

GEORGIA INSTITUTE OF TECHNOLOGY
School of Electrical and Computer Engineering

EE2201

Project No. 2 – Frequency-Domain Analysis

Date Assigned: May 3, 1999 (EE2201A), May 4, 1999 (EE2201B)

Date Due: May 19, 1999 (EE2201A), May 20, 1999 (EE2201B)

1 Warmup

In this project we will use a feature of Matlab that you may not have experienced. Matlab has a symbolic tool box that is based on MAPLE. With this tool box we can evaluate mathematical expressions symbolically and then use Matlab's numerics and plotting to evaluate formulas and display results. We will use only a rudimentary set of the symbolic capabilities.

1.1 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

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

Type `help syms` to learn more.

1.2 Substituting Values

We generally want to substitute values for 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; i.e.,

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

produces the new symbolic cosine wave `v1=100*cos(120*pi*t)`. Type `help subs` to learn more.

We can evaluate this symbolic function of `t` for 101 values covering one period and convert the values to Matlab's double precision floating-point form using the statement

```

tn=(0:.01:1)/60;
v1n=double(subs(v1,t,tn));
plot(tn,v1n)

```

Type `help double` to learn more. Note that we have use the `plot()` function to plot these numeric values. Another way to plot values of a symbolic function is to use `ezplot()` as in

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

Matlab will decide on the spacing of points for you. Type `help ezplot` to learn more.

1.3 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. The Fourier synthesis equation for a periodic signal $x(t) = x(t + T)$ is

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

where $\omega_0 = 2\pi/T$. 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

$$x_N(t) = \sum_{k=-N}^N a_k e^{jk\omega_0 t}. \quad (2)$$

To illustrate how we can use symbolic math in Matlab, consider the following Matlab function, which evaluates Eq. (2):

```

function x=fouriersynth(a,T)
%USAGE x=fouriersynth(a,T)
% a=(numeric) array of Fourier series coefficients
%      (a must be odd length 2*N+1)
% T=(numeric) period of time signal
% x=(symbolic) signal synthesized
(1)  a=a(:);
(2)  syms t
(3)  N=(length(a)-1)/2;
(4)  k=-N:N;
(5)  w=exp(j*2*pi*k*t/T);
(6)  x=simple(w*a);

```

This function takes in two numeric variables: **a** is a numeric row or column vector of odd length $2*N+1$, which contains numerical values of the Fourier coefficients with the “DC” coefficient in location $N+1$. **T** is the numeric value of the period.

Problem 1: Type the above function into a file named `fouriersynth.m` for later use. Study this program and explain what each of the numbered statements does. Explain how this program gives a symbolic variable that is the same as Eq. (2).

Here is an example of how we would use this function to synthesize a periodic function whose coefficients are all equal to $1/2$ and whose period is 2.

```

syms t ak w k
N=10;T=2;
a=ones(1,2*N+1)/2;
w=fouriersynth(a,T);
ezplot(w,[-T,T]);grid
axis([-2 2 -3 11]) %ezplot doesn't seem to want to show the whole thing

```

Problem 2: 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 $N \rightarrow \infty$?

Now suppose that we have found the Fourier series coefficients to be

$$a_k = \begin{cases} -j \frac{4 \sin(\pi k/2)}{\pi^2 k^2} & k \neq 0 \\ 0 & k = 0. \end{cases} \quad (3)$$

Problem 3: Modify the above Matlab program to use the Fourier coefficients in Eq. (3) and run it. (Recall that you can make a vector ranging from -10 to +10 with the Matlab statement `-10:10`. Use this to evaluate the Matlab version of the formula in Eq. (3). Matlab will complain if you evaluate the top formula in Eq. (3) for $k = 0$. All you have to do is put in the correct value as a separate statement.) What is the signal whose Fourier coefficients are given by Eq. (3)? What happens if you change the period to $T=4$ while keeping the coefficients the same? Does the amplitude of the signal change? Change the DC level to 1. What is the effect of this change on the waveform? Increase and decrease N . What is the effect of this parameter? Where does the approximation error seem to be largest? Hand in a copy of the plot for $T=2$, $N=20$ and DC level 1.

Problem 4: Modify your program to synthesize the derivative of the signal. Test your program. Does the result look like the derivative of the original signal? What happens at the points of discontinuity of slope of the original signal? Also modify your program to produce the derivative of the signal delayed by 1 second. Hand in a plot of this showing the cases $N = 10, 20, 40$ all on the same graph.

