STANFORD UNIVERSITY
DEPARTMENT of ELECTRICAL ENGINEERING

EE 102B   Spring 2013

Lab #1: Numerical Evaluation of Fourier Series

Date Assigned: April 3, 2013   Date Due: April 10, 2013

Lab Report: Please turn in an annotated MATLAB script of your work in Section 3 with explanations as requested as this week's informal lab report.

---

1 Introduction & Objective

The goal of this laboratory project is to use MATLAB to study the representation of periodic signals and filtering of such signals using an LTI system. We will base our work on the analytical evaluation of the Fourier series coefficients, which you will have to do by the methods of calculus and pencil and paper. MATLAB will enter the picture when we evaluate the synthesis formula. Another approach would be to use a symbolic algebra package Maple, which is built into MATLAB, to derive formulas for the Fourier Series coefficients. Still another approach would be to use MATLAB's numerical integration functions to numerically evaluate the analysis expression. It would be useful to explore these latter approaches, it time would permit.

2 Warmup

MATLAB has many features, and you will only learn them all by using them. As you know, MATLAB is based on numerical linear algebra functions that are highly optimized. To write efficient programs you must use matrix/vector operations as much as possible. The following list is comprised of simple MATLAB operations or notation that you might not remember. Try to predict what they do before simply typing them at the MATLAB prompt.

```
2i
2j
2*j
x=(0:10)
x(:)
x.*x
x*x'
x'*x
tt=(-2:.1:2)
tt.*( tt>=0 & tt<=1) %explain what's going on here
```

Remember these things. They will be useful in making good MATLAB programs.

2.1 Fourier Series Analysis and Synthesis

The Fourier Series representation applies to periodic signals. Recall the *analysis* integral and *synthesis* summation for the Fourier Series expansion of a periodic signal $x(t) = x(t + T_0)$. The Fourier synthesis equation for a periodic signal $x(t) = x(t + T_0)$ is

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

where $\omega_0 = 2\pi/T_0$ is the *fundamental* frequency. To determine the Fourier series coefficients from a signal, we must evaluate the *analysis* integral for every integer value of k:

$$a_k = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-jk\omega_0 t} dt \tag{2}$$

where $T_0 = 2\pi/\omega_0$ is the *fundamental* period. If necessary, we can evaluate the analysis integral over any period; in (2) the choice $[0, T_0]$ was a convenient one, but integrating over the interval $[-\frac{1}{2}T_0, \frac{1}{2}T_0]$ would also give exactly the same answer.

3 Exercises:

3.1 Determine Fourier Series Coefficients for Two Signals

The first thing to do is determine an analytic formula for the Fourier coefficients for two different periodic waveforms.

3.1.1 Periodic Pulse Wave

Show that the Fourier coefficients for the periodic pulse wave in Problem 1.4 are as given there.

3.1.2 Full-Wave Rectified Sine

Consider the following periodic signal which is defined over one period to be:

$$x(t) = |120\sqrt{2}\sin(120\pi t)| \quad \text{for} \quad -\frac{1}{240} \le t \le \frac{1}{240} \tag{3}$$

The period is $T_0 = 1/120$ seconds. This signal is called a *full-wave rectified* sine, because $x(t) = |120\sqrt{2}\sin(120\pi t)|$ would be the output of a full-wave rectifier with an input of $120\sqrt{2}\sin(120\pi t)$. These types of signals and their Fourier Series are often used in DC power supply design. Show that the Fourier series coefficients for this waveform are

$$a_k = \frac{(240\sqrt{2}/\pi)}{1 - 4k^2} \tag{4}$$

(To derive this result, express $\sin(120\pi t)$ in terms of couplex eponentials and the integral will be relatively easy to evaluate.) In your summary of this lab work, include the derivations of the Fourier series coefficients for these two periodic signals.

3.2 Write a Function for Fourier Synthesis

In this project we are going to determine Fourier series representations for periodic waveforms, synthesize the signals, and then plot them. In general, the limits on the sum in (1) are infinite, but for our computational purposes, we must restrict the limits to be a finite number N, which then gives the $2N + 1$ term approximation:

$$x_N(t) = \sum_{k=-N}^{N} a_k e^{j(2\pi/T_0)kt}. \tag{5}$$

(Sometimes it is convenient to define the *fundamental frequency* in rad/s as $\omega_0 = 2\pi/T_0$.) One of the things we want to study is what happens as N becomes very large.

The Fourier synthesis can be done with a "sum of complex exponentials" function based on (5). That M-file was called `cos_syth`. The required calling sequence of this synthesis function, called `syn_fourier`, is given below. This MATLAB function implements the computation given in (5). When we use `syn_fourier` for Fourier synthesis, the vector of frequencies will consist of frequencies that are all integer multiples of the fundamental frequency. In addition, we must include both the positive and negative frequency components. Therefore, the input vector of complex amplitudes `ak` will be a vector of length $L = 2N + 1$ containing the $\{a_k\}$ Fourier coefficients in the order $\{a_{-N}, a_{-(N-1)}, \ldots, a_{-1}, a_0, a_1, \ldots, a_N\}$ and the vector `fk` should contain the harmonic frequencies $\{-Nf_0, -(N-1)f_0, \ldots, -f_0, 0, f_0, \ldots, Nf_0\}$ where $f_0 = 1/T_0$ is the fundamental frequency of the periodic signal.

You must write an M-file (called `syn_fourier`) that will synthesize a waveform from complex amplitude and frequency information. The first input must be a vector of times at which (5) is to be evaluated. The second The first few statements of the M-file are the comment lines given below that explain the arguments:

```
function        xx = syn_fourier(tt, ak, fk)
%SYN_FOURIER  Function to synthesize a sum of complex
%             exponentials over the time range given by tt
%  usage:
%    xx = syn_fourier(tt, ak, fk)
%      tt = vector of evaluation times, for the time axis
%      ak = vector of complex Fourier coefficients
%      fk = vector of frequencies
%              (usually contains both negative and positive freqs)
%      xx = vector of synthesized waveform values
%
%    Note: fk and ak must be the same length.
%             ak(1) corresponds to frequency fk(1),
%             ak(2) corresponds to frequency fk(2), etc.
%
% Note: the output might have a tiny imaginary part even if it
%       is supposed to be purely real.  If so, take the real part.
%
```

Thus, after the time axis, coefficient, and frequency vectors are computed, the synthesized waveform can be plotted using

```
xx=syn_fourier( tt, ak, fk );
plot(tt,xx)
```

The form of (5) suggests that $x(t)$ can be computed using two for loops; one over the evaluation times and an inner for loop over the coefficient index. This will work, but it is not the way to do things in MATLAB. You are hereby challenged to find a one-line statement that will evaluate (5) for the given input data. This solution will use only matrix multiplies and will take advantage of the fact that in MATLAB `B=exp(A)` is a matrix where the entries are $b_{ij} = \exp(a_{ij})$.

Hand in a printout of your M-file.

3.3 Synthesis of the Periodic Pulse Signal of Problem 1.4

Start a script file for this exercise using the MATLAB editor. You will add to this file as you progress through the following In this part, use the signal definition given in Problem 1.4 of Problem Set #01.

(a) Using `subplot(211)`, make a plot of the ideal waveform $x(t)$ over the range $-4 \leq t \leq 12$. Label the time scale in terms of seconds.

(b) Using the formula that you obtained in Section 3.1, compute the Fourier series coefficients for `-N:1:N` where `N=10`. Now with `subplot(212)` and `stem( )`, make a plot of the spectrum of the signal. Label the frequency axis as frequency assuming that the period is $T_0 = 8$.

(c) Now assume `N=10` and compute a vector of the corresponding frequencies `fk` for the coefficients `ak`. Use your M-file `xx=syn_fourier( tt, ak, fk )` to compute an approximation to the periodic pulse signal, and using the `hold` command, add this curve in a different color or line type to the plot of the ideal waveform. Use a time sampling that is fine enough to give a smooth plot and clearly show the Gibbs phenomenon.[1] Be sure to add the appropriate instructions to your script.

(d) Using a finite number of terms is exactly equivalent to passing the ideal discontinuous pulse signal through an ideal lowpass filter. What is the cutoff frequency of the ideal lowpass filter if we use $N = 10$ terms? Is your answer unique? Put your answer to this as a comment in your script.

(e) Investigate what happens if `N` is both less than and greater than 10. Add a two more examples to the first subplot. Use the `legend` command to indicate the number of terms for each approximation. What do you observe as `N` increases? Again add this as a comment.

(f) Now consider the signal $x(t)$ as input to an LTI filter with impulse response $h(t) = \delta(t) - \delta(t-4)$ and frequency response $H(j\omega) = 1 - e^{-j\omega 4}$. Modify your `N=10` approximation to compute the periodic output of the filter. Does the computed output look as it should for the convolution $y(t) = h(t) * x(t)$? Explain with a brief comment in the script file.

(g) Now clean up your script file and print it out to hand in.

3.4 Synthesis of a Full-Wave Rectified Sine Wave

Start a new script file for this exercise using the MATLAB editor. You will add to this file as you progress through the following. In this part, use the signal definition given in (3). Most of the following tasks are similar to those for the pulse signal, so a copy of the MATLAB script file for that part would be a good starting point for this part.

(a) Using `subplot(211)`, make a plot of the ideal waveform $x(t)$ over the range $-1/120 \leq t \leq 1/120$, which would be two periods of the periodic signal. Use at least 200 samples across each period in order to obtain a smooth plot. Label the time scale in terms of seconds.

(b) Using the formula (3) obtained in Section 3.1.2, compute the Fourier series coefficients for `-N:1:N` where `N=5`. Now with `subplot(212)` and `stem( )`, make a plot of the spectrum of the signal. Label the frequency axis as frequency assuming that the period is $T_0 = 1/120$.

[1] Remember that in order to use the `syn_fourier()` function for Fourier synthesis, we must include all the $\{a_k\}$ coefficients for both the positive and negative indices k. Likewise, the frequency vector `fk` must contain both positive and negative harmonics.

(c) Now assume `N=5` and compute a vector of the corresponding frequencies `fk` for the coefficients `ak`. Use your M-file `syn_fourier( , , )` to compute an approximation to the full-wave rectified sine signal, and using the `hold` command, add this curve in a different color or line type to the plot of the ideal waveform. Use a time sampling that is fine enough to give a smooth plot. Be sure to add the appropriate instructions to your script.

(d) Using a finite number of terms is exactly equivalent to passing the ideal discontinuous pulse signal through an ideal lowpass filter. What is the cutoff frequency of the ideal lowpass filter if we want the output to be a constant as required in a DC power supply? Is your answer unique? What is the value of the constant output, and how is it related to the Fourier coefficients? Put your answers to these questions as a comment in your script.

(e) Investigate what happens in this case as `N` increases. Increase `N` until you can no longer see a difference on the computer screen between the synthesis and the ideal full-wave rectified signal. What is this number? What do you observe as `N` increases? Is the behavior of the approximation different from the case of the periodic pulse signal? If so, how is it different and why? Again add this as a comment in your script.

(f) Now consider passing the signal $x(t)$ through a filter with impulse response $h(t) = \alpha e^{-\alpha t} u(t)$.[2] The purpose of such a lowpass filter would be to extract the DC component of the full-wave rectified signal. Use your `N=5` approximation with $\alpha = 100\pi$ to compute the periodic output of the filter, and plot it on the same graph as the full-wave rectified sine wave. Does this filter do a good job of extracting the DC component, or are there "ripples" on the output signal? Experiment with different values of α to find a filter that leaves only a small ripple in the output.

(g) The previous experiments should convince you that the ripple size decreases with decreasing α. This is because the nominal cutoff (3 dB) frequency of the filter is, equal to α. Assuming that the fundamental frequency component of the input full-wave rectified sine wave is the major contributor to the ripple, derive a formula for the peak-to-peak ripple size as a function of α. You make a crude verification of your formula by measuring the ripple size on a plot for various values of α.

4 Summary

As a report on this lab experiment, include the following:

1. Your work on verifying the formulas for the Fourier coefficients of the pulse signal and the full-wave rectified sine wave.
2. A printout of your M-file `syn_fourier( )`.
3. Printouts of your annotated scripts for each of the waveforms as well as copies of the plots that the scripts produce.
4. Your formula for the peak-to-peak ripple remaining in the filtered output of the full-wave rectified sine wave.

[2] Such a filter could be implemented with a simple RC circuit.