-bit WAV file.

- **4.1) Describe the frequency components that are present in this signal. Is there a pattern? What are the lowest two frequencies present? Is DC present? (8)**
- **4.2) Include a copy of the MatLab plots (4)**
- **4.3) Include a copy of your modified Python routine 'gen_wav', with comments (4)**

Questions that will be answered in future lectures and labs

- ***No response to these questions is required for your report. These are just intended to be thought provoking.***
- How are frequency spectra computed?
- Why does a sinusoid have a spike in the frequency domain?
- Why is the magnitude in the frequency domain half of the sinusoidal amplitude?
- Why did the absolute value operation create such a change in the signal?