EE224 Fourier Series Lab

In this lab you will experience the power of the Fourier series analysis and synthesis.

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

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

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

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

Although a infinite number of harmonics may be required for a general signal, in most situations, a finite number of them provide a practically good 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_ks cannot be evaluated precisely. Although there are more efficient methods to perform the Fourier analysis directly on the sampled signals, the Fast Fourier Transform (FFT), we will use a simple method to approximately evaluate those integrals: the Riemann approximation. Assuming that the signal $x(t)$ reasonably smooth (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)

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

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 $k\omega_0$ the frequency of the kth harmonic.

1.2 Fourier Synthesis

The synthesis formula allows to generate periodic signals from the linear combination of harmonic complex exponentials. Here we assume that the number of non-zero coefficient a_ks is finite. Then

$$\tilde{x}(t) = \sum_k a_k e^{jk\omega_0 t}$$

In particular,

$$\tilde{x}[T_s n] = \sum_k a_k e^{jk\omega_0 T_s n}$$

2 Laboratory Assignment

2.1 Exercise 1.

Write a Matlab function "fanal" which takes as inputs x_T, a vector of samples of $x(t)$ covering 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 considering the product of the row vector x_T and the column vector $e^{-jk\omega_0 T_s n}$. A template for the function is provided.

2.2 Exercise 2.

Let's 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) 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 of the integrals in closed form as is done in the text.

2.3 Exercise 3.

Write a Matlab function "fsynt" which has the following inputs: A column vector $C = [a_0, a_1, \ldots, a_m]$ of coefficients where m is the largest integer corresponding to a non-zero coefficient, the sampling time T_s, the fundamental period T, and produces $\tilde{x}_T(T_s n)$ the vector of T/T_s (integer) samples corresponding to one period of $\tilde{x}$. Note that your program should verify that T/T_s is an integer. You will also need to extend C to include the complex conjugate coefficients corresponding to the negative $k's$. Finally, you may want to generate an array `F` whose columns are the vectors $e^{jk\omega_0 T_s n}$. $\tilde{x}_T$ is then computed as `F*CC`. Where `CC` is a vector containing all the coefficients from $-m$ to m. A template for this function is provided.

2.4 Exercise 4.

Load the data file lab4.mat from the course webpage. It contains "trumpet" and "whistle". Both signals are sampled at $f_0 = 44100$ Hz and they represent one period of synthetic trumpet and whistle tones generated by a toy electric keyboard.

a) Compute the period of the two signals.

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

c) For each signal, plot the magnitude and phase spectra of the frequency components from part b. 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 coefficient in part b), the sampling time $T_s = 1/f_0$, and your function "fsynt".

e) 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.

f) For each signal and its approximation, generate a new signal by repeating them for 1000 periods. Use the function "sound" 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 you may need to scale the signals to have magnitude 1, see help sound)

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 components above the ninth and does not have any even harmonic components, the trumpet sound has important harmonics above the ninth; this can be argued from the error in the approximation.