Signals Systems and Transforms
EEET-332
Lab 10

For each section using MATLAB:

1) Create a new *.m (or *.mlx) file and call init() in the first line. Save it. Remember, no spaces in the file name!
2) Copy last lab's init.m, fft_ifft.m, make_plot.m and make_stem.m (or *.mlx) functions to the same directory. Use the init and plot files defining fig_num as a global so that both make_plot and make_stem can use it.

A quiz will be given at the beginning (1st 10 minutes) of the lab covering the content of the prelab. One quiz will be dropped. NO make-up quizzes will be given.

Prelab:

1) Based on the script below, determine the value of each expression.
   a. What are the values of t?
   b. How many zeros are in the y1 array?
   c. What are the values of y2 when the script finishes?
   d. What are the values of y3 when the script finishes?

```
T=5;
t=0:T/5:T
y1=zeros(size(t))
y2=zeros(size(t))
t_greater_than_2=find(t>2);
y2(t_greater_than_2)=3*t(t_greater_than_2)
y3=zeros(size(t))
t_between_1_and_3=find(t>=1 & t<=3);
y3(t_between_1_and_3)=3
```

Signals Systems and Transforms
EEET-332
Lab 10

Note: You may change variable names and comments to make them make more sense to you. Section 1 will be used with several homework problems.

Section 1: FFT and IFFT – Create a new script named section1.m for the code below.

```
init();
N=16; %number of samples in time and freq domain
n=0:N-1; %index for freq domain.
T=9; %signal period
Ts=T/N; %sample period
t=0:Ts:T-Ts;
```

1) Calculate Ts and ws (sample angular frequency = 2*pi/Ts)

Ts = 0.5625 sec ws = 11.1701 rad/sec

2) Complete the table for the t array

0.0000	0.5625	1.125	1.6875	2.25	2.8125	3.3750	3.9375	4.5000	5.0625
5.6250	6.1875	6.75	7.3215	7.8750	8.4375				

Signals Systems and Transforms
EEET-332
Lab 10

3) Add the MATLAB code below to your script to create the input waveform shown in Figure 1.

Note: The plot command in make_plot.m (or *.mlx) was **changed from plot(x_data,y_data) to plot(x_data,y_data,'o')** to show the data points seen in figure 1.

```
y=zeros(size(t));
t_between_1_and_2=find(t>=1 & t<=2);
y(t_between_1_and_2)=-t(t_between_1_and_2);
t_between_2_and_7=find(t>2 & t<7);
y(t_between_2_and_7)=0;
t_greater_than_7=find(t>= 7 );
y(t_greater_than_7)= t(t_greater_than_7);
make_plot(t,y,'Input function','n','y');
```

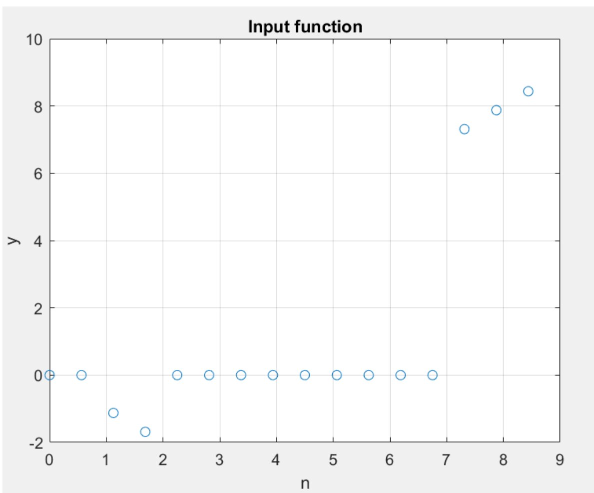


Figure 1.

4) Copy fft_ifft.m (or *.mlx) from the last lab and call it from your script after the make_plot command. Make sure you include returned variables in your function call.

The fft_ifft function will compute the Fourier transform, and the inverse Fourier transform (returning the original function). Frequency, spectrum, and Fourier are all used to refer to the fft data. The subscript m is generally used in the frequency domain, and n is used in the time domain. Sometimes m and n are interchanged.

