

EGR 111 Fourier Series

This lab introduces Fourier Series which allow periodic signals to be written as the sum of sinusoids.

New MATLAB Commands: none

Resources (available on course website): Trumpet.wav

1. Fourier Series

Joseph Fourier discovered that periodic signals can be written as the sum of harmonically-related sinusoids. The Fourier Series allows us to see which frequencies are present in a signal and to predict how the signal will behave as it passes through filters or other systems.

Fourier Series are used in many engineering situations: seismic engineering, vibrating mechanical systems, and with audio, radio, and micro-wave signals. They are also used to compress data (as is done with JPG photos) by capturing the most important data (think about the filtering examples of an earlier lab).

In order to demonstrate the Fourier Series, we will consider the following sum where the fundamental frequency $f_0 = 1$ Hz:

$$\begin{aligned}x(t) &= \sum_{\substack{n=1 \\ \text{odd}}}^{\infty} \frac{1}{n^2} \cos(2\pi n f_0 t) \\ &= \frac{1}{1^2} \cos(2\pi 1t) + \frac{1}{3^2} \cos(2\pi 3t) + \frac{1}{5^2} \cos(2\pi 5t) + \dots \\ &= \cos(2\pi 1t) + \frac{1}{9} \cos(2\pi 3t) + \frac{1}{25} \cos(2\pi 5t) + \dots\end{aligned}$$

You can think of each term as the product of a cosine (a “tone”) and an amplitude (a loudness). For example, the second term has frequency 3 Hz and amplitude 1/9. Note that only the odd values of n (1, 3, 5, 7, ...) are included in the summation. The first term in the summation, where $n = 1$ is called the fundamental frequency. Also notice that the frequency of each term is an integer multiple of the fundamental frequency $f_0 = 1$ (that is $1 \times f_0$, $3 \times f_0$, $5 \times f_0$, etc.) The term with $n = 3$ is called the third harmonic, the term with $n = 5$ is called the fifth harmonic, and so on. Since the amplitude of each term is $\frac{1}{n^2}$, as n increases the amplitude decreases rapidly, which allows us to approximate the sum using the first few terms (instead of attempting to add an infinite number of terms).

Exercise 1: For the Fourier Series shown above, use MATLAB to generate the first term $x_1(t) = \frac{1}{1^2} \cos(2\pi 1t)$ and the third term $x_3(t) = \frac{1}{3^2} \cos(2\pi 3t)$ for $-1 \leq t \leq 1$ second.

Make a plot with the first and third terms, and the sum of the first and third terms all on the same graph. Note you can put three functions on the same graph by defining the time vector t , calculating the functions x_1 and x_3 , and using the following command:

`plot(t, x1, t, x3, t, x1+x3)`. Add a legend to the graph to label the curves.

Exercise 2: Repeat Exercise 1 except this time plot the terms $n = 1, 3,$ and $5,$ and their sum.

Suppose we want to add up many, many terms of a Fourier Series. One approach is as follows:

```
clear
t = linspace(-1,1,1000); % define time vector
fo = 1; % fundamental frequency (in Hz)
x = 0; % initialize the sum

n = 1; % the first value of n
x = x + 1/n^2*cos(2*pi*n*fo*t); % add n = 1 term
n = 3; % the next value of n
x = x + 1/n^2*cos(2*pi*n*fo*t); % add n = 3 term
n = 5; % the next value of n
x = x + 1/n^2*cos(2*pi*n*fo*t); % add n = 5 term
```

Using this approach, we could add any number of terms that we want, but it would be very tedious to add a large number of terms. Of course there is an easier way to repeat a command in MATLAB.

Exercise 3: Write a script file to calculate the sum of the terms $n = 1, 3, 5, \dots 99$ of the Fourier Series shown above, and plot the result for $-1 \leq t \leq 1$ second. What shape does the sum converge to?

Checkpoint 1: Show the instructor your script file and plot from Exercise 3.

2. Fourier Series for Instruments

You may have noticed in a previous lab that the music we generated using sinusoids did not sound much like a real instrument. In addition to the fundamental frequency, most instruments generate many harmonics. Whereas the fundamental frequency determines the pitch of a note (see http://en.wikipedia.org/wiki/Piano_key_frequencies), the amplitude of the various harmonics determine the quality of the sound (called timbre) that allows us to identify difference instruments even when they play the same note.

To demonstrate the effect of the harmonics, let's write a program to generate the fundamental and harmonics for a trumpet and compare it to a recording of a real trumpet from the file Trumpet.wav, which should be in your P:\MATLAB folder. The amplitudes of the harmonics of the original trumpet recording are given below.

Harmonic (n)	Amplitude
1	0.014447
2	0.192465
3	0.690787
4	1.000000
5	0.603843
6	0.235496
7	0.128556
8	0.067857
9	0.027369
10	0.011039

The fundamental frequency of the original recording is $f_0 = 293.4$ Hz and the sampling rate is $f_s = 44100$ Hz. The fundamental frequency f_0 corresponds to the first harmonic ($n=1$) in the table above. You can generate this fundamental term as follows:

```
A = [0.014447 0.192465 0.690787 1.000000 0.603843 ...
0.235496 0.128556 0.067857 0.027369 0.011039];
fo = 293.4;      % fundamental frequency (Hz)
fs = 44100;     % sampling rate (Hz)
T = 1/fs;      % sampling interval (Sec)
t = 0:T:2;     % generate sample times for 2 seconds
x1 = A(1)*cos(2*pi*1*fo*t); % fundamental term
soundsc(x1,fs) % send to speakers
```

Compare the sound of the fundamental term to the original recording (Trumpet.wav). Do they sound similar?

Exercise 4: Write a program that uses a loop to add each of the harmonics with the amplitudes shown above and to send the result to the speaker after each harmonic is added. Put the `pause` command inside the loop after the `soundsc` command, so that the program will stop until the user presses a key. Also print the value of the harmonic number to the screen so you can tell how many harmonics have been added. Does the sum of the harmonics sound closer to the original as more harmonics are added?

Checkpoint 2: Show the instructor your script file for Exercise 4.