1.4 Symbolic Integration and Fourier Analysis

To determine the Fourier series coefficients from a signal, we must evaluate the integral

$$a_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-jk\omega_0 t} dt \quad (4)$$

where $\omega_0 = 2\pi/T$. Using MAPLE, Matlab can evaluate integrals like this for you. Here is an example.

```

syms t ak w k
ak=simple( (1/4)*int(t*sym('exp(-j*2*pi*k*t/4)'),t,-1,1) )

```

```

N=10;
a=double( subs(ak,k,-N:N) );
a(N+1)=0;
w=fouriersynth(a,4);
ezplot(w,[-4,4]);grid

```

Note that we first obtain a symbolic expression ak for the coefficients. Then we obtain numeric values a . (In this case, the DC value gives a problem because of dividing by zero so we just put in the correct value.) Then we synthesize the signal.

Problem 5: Type in this program and run it. Give an equation valid over one period for the exact signal that is analyzed by the above integration. What is the period? If you leave off the ; on the expression for ak as above, the general expression for the Fourier coefficients will be typed out. What is the general expression for a_k in this case? Try larger or smaller values of N in the approximate synthesis. What happens at the discontinuity?

2 Analysis of a Power Supply Circuit

Figure 2 depicts a circuit for creating a DC voltage from an AC voltage. The diode bridge implements a full-wave rectifier, and the RC circuit provides lowpass filtering. The block

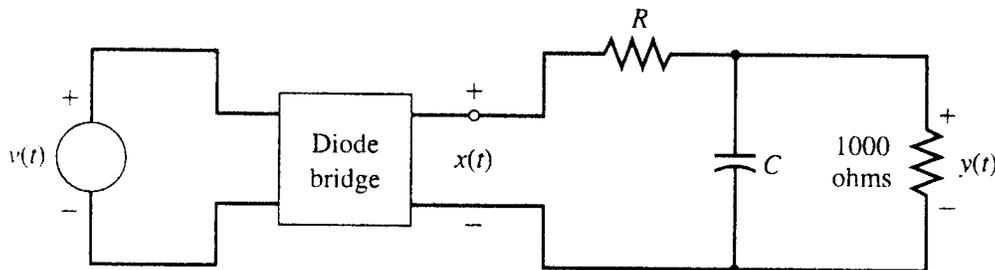


Figure 1: Diode bridge power Supply circuit.

diagram shown in Figure 2 represents the operations performed by the different parts of the circuit. The full-wave rectifier is defined by $x(t) = |v(t)|$. Its purpose is to generate a periodic signal with a non-zero DC component. The purpose of the lowpass filter is to remove most

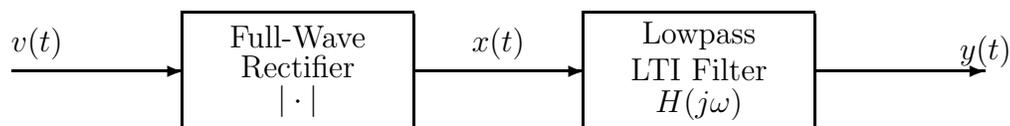


Figure 2: Block diagram representation of power supply.

of the high frequencies in the output of the rectifier, leaving the DC component.

Problem 6: Using impedance methods, determine the frequency response of the lowpass filter, $H(j\omega) = Y(j\omega)/X(j\omega)$. Your answer should be in terms of R and C .

2.1 Fourier Analysis of the Full-Wave Rectifier Output

Assume that the input is a power line voltage $v(t) = 120\sqrt{2} \cos(120\pi t)$.

Problem 7: Make symbolic Matlab expressions for the functions $v(t)$ and $x(t)$. The output of the full-wave rectifier is a periodic signal such that $x(t) = x(t + T)$. What is the period T ? Use `subplot()` and `ezplot()` to make a plot showing both $v(t)$ and $x(t)$ over the range $-2T \leq t \leq 2T$.

Since $x(t)$ is periodic, it can be represented approximately by a truncated Fourier series of the form of Eq. (2) where $\omega_0 = 2\pi/T$ and the Fourier coefficients are given by Eq. (4).