Signals Systems and Transforms
EEET-332
Lab 10

5) Using figure #2 generated by the script (spectrum amplitude) and the “data cursor”, click on the plot to help you complete the list below of c_m values (some are given, rounded to the nearest thousandth).

   a) DC: m = 0 and $|c_m|$ = 1.301
   b) 1st harmonic: m = 1 and $|c_m|$ = 1.449
   c) -1st harmonic: m = 15 and $|c_m|$ = 1.449
   d) 2nd harmonic: m = 2 and $|c_m|$ = 1.339
   e) -2nd harmonic: m = 14 and $|c_m|$ = 1.339

6) (True/False) The DC value is the average value of the waveform True

MATLAB’s arrangement of the FFT is confusing and inconvenient. Change the fft_ifft.m function (or *.mlx) so it shifts the spectrum, putting the DC value in the center. Changes are highlighted.

```
function [m_ctr,cm_ctr,yy] = fft_ifft(t,n,y,N)
  % Calculate, display F(m).
  % NOTE: Matlab fft() returns N times spectrum so N is divided out
  %       Matlab ifft() used later will scale it back up by N
  m_ctr=-N/2:N/2-1;
  cm_ctr = fftshift(fft(y,N)/N);
  make_stem(m_ctr,abs(cm_ctr),'shifted spectrum','m(center)','abs(cm)');

  % Reconstruct y (called yy) using inverse FFT (IFFT).
  % NOTE: Matlab fft() returns N times spectrum so N is was divided out
  %       Matlab ifft() now expects fft() scale up by N
  yy = real(ifft(N*fftshift(cm_ctr))); % scrub imaginary vestiges
  make_plot(t,yy,'Reconstructed Waveforms','seconds','reconstructed y');
end
```

Do not forget to modify section1.m to call the new fft_ifft function.

7) Repeat the exercise using the Shifted Spectrum Amplitude and the “data cursor” to click on the plot. Use these results to complete the list below of c_m values.

   f) DC: m_ctr = 0 and $|c_m|$ = 1.301
   g) 1st harmonic: m_ctr = 1 and $|c_m|$ = 1.449
   h) -1st harmonic: m_ctr = -1 and $|c_m|$ = 1.449
   i) 2nd harmonic: m_ctr = 2 and $|c_m|$ = 1.339
   j) -2nd harmonic: m_ctr = -2 and $|c_m|$ = 1.339

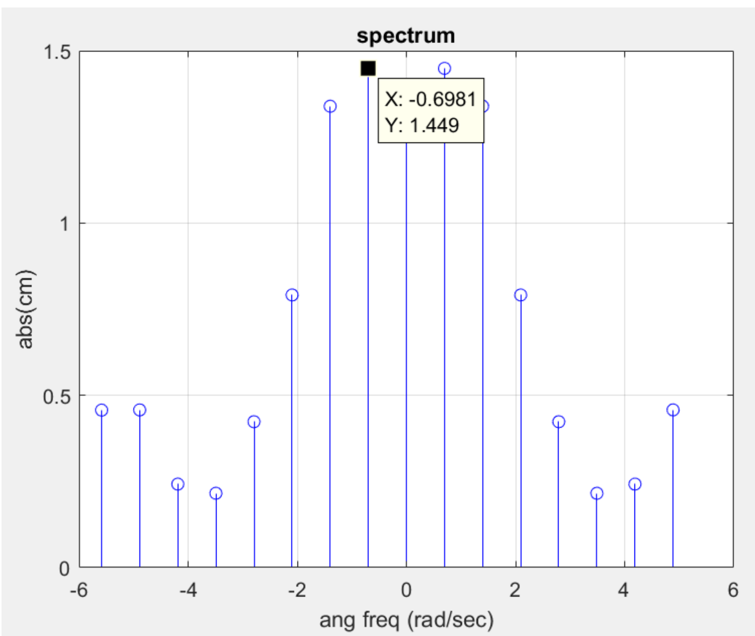
Signals Systems and Transforms
EEET-332
Lab 10

