Exercise 2 — EE6733 – Reading Course

Target Deadline: Wednesday May 22nd, 2019

FREQUENCY DOMAIN REPRESENTATION OF SIGNALS

ALL SCRIPTS SHOULD APPLY THE CODING STANDARDS WE DISCUSS IN CLASS.
READABILITY, EFFICIENCY, MODULARLIZATION AND GENERALIZATION ARE IMPORTANT CONSIDERATIONS BEYOND FUNCTIONALITY

All periodic signals can be represented as a summation of sinusoids according the Fourier Series:

$$g_{T_0}(t) = a_0 + \sum_{n=1}^{\infty}[a_n \cdot cos(2\pi f_0 t) + b_n \cdot sin(2\pi f_0 t)] \quad (1)$$

In (1) the fundamental frequency $f_0 = {}^{1}/_{T_0}$ where T_o represents the period of the signal. Also,

$$a_0 = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} g_{T_0}(t)dt, \qquad a_n = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} g_{T_0}(t) \cdot cos(2\pi f_o nt)dt, \qquad b_n = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} g_{T_0}(t) \cdot sin(2\pi f_0 nt)dt$$

are called the coefficients of the series. Euler provides formulas for translating $sin(x)$and $cos(x)$into exponential form, so that (1) can be represented in its complex exponential form:

$$g_{T_0}(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{-j(2\pi n f_0)t} \text{ , where } \quad c_n = \int_{-T_0}^{T_o} g_{T_0}(t) \cdot e^{-j(2\pi n f_0)t}dt \quad (2)$$

For signals that are not periodic, we can represent them in terms of a periodic signal whose period approaches infinity, $T_0 \to \infty$.

$$g(t) = \lim_{T_0 \to \infty} g_{T_0}(t) \quad (3)$$

When we apply this limit to (2) we get the inverse Fourier transform:

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot e^{j2\pi f t}df \quad (4)$$

where $G(f) = \lim_{T_0 \to \infty} c_n$ and is known as the Fourier Transform:

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-j2\pi f t}dt \quad (5)$$

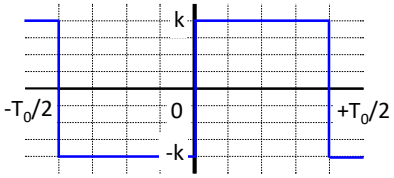
While the math is somewhat complicated to derive the Fourier transform and its inverse, the interpretation of the equations is rather straightforward as follows:

The coefficients in the Fourier series (represented in either (1) or (2)) provide the amplitudes of the sinusoidal components (each with a distinct frequency) that must be summed to create the *periodic* function of interest. A plot of these amplitudes against their frequencies is called a line spectrum and represents the 'frequency content' of a periodic signal.

The Fourier Transform (represented in (5)) provides a way to determine the 'frequency content' of a signal that is *not periodic*. Plotting the amplitude of this complex signal produces a continuous spectrum which is called an amplitude spectrum. It provides the amplitudes of the sinusoidal components which make up the signal, just as the Fourier series does, but in the case of *non-periodic* signals, the components are continuous across the spectrum of frequencies. The phase of the complex signal can also be plotted against frequency, and collectively the amplitude spectrum, and phase spectrum are known as the frequency spectrum of the signal. The Inverse Fourier Transform (represented in (4)) simply provides a way to determine the time course of this signal, given its 'frequency content'.

Useful functions for this assignment: `stem()`, `fft()`, `abs()`, `sin()`, `plot()`, `subplot()`, `xlabel()`, `ylabel()`, `title()`, `text()`

Question 1: Complete this question by following the steps below. If you do it properly, you will see that a square wave is really just the sum of a bunch of sinusoids whose frequencies and amplitudes can be determined with the Fourier series.



$$g(t) = \begin{cases} -k & if \quad -\frac{T_0}{2} \le t < 0 \\ +k & if \quad 0 \le t < \frac{T_0}{2} \end{cases}$$

In matlab, generate a square wave defined as above (assuming periodicity: $g(t + T_0) = g(t)$). To do this, calculate by-hand the amplitudes and frequencies of the sinusoids which must be summed according to equation 1 (and its corresponding coefficients). Use $T_0 = 2\pi$ and k = 1 mV. Other parameters can be defined as follows: fs = 20Hz, T=10 sec.

- If you complete the calculations correctly, you should find that $a_0 = 0$. What does this tell you about the mean value (in circuit terms we call this the DC component) of the signal?
- If you complete the calculations correctly, you should find that $a_n = 0$. What does this tell you about the kinds of sinusoids you have to add together to get this square wave? Does that make sense based on the shape depicted in the figure?
- If you complete the calculations correctly, you should find that $b_n = \frac{(4k)}{(n\pi)}$ for $n = 1,3,5$ .... You may need to use the following identities to get this result: $\cos(-x) = \cos(x)$ and $\cos(0) = 1$. Use the expression for b_n to generate the square wave by adding the components together.

Have your matlab script generate a figure (4 rows, 1 column) which plots the first 3 sinusoid components, followed by the square wave produced by summing these 3 components. Indicate on the figure the frequencies in radians and Hz of each component (recall that $\omega = 2\pi f$ to convert from radians/sec to cycles/sec = Hz).

Copy your matlab script so that you can modify it to change the number of sinusoids you add together from N = 2, to 5, 10 and 100. Plot the resulting square waves in 1 figure (4 rows and 1 column). Note the improvement as N gets larger. Are there any negative influences as N get's larger?

Question 2: Use matlab to generate a line spectrum for the square wave you analysed in question 1 (Amplitude in mV, Frequency in Hz). Include the first 3 components in your line spectrum by extracting the values from the code you use to generate the square wave). Use `stem(f,a. 'r:')` to generate the line plot (f represents frequencies of components, a, represents amplitudes).

Now, use the Fourier Transform to determine the frequency content of the square wave output with N=3. Use the code from Question 1 to generate the wave, and use the code that follows to calculate its amplitude spectrum (we will learn more about the code in Exercise 3):

```
df = 1/To;
f = 0:df:fs/2-df;
Aspectrum = abs(fft(sq_wave));  %sq_wave:  square wave created with N=3
Aspectrum = (2/Nsamples)*Aspectrum(1:Nsamples/2); %Nsamples: Number of samples in sq_wave
…
plot(f, Aspectrum)
```

Superimpose the results from the Fourier transform onto the line spectrum you created based on your manual calculations from question1. Do they correspond?