Problem 8: Write a program to evaluate the symbolic Fourier coefficients of $x(t)$. Then evaluate the numeric values of the coefficients for $N = 10$ and use these to synthesize an approximation $x_{10}(t)$. Use `subplot(31x)` to make a plot with the exact $x(t)$ at the top and $x_{10}(t)$ in the middle plot. (The third space will be used for plotting the output $y(t)$ of the lowpass filter.) At what points is $x_{10}(t)$ most different from the exact $x(t)$?

Now we know that the exact output of the lowpass filter is

$$y(t) = \sum_{k=-\infty}^{\infty} a_k H(jk\omega_0) e^{jk\omega_0 t} \quad (5)$$

where the a_k are the Fourier coefficients of $x(t)$; i.e., the Fourier coefficients of the output are $b_k = a_k H(j\omega_0)$.

Problem 9: Suppose that we require that the DC component of the output be 5 volts. Use the frequency response $H(j\omega)$ found in Problem 6 to find R and C so that the DC component of the output is 5 volts, and so that the magnitude of the component at $\omega = \omega_0$ is less than or equal to 1% of its magnitude in $x(t)$.

Using our $2N + 1$ term approximation for the input, the approximate output is

$$y_N(t) = \sum_{k=-N}^N a_k H(jk\omega_0) e^{jk\omega_0 t}. \quad (6)$$

Problem 10: To evaluate Eq. (6), you need to create a symbolic function for the frequency response using the values of R and C determined in Problem 7. Evaluate this symbolic function at the correct frequencies, obtaining numerical values for $H(j\omega_0)$, and then multiply these by the appropriate numerical values of the Fourier coefficients used to synthesize $x_{10}(t)$. (If \mathbf{H} is a vector of frequency response values and \mathbf{a} is a vector of Fourier coefficients for the input, then $\mathbf{a}.*\mathbf{H}$ would be the Fourier coefficients of the output.) Use this approach to synthesize the output $y_{10}(t)$ and plot it in the third part of the plot begun in Problem 6.

2.2 Comparison to the Solution of Project 1

Problem 11: Figure 3 shows the solution that was obtained for Project 1, where the power supply circuit was approximated using finite-difference methods. Compare this figure to your result in Problem 9. Why does the simulation in Figure 3 start at $t = 0$ and build up to the steady state, while your result in Problem 9 does not?

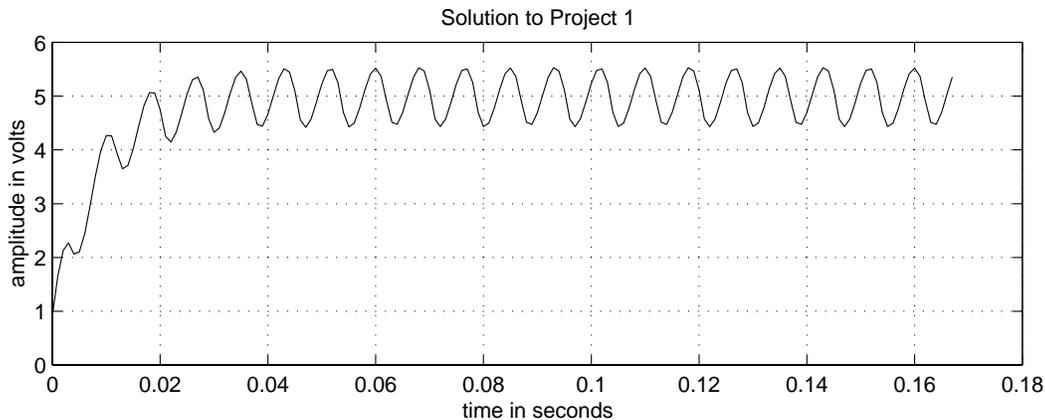


Figure 3: Solution to Project 1.

2.3 Optional Problem

Problem Optional: You may wish to experiment with different values of R and C to see the effect of the filter parameters on the output. Another instructive thing to do is to make a three part plot with the numeric values of the a_k s plotted versus frequency on top (i.e., a `stem()` plot of the spectrum of the input), the frequency response of the lowpass filter in the middle (evaluate it over a finely divided frequency scale to show what happens between the frequency components of the input), and the b_k s versus frequency on the bottom (i.e., a `stem()` plot of the spectrum of the output).

3 Report

Prepare a concise report on this project. Organize it by giving answers to each of the eight problems posed above. In some cases you will simply need to give brief answers to any explicit questions that were asked. In all cases, give listings of MATLAB functions and scripts that you use to solve the problem. Also include (possibly hand annotated) hard copies of the graphs to refer to in your discussion.