The spectrum can be further enhanced by displaying the actual frequency and not just the harmonic number. The frequency of any coefficient can be calculated by multiplying the harmonic number by the fundamental frequency. Add a make_stem in your script after the call to fft_ifft to plot cm_ctr using the angular frequency for the x-axis variable.

$$angular\ frequency = m\_ctr * 2 * pi/T$$

8) From Figure 4 complete the list below of c_m values (plot given below)

   a) DC: ω = ____0____ and $|c_m(0)|$ = 1.301
   b) 1^{st} harmonic: ω = __0.698132__ rad/sec and $|c_m(1)|$ = ____1.449____
   c) -1^{st} harmonic: ω = = - 0.698132 rad/sec and $|c_m(-1)|$ = 1.449 (shown on plot)
   d) 2^{nd} harmonic: ω = __1.39626__ rad/sec and $|c_m(2)|$ = ____1.339____
   e) -2^{nd} harmonic: ω = -1.39626 rad/sec and $|c_m(-2)|$ = 1.339



Submit:

Section 1 blanks completed (handwritten is acceptable) in your report.

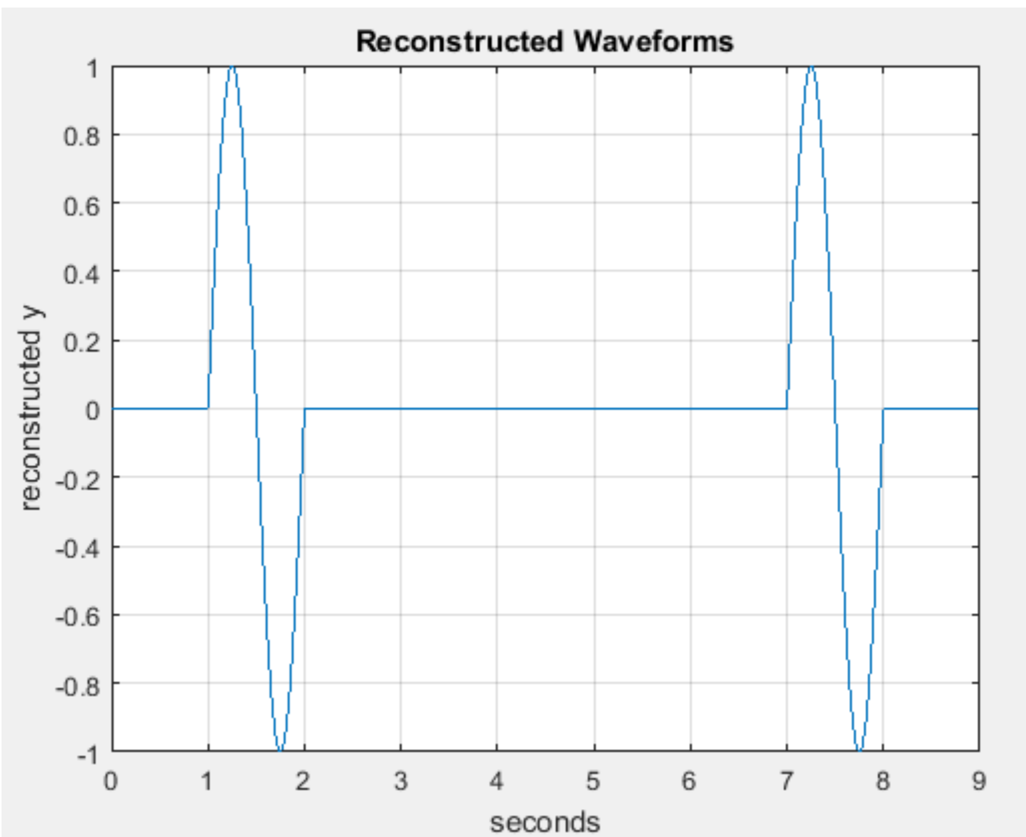
Signals Systems and Transforms
EEET-332
Lab 10

Section 2: Other waveforms

1) Create a new script, named section2.m, and duplicate the code from section1.m into it.
2) Change the function to the sine wave described below.
   a. Set N to 1024.
   b. If you changed make_plot to contain an ‘o’, remove it.
   c. Write new MATLAB code to generate the y shown below (period of sine pulse is Tp=1) where the function is

$$y = \sin\left(\frac{2\pi}{Tp}t\right) when\ 1 \le t \le 2\ and\ 7 \le t \le 8$$

   d. Run the program with the new input and print the reconstructed waveform.



