

ECE 460 – Introduction to Communication Systems
MATLAB Tutorial #1

Evaluating Exponential Fourier Series

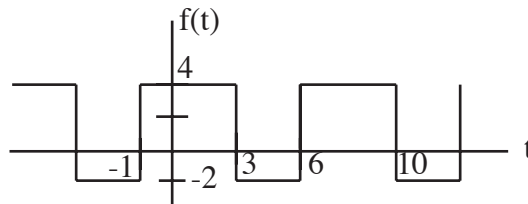
The homework assignments in this course contain problems that must be completed using MATLAB. A minimal knowledge of MATLAB is required to get started. Advanced MATLAB features will be introduced in tutorials posted on the homework web page. Students who have not used MATLAB before should go to the following web page:

http://www.mathworks.com/academia/student_center/tutorials/launchpad.html

View the “Getting Started” and “MATLAB Examples” videos for an overview. Then proceed to the Interactive MATLAB Tutorials and view “Navigating the MATLAB Desktop” and “MATLAB Fundamentals.” This should give you enough information to proceed with the remainder of this tutorial. You may need to refer to the other Interactive MATLAB Tutorials if unfamiliar commands are used. Additionally, MATLAB has extensive online help and documentation.

Finding the Exponential Fourier Series Coefficients

For the waveform, $f(t)$, shown below:



we can write the exponential Fourier Series expansion as:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t}$$

where

$$F_n = \frac{1}{T} \int_T f(t) e^{-jn\omega_0 t} dt = \frac{1}{7} \left[\int_{-1}^3 4e^{-jn2\pi t/7} dt - \int_3^6 2e^{-jn2\pi t/7} dt \right]$$

$$= \dots = \frac{1}{jn\pi} \left[-3e^{-jn6\pi/7} + 2e^{jn2\pi/7} - e^{-jn12\pi/7} \right]; \quad n \neq 0$$

and

$$F_0 = \frac{1}{T} \int_T f(t) dt = \frac{1}{7} \left[\int_{-1}^3 4 dt - \int_3^6 2 dt \right] = \frac{1}{7} [16 - 6] = \frac{10}{7}$$

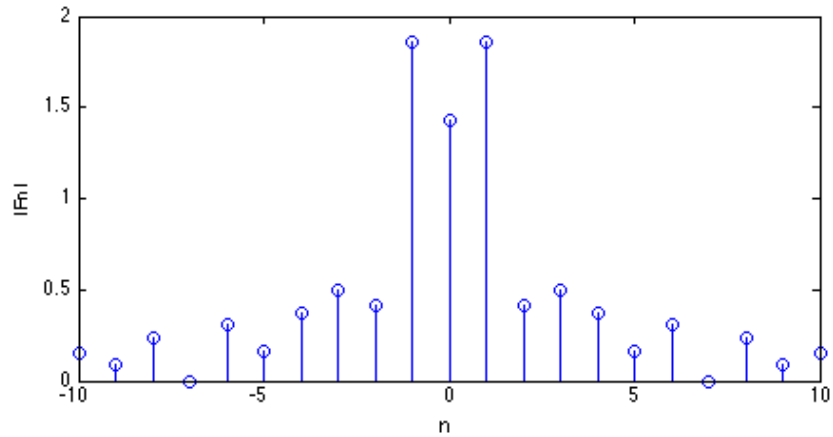
These coefficients may be simplified further but since we will be using MATLAB to evaluate them that is not necessary.

