25.108 Introduction to Engineering

Experiment I: Expressing Periodic Waveforms as the sum of sine's and cosines.

Objectives: To learn about trigonometric expansions in Matlab; To introduce the concepts of time and frequency spectrum using Matlab. To illustrate that any periodic waveform can be expressed as a sum of sine's and cosines (Fourier Series) and to show the relationship between the sharpness of a waveform (squareness) and the number of terms required. This laboratory will help you to do some basic MATLAB functions such as plotting, and manipulating trigonometric functions.

Before you come into lab: Learn the basic theory behind the experiment in Lecture.

Basic Theory: In the early 1800's Fourier showed that any periodic waveform could be expressed as an infinite sum of sine's and cosines. He developed the following relationship.

$$\boxed{x(t) = \frac{a_o}{2} + \sum_{n=1}^{\infty}\left[a_n \cos n\omega t + b_n \sin n\omega t\right]}$$

$$\text{where } a_o = \frac{2}{T}\int_a^{a+T} x(t)\,dt$$

$$a_n = \frac{2}{T}\int_a^{a+T} x(t)\cos n\omega t\,dt$$

$$b_n = \frac{2}{T}\int_a^{a+T} x(t)\sin n\omega t\,dt$$

③ Polar Fourier Series

- Another form of FS is obtained by combining the sine and cosine terms to give a single component with a phase angle

$$\boxed{x(t) = d_0 + \sum_{n=1}^{\infty}\left[d_n \cos(n\omega t + \theta_n)\right]}$$

$$\text{where } \quad d_0 = a_0 = C_0$$

$$d_n = \sqrt{a_n^2 + b_n^2} = 2|C_n|, \quad \theta_n = -\tan^{-1}\left(\frac{b_n}{a_n}\right)$$

Analog Signals 17

In this laboratory we are going to expand a square wave and triangular wave into it's constituent components and show the effect of filtering on wave shapes.

Procedure:

Download the SpectrumAnalyzer function SpectrumAnalyzer.m from the website. Instructions for using SpectrumAnalyzer

SpectrumAnalyzer(ArrayYouWishToAnalyze,SamplingFrequency)

Output is a plot in dB showing the output FFT.

A. Create a series of sinusoids.

   1. Create a 10 second time axis, sampling every 0.0001 seconds (Fs=10000 Hz).
   2. Create and plot about 10 cycles of y= cos ($2*\pi*f_0*t$) where t is your time axis and f_0 is 200 Hz.
      a. Now try to visualize it in the spectral domain by using SpectrumAnalyzer(y,10000). Note that there are 2 lines at 200 Hz and –200 Hz
      b. Listen to the sound by using the soundsc(y,10000) function

   3. Create another sinusoid at the second harmonic of the sinusoid (400Hz). Plot it on the same graph as the first, and play it using your sound card.
      a. Visualize it using the SpectrumAnalyzer Function. Now you will see a pair of lines at 400 and -400 Hz.
      b. Plot on the same graph as before

B. Create a square wave from the sinusoids, after each step plot a small section, and run the spectrum Analyzer function.

   1. Add first harmonic term you crated $+(\sin(k*\pi/2)/k)*\cos(k*2*\pi*f_0*t)$ for k=3
   2. now do it for k=1,3,5 (using the scaling coefficient on each term)
   3. now do it for k=1,3,5,7 (etc)
   4. now do it for k=1,3,5,7,9,11,13
   5. Use a for loop with a pause to do 21 harmonics (only odd).

Comment on the squareness of the output as a function of the number of harmonics that you use. Play it on your sound card using the soundsc(y,Fs) and listen to the difference.

C. Remove all the Harmonics using a low pass filter.
   1. Design a filter to remove all harmonics from the square wave with the exception of the first. Let Fs=10000 Hz, Pass band = 220 Hz, stop band =300 Hz
   2. Export the coefficients to workspace (Num)
   3. Run y=filter(Num,1,SquareWave) to remove harmonics. Plot and listen on your sound card. Run through the spectrum analyzer and compare to part A.

D. Look at a song in the time and frequency domain. Type “Load Handel”. This will create an array (y,Fs) where y has the music and Fs is the sampling frequency 8192 Hz.

   a. play the song using your sound card
   b. look at the song in the frequency domain using spectrumanalyzer
   c. look at the first 10000 points in the time domain.
   d. Speed the music, play using sound card at Fs=12000
   e. Design a low pass filter to pass frequencies from 0 to 2000 Hz. Follow the procedure from before and filter your waveform, plot the spectrum and listen to the output. It does not sound so good anymore.

**** In your future classes and career keep the following in mind: Very square waveforms, like idealized bits (1’s and 0’s), require large amounts of bandwidth. As digital speeds become higher and higher, circuit boards look more and more like a low pass filter. Low pass filtered square pulses become less and less square and errors can occur when the circuitry cannot determine whether the waveform is a 0 or a 1.

Laboratory Report: Write a Laboratory report describing what you did. Hand in all code, plus plots created in each step into a lab report. Make comments and observations.