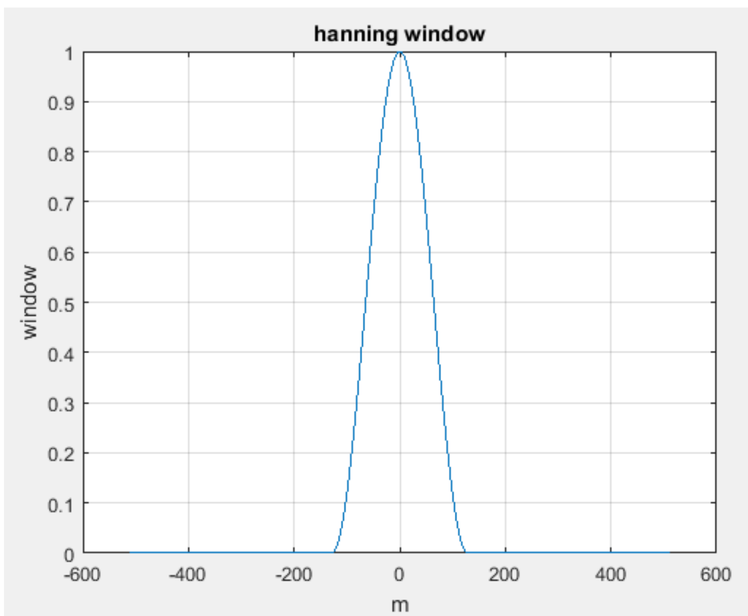
Prepare the reconstructed waveform (Figure #3 generated by section2.m) for sign-off.

Signals Systems and Transforms
EEET-332
Lab 10

Section 3: Windowed spectrum

One application for transforms is data compression, that is, using less data to describe the original signal. In imaging applications, low-frequency data is much more important than high-frequency data. Part of the compression process includes tossing some of the high-frequency cm values. Doing this can cause an overshoot or ringing at simple discontinuities (Gibb's phenomenon). Windowing provides a way to remove high-frequency terms while smoothing the remaining terms, avoiding significant discontinuities and ringing. The sample window below will be multiplied by the spectrum. Its gradual descent to zero will prevent ringing.

Windowing requires creating an array the same size as the spectrum (1024 data points in this case) with zero for the high-frequency values. In the example below, the values of m = -512 to -128 and 128 to 512 are zero. The window is between -128 and 128 and gradually tappers to zero on both ends.



Signals Systems and Transforms
EEET-332
Lab 10

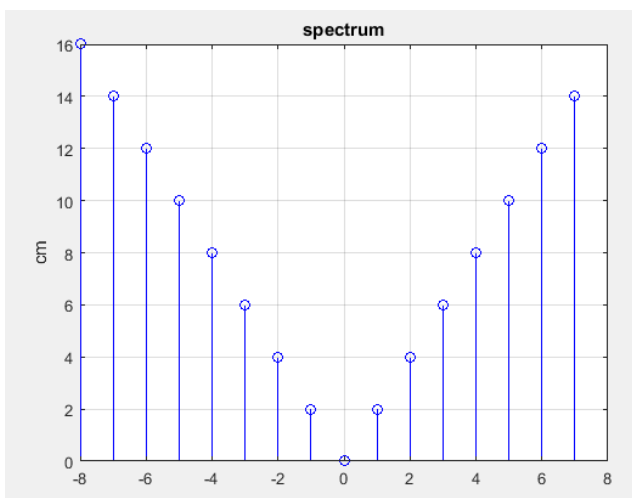
1) Create a new script named section3.m and copy the code below into it.

```
init();
N=16;
M=3;
m=-N/2:N/2-1

cm =                                                        % step a
make_stem(m,cm,'spectrum','m','cm');

win=                                                        % step b
m_between_negM_and_posM=                                    % step c
win(                           ) = hanning(2*M+1)           % step d
make_stem(m,win,'window','m','win');

cm_win=                                                     % step e
make_stem(m,cm_win,'windowed spectrum','m','cm_win');
```