Plotting the Amplitude and Phase Spectrum

In order to plot the amplitude and phase spectrum of this pulse train we need to evaluate the values of F_n . Due to the divide by zero condition that occurred above, we need to handle the case for $n = 0$ separately. There are a number of methods for handling this. One method is to evaluate the positive, negative and zero values of n separately. In this case we will plot the spectrum for $-10 \leq n \leq 10$ using the following MATLAB commands.

```
>> clear
>> nneg=-10:-1;
>> npos=1:10;
>> Fnneg=(1./(j*nneg*pi)).*(-3*exp(-j*nneg*6*pi/7) ...
+2*exp(j*nneg*2*pi/7)+exp(-j*nneg*12*pi/7));
>> Fnpos=(1./(j*npes*pi)).*(-3*exp(-j*npes*6*pi/7) ...
+2*exp(j*npes*2*pi/7)+exp(-j*npes*12*pi/7));
>> F0=10/7;
>> n=[nneg, 0, npos];
>> Fn=[Fnneg, F0, Fnpos];
>> stem(n, abs(Fn))
```

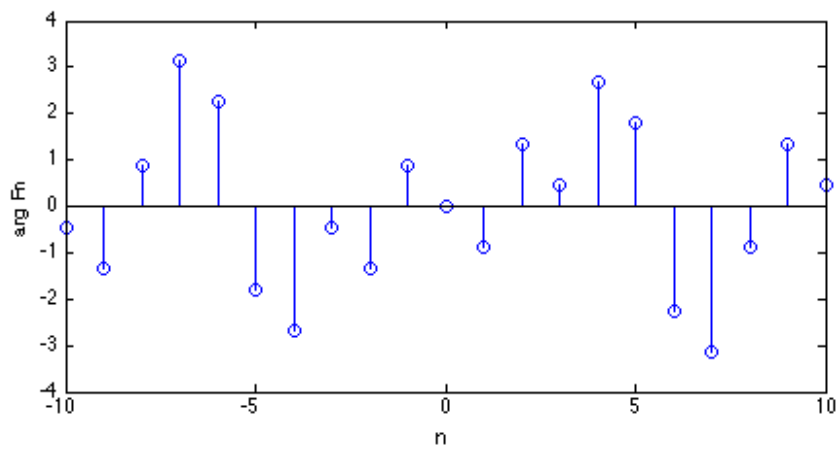
which results in the following amplitude spectrum:



The phase spectrum can then be found using:

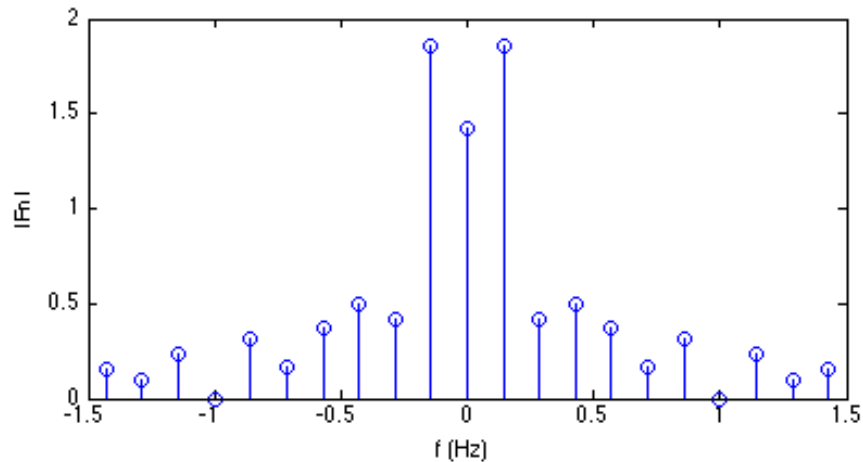
```
>> stem(n, angle(Fn))
```

which results in the following phase spectrum:



In this example, the fundamental frequency, $f_0 = 1/T = 1/7$. Thus, the amplitude spectrum can also be plotted versus the frequency in Hz with:

```
>> stem(n*(1/7), abs(Fn))
```

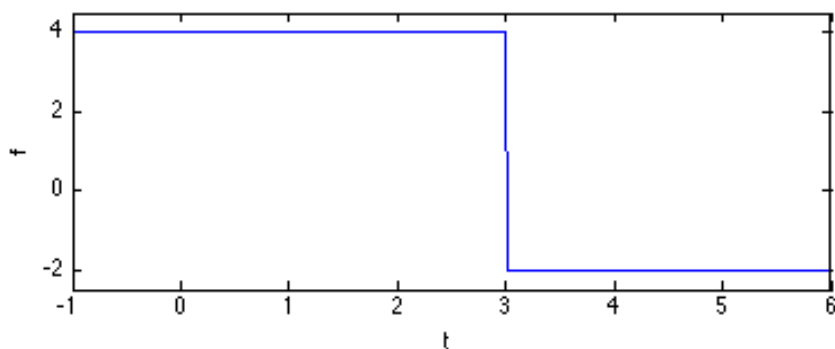


Plotting Functions with Discontinuities

In order to observe the accuracy of a truncated Fourier Series we need to plot it along with the exact function. There are many methods for plotting the exact function $f(t)$. The following method makes use of logical operators.

```
>> t=-1:.01:6;
>> fexact=4*(t<=3)-2*(t>=3);
>> plot(t,fexact)
```

and results in the plot:



Plotting the Truncated Fourier Series

We can use the truncated exponential Fourier series as an approximation to the function, $f(t)$. Recall that we must always use a symmetric range of n values ($-n_0 \leq n \leq n_0$) to obtain a real function. For $n_0 = 3$:

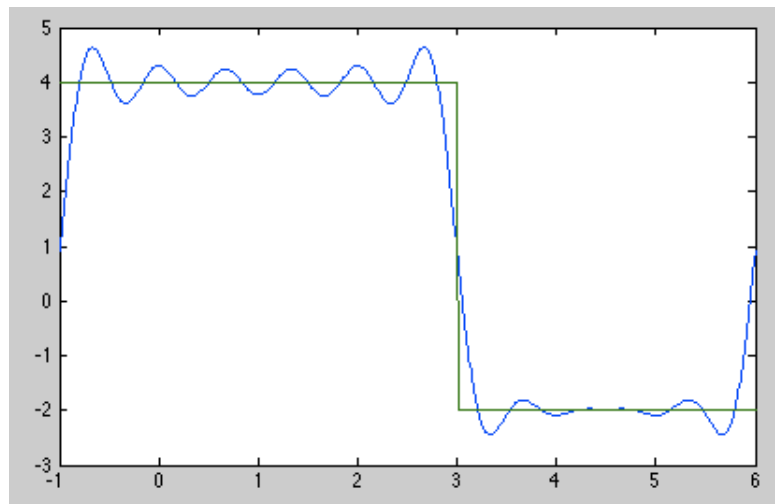
```
>> clear
>> nneg=-3:-1;
>> npos=1:3;
```

```

>> n=nneg;
>> Fnneg=(1./(j*n*pi)).*(-3*exp(-j*n*6*pi/7) ...
    +2*exp(j*n*2*pi/7)+exp(-j*n*12*pi/7));
>> n=npos;
>> Fnpos=(1./(j*n*pi)).*(-3*exp(-j*n*6*pi/7) ...
    +2*exp(j*n*2*pi/7)+exp(-j*n*12*pi/7));
>> F0=10/7;
>> n=[nneg,0,npos];
>> Fn=[Fnneg,F0,Fnpos];
>> k=0;
>> for t=-1:.01:6
k=k+1;
fapprox(k)=sum(Fn.*(exp(j*n*2*pi*t/7)));
end
>> t=-1:.01:6;
>> fexact=4*(t<=3)-2*(t>=3);
>> plot(t,fapprox,t,fexact)

```

Note that the first 10 lines calculate the F_n values as shown in a previous section. This is followed by a for loop that evaluates the series summation for each value of t . The values are placed in a vector `fapprox`. The following curves result from this MATLAB code:



Using only seven terms the truncated Fourier series makes a reasonable approximation to the pulse train. Note that as more terms are added the approximation improves. For $n_0 = 30$:

