EE224 Fourier Series Lab

In this lab you will approximate the Fourier series integral in Matlab to compute the Fourier series decomposition and analyze signal frequency components.

1 Prelab

1.1 Fourier Analysis

Begin by reviewing the Fourier analysis formula described in the book and in class. Given a periodic signal $x(t)$ with period T, $x(t)$ can be represented by

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} \qquad (1)$$

where $\omega_0 = \frac{2\pi}{T}$ and

$$a_k = \frac{1}{T}\int_0^T x(t) e^{-jk\omega_0 t}\, dt \qquad \text{for all k} \qquad (2)$$

Although an infinite number of harmonics may be required for a general signal, in most situations, a finite number of them provide an acceptable approximation.

If the signal $x(t)$ is not given as a mathematical function but we only have its sampled recorded version of it, the integrals to compute the coefficients, a_k's cannot be evaluated precisely. Although there are more efficient methods to perform the Fourier analysis directly on the sampled signals, which we will study later in the course, we will use a simple method to approximately evaluate those integrals: the Riemann approximation.

Assuming that the signal $x(t)$ is reasonably nice (Riemann integrable) and that the sampling time T_s is an integer fraction of the period T, i.e., we have an integer number N of samples in a period (this is not really necessary but simplifies our code)

$$a_k = \frac{1}{T}\int_0^T x(t) e^{-jk\omega_0 t}\, dt \simeq \frac{T_s}{T}\sum_{n=0}^{T/T_s} x(T_s n) e^{jk\omega_0 T_s n}\ . \qquad (3)$$

The approximation gets better and better as the number of samples goes to infinity.

Note however that, fixing the sampling time limits the quality of the approximation especially for larger k's. This is because T_s becomes too large with respect to ω_0 the frequency of the kth harmonic.

1.2 Fourier Synthesis

The synthesis formula generates periodic signals from the linear combination of harmonic complex exponentials. Here we assume that the number of non-zero coefficient a_k's is finite. Then

$$\tilde{x}(t) = \sum_{k=-N}^{N} a_k e^{jk\omega_0 t} \qquad (4)$$

In particular,

$$\tilde{x}(T_s n) = \sum_{k=-N}^{N} a_k e^{jk\omega_0 T_s n} \qquad (5)$$

2 Laboratory Assignment

2.1 Fourier Analysis and Synthesis

Exercise 1. Write a Matlab function "fanal" which takes as inputs x_T, a vector of samples of $x(t)$ worth one period of $x(t)$, k the index of the desired kth coefficient, the period T, and the sampling time T_s and produces as output the approximate a_k coefficient according to (3). Note that you can take easily advantage of vectorization in your code by considering the product of the row vector x_T and the column vector $e^{-jk\omega_0 T_s n}$.

Exercise 2. Construct a test signal. Define in Matlab

x_T=[ones(1000,1);zeros(1000,1)];

This is one period of a square wave signal. Let the fundamental period be T = 0.01 sec.

a) Plot the signal and explain why the sampling time is $T_s = T/2000$.
b) Use the function you wrote in the previous exercise to compute a_0, a_1, a_2, a_3.
c) Without computing them, write down a_{-1}, a_{-2}, a_{-3}. What is the relationship between a_k and a_{-k}?
d) Verify the coefficients you have computed approximate reasonably well the actual coefficients obtained by evaluation of the integrals as done in the textbook.

Exercise 3. Construct a test signal. Define in Matlab

x_T=[(1:1000)/1000;zeros(1000,1)];

This is one period of a ramp wave signal. Let the fundamental period be T = 0.1 sec.

a) Plot the signal and explain why the sampling time is $T_s = T/2000$.
b) Use the function you wrote in the previous exercise to compute a_0, a_1, a_2, a_3.
c) Without computing them, write down a_{-1}, a_{-2}, a_{-3}.
d) Verify the coefficients you have computed approximate reasonably well the actual coefficients obtained by evaluation the integrals of equation 2 in closed form.

Exercise 4. Write a Matlab function "fsynt" which has the following inputs:

- A column vector $C = [a_0, a_1, \ldots, a_N]$ of coefficients where N is the largest integer corresponding to a non-zero coefficient,
- The sampling time, T_s,
- The fundamental period, T.

The function output should be: $\tilde{x}_T(T_s n)$, the vector of T/T_s (integer) samples corresponding to one period of $\tilde{x}$. Your program should check that T/T_s is an integer. Your program should use the symmetry property of Fourier coefficients to extend C to include the complex conjugate coefficients corresponding to the negative k's.
To speed up your code, you can parallelize the computations by generating an array F whose columns are the vectors $e^{jk\omega_0 T_s n}$. $\tilde{x}_T$ is then given as F*CC, where CC is a vector containing all the coefficients from $-N$ to N.

2.2 Fourier Analysis and Synthesis with Real Signals

Load the file lab4.mat from the course webpage associate with this lab. It contains “trumpet” and “whistle”. Both signals are sampled at $f_0 = 44{,}100$ Hz and they represent one period of synthetic trumpet and whistle tones generated by a toy electric keyboard.

a) Compute the period of each of the two signals.

b) For each signal, compute and report the first 9 harmonic coefficients, i.e., a_k’s with $k = -9, \ldots, 9$ using the function “fanal” you have developed.

c) For each signal, plot the spectrum of these frequency components, and briefly comment on their main differences. You may use the Matlab function stem to plot the spectrum.

d) For each signal, synthesize an approximation by using the periods in part a), the coefficients in part b), the sampling time $T_s = 1/f_0$, and your function “fsynt”. For each signal, plot on the same plot the given signal and its synthesized approximation. Comment on the quality of the approximation in both cases.

e) For each signal and its approximation generate a new signal by repeating them for 1000 periods. Use the function “soundsc” in Matlab and the sampling frequency f_0 to hear the sound you have generated. Can you hear the difference between the originals and the approximations? Comment.

Note that the trumpet sound has a much richer spectrum and corresponds to a more complex sound. While the whistle sound essentially does not have any harmonic component above the ninth and neither has any even harmonic components, the trumpet sound has important harmonics way after the ninth; this can be argued from the error in the approximation.