2) Follow the instructions below to practice creating a window. Each step helps you complete a highlighted line of code in the script.
   a. Define cm as a simple function 2*abs(m). Complete the table and verify that you get the same plot. Remember not to use a semi-colon at the end of the cm assignment so you can see the values of cm in the transcript window.

cm

16	14	12	10	8	6	4	2	0	2
4	6	8	10	12	14				



Signals Systems and Transforms
EEET-332
Lab 10

b. Define win as a zero vector the same size as cm. To do this, refer to previous sections to see how the zeros() and size() functions were used. Complete the table and verify that you get the same plot.

win

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0				

The window will now be placed in the center of the win vector.

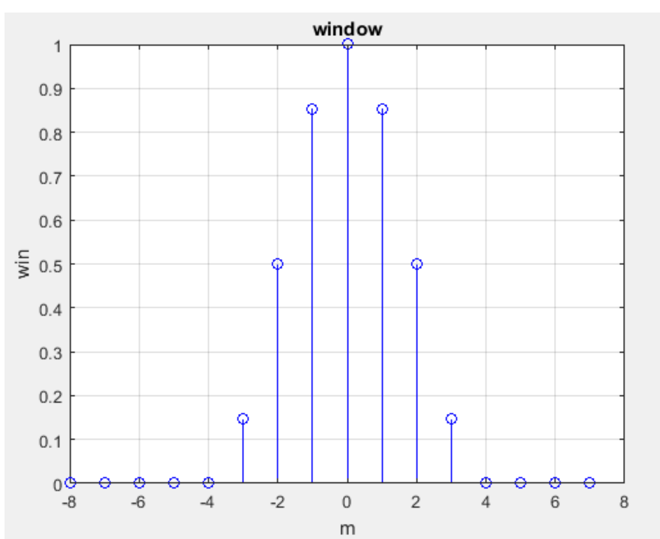
c. Use the find() command to locate the m values between –M and M.
d. Load the Hanning window values in the center of the win vector, from -M to M.

The most common error in this step is "Unable to perform assignment because the left and right sides have a different number of elements." Make sure the size of "m_between_negM_and_posM" is equal to 2*M+1. If not, fix the find command making sure you include the end points, –M and M.

Complete the table and verify that you get the same plot.

win

0	0	0	0	0	0	0.25	0.75	1	0.75
0.25	0	0	0	0	0				

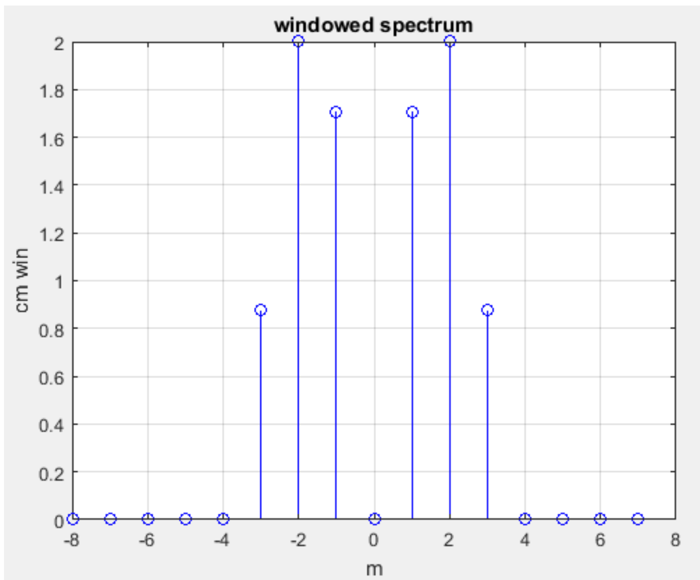


Signals Systems and Transforms
EEET-332
Lab 10

e. Define cm_win by multiplying the cm and win vectors (remember to use a dot). Complete the table and verify that you get the same plot.

cm_win

0	0	0	0	0	0	1	1.5	0	1.5
1	0	0	0	0	0				



Prepare the windowed spectrum plot for sign-off.

Section 4: Reconstructed waveform from the windowed spectrum

Create a new script named section4.m, and duplicate the code from section2.m into it.

The fft_ifft function will be modified to include the Hanning windowing function in the following steps.

Signals Systems and Transforms

EEET-332

Lab 10

1) Create a new script named fft_hanning_ifft.m, and duplicate the code from fft_ifft.m into it.
2) Add hanning to the copied function's name and Mwin to the parameter list, as shown in the box below (highlighted text in the first line). Mwin is the number of non-zero terms in the window.
3) Use the steps below to complete the highlighted lines of code.
    a. Fill the center of win with the hanning window.
    b. Create cm_ctr by multiplying cm with the new window. Do not forget the dot '.' operator in front of the star symbol.

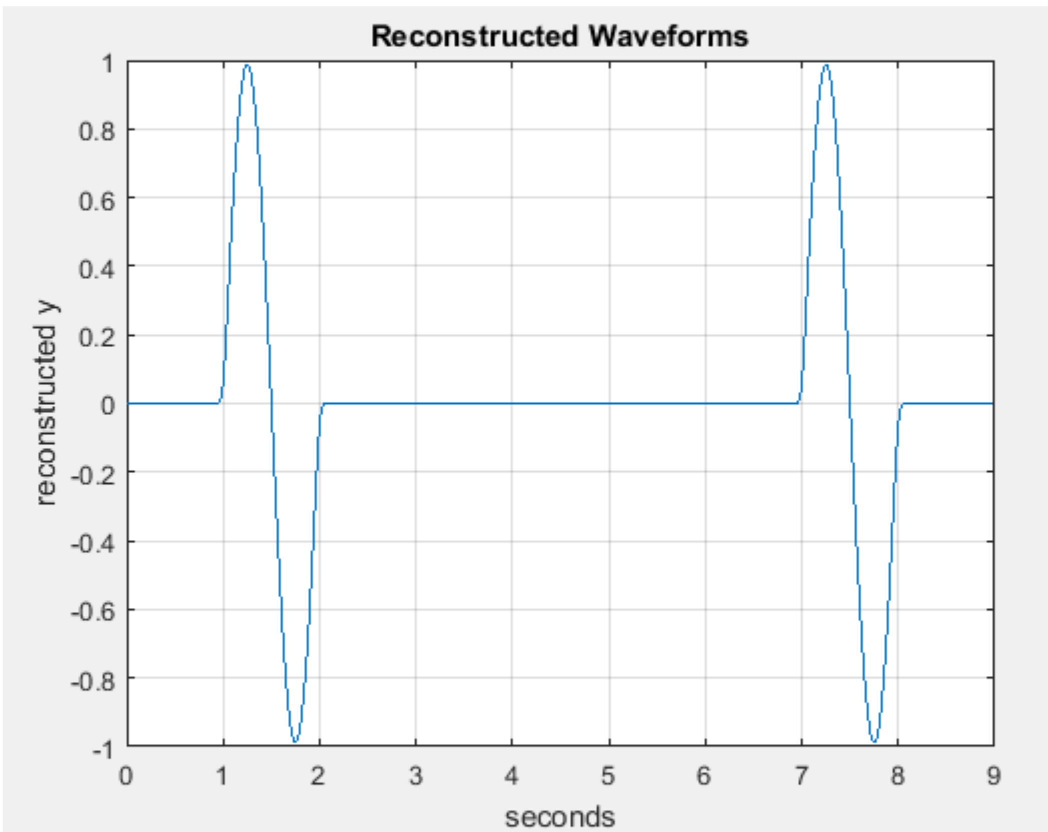