Signal Processing First
Lab 06: Fourier Series

Pre-Lab and Warm-Up: You should read at least the Pre-Lab and Warm-up sections of this lab assignment and go over all exercises in the Pre-Lab section before going to your assigned lab session.

Verification: The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply demonstrate the step to the TA or instructor. Turn in the completed verification sheet to your TA when you leave the lab.

1 Introduction & Objective

The goal of the laboratory project is to show that Fourier Series analysis is a powerful method for predicting the response of a system when the input is a periodic signal. Since we will be doing Fourier Series for continuous-time signals, the formulas are integrals. As a result we will use MATLAB's Symbolic Toolbox which actually relies on MAPLE to do symbolic integration and symbolic algebra. Finally, we will use Fourier Series to analyze the same power supply problem as in Lab 04, but now we can analyze the problem in the frequency domain.

2 Background: Fourier Series Analysis and Synthesis

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:

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

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 is [-1/2 T_0 , 1/2 T_0] sometimes a convenient one, but integrating over the interval [0, T_0]
would also give exactly the same answer.

3 Pre-Lab

In this project, we will use a feature of MATLAB that you may not have experienced. MATLAB has a symbolic toolbox that is based on the symbolic algebra program called MAPLE, which you might have used in calculus. With this toolbox we can evaluate mathematical expressions symbolically and then use MATLAB's numerical functions and plotting functions to evaluate formulas and display results. We will use only a rudimentary set of the symbolic capabilities.

3.1 Symbolic Variables

You can define symbolic variables in several ways. The easiest is to simply type

```
syms v t A omega phi
```

Note that we use spaces instead of punctuation. This defines the variables `v, t, A, omega,` and `phi` as symbolic variables. Another way is to type `v=sym('v').` Now when we use these variables they are interpreted as symbolic variables and we can make up symbolic mathematical expressions using the operators of MATLAB. For example the statements

```
syms v t A omega phi
v = A*cos(omega*t + phi)
```

would be used to define a "symbolic cosine wave", which would be written in standard mathematical notation as

$$v(t) = A\cos(\omega t + \varphi).$$

Type `help whos` to see how the symbolic variable is stored in the workspace; also consult `help syms` to learn more

3.2 Symbolic Integrals and Derivatives

MATLAB can do integration and differentiation with the functions `int()` and `diff()`. Verify that the following give the expected result from Calculus:

```
syms t A omega phi
xt = int(A*t^3 + pi, t)
yt = diff(A*cos(omega*t + phi), t)
zz = int(A*cos(omega*t + phi), t, 0, 1/omega)
zsimp = simple(zz)
```

Notice that you have to tell these functions which variable is the "variable of interest" for taking the derivative or doing the integral. The fourth line shows that you can also perform a definite integral by giving the limits of integration; and the limits can be symbolic. Furthermore, the last line shows that you might want to force MAPLE to simplify the formula.

Idea: you might speculate on using `int()` to do an integral that would arise in convolution such as:

$$\int_0^t e^{-(t-\tau)}d\tau$$

3.3 Substituting Values

We generally want to substitute values for some of the symbolic variables, so that we can evaluate the functions that they represent. For example, we could obtain a symbolic cosine wave with amplitude `A=100`, `omega=120*pi`, and `phi=0` by using the `subs( )` function:

```
v1 = subs(v, {A,omega,phi}, {100,120*pi,0})
```

which produces the new symbolic cosine wave `v1=100*cos(120*pi*t)`. Consult `help subs`.

We can evaluate `v1` which is a symbolic function of `t` for 101 values covering one period, and then convert the values to MATLAB's double precision floating-point form using the statements:

```
tn = (0:0.01:1)/60;
v1n = double( subs(v1,t,tn) );  %-- convert to numeric
plot(tn, v1n)
```

Type `help double` to learn more about converting to the numerical format called double-precision.

Although we have use the `plot( )` function to plot these numeric values, there is another way that works much nicer for symbolic functions. This is `ezplot( )` which figures out the grid to use and frees the user from specifying the vectors `tn` and `v1n` above. Consult `help ezplot` to learn more.

```
ezplot(v1, [0,1/60])
```

4 Warmup

4.1 Symbolic Function for Fourier Synthesis

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

$$x_N(t) = \sum_{k=-N}^{N} a_k e^{jk\omega_0 t} \quad (3)$$

To illustrate how we can use symbolic math in MATLAB, consider the following function for evaluating (3):

```
function xt = fouriersynth( ak, N, T0 )
%FOURIERSYNTH  synthesize Fourier Series formula SYMBOLICALLY
%
% usage: xt = fouriersynth( ak, N, T0 )
%
%  ak = (symbolic) formula for the Fourier Series coefficients
%   N = (numeric) use the Fourier coeffs from -N to +N
%  T0 = (numeric) Period in secs
%  xt = (symbolic) signal synthesized
%
% example:
%   syms ck k xt
%   ck = sin(k)/k
%   xt = fouriersynth( ck, 4, pi );
```

```
syms t k wwk
kk = -N:N;
try
    ak_num = subs( ak, k, kk+((kk==0)+sign(kk))*1e-9 )
catch
    error('FOURIERSYNTH: ak must use k as its variable')
end
wwk = exp(j*(2*pi*kk'/T0)*t);
xt = simple( ak_num*wwk );   %-- inner product
```

This function takes in one *symbolic* variable and two *numeric* variables: `ak` is a symbolic expression for the Fourier coefficients a_k as a function of k; `N` controls the number of Fourier coefficients to use in the synthesis, and `T0` specifies the numeric value of the period for the synthesized signal.

(a) You can download `fouriersynth.m` for later use, or you can type it into a file with that name. Study this program and explain what the last two lines do. Explain what is contained in the vector `ak_num`. Explain how this program creates a symbolic variable that is the same as the Fourier Synthesis summation formula (3).

(b) Here is an example of how we would use this function to synthesize a periodic function whose Fourier coefficients are given by $a_k = \sin(k)/k$ and whose period is 2.

```
syms wt t ak k
N = 10;  T0 = 2;
ak = sin(k)/k;
wt = fouriersynth(ak, N, T0);
ezplot(wt, [-To,T0]); grid on
axis tight    %---make ezplot show the whole thing
```

Type in these instructions and observe the plot. Try increasing N and running the program again. What happens as you increase N? What signal would result when $N \to \infty$? What is the DC value of the signal? From the plot? From the a_k formula?

4.2 Symbolic Integration and Fourier Analysis

Using MAPLE, MATLAB can evaluate Fourier Series analysis integrals such as (2). Here is an example:

```
syms t xt XNt ak wwk k omega
T0 = 6;
xt = sin(2*pi*t/T0);
wwk = exp(-j*(2*pi*k/T0)*t)
ak = (1/T0)*int( xt*wwk, t, 0, T0/2)
ak = simple( ak )
N = 9;
xNt = fouriersynth( ak, N, T0);
ezplot(xNt, [-T0/2,2*T0]); grid on
```

Note that we first obtain a symbolic expression `ak` for the coefficients by doing an integral followed by a simplification. Then we synthesize the signal.

Type in this program and run it. Give an equation valid over one period for the exact signal that is Fourier analyzed by the above integration. What is the fundamental period and the fundamental frequency? The semicolon `;` is omitted from the expression for `ak` above, so the general expression for the Fourier coefficients will be typed out. Write the mathematical formula that is the general expression for a_k in this case. Try larger or smaller values of N to see the convergence of the approximate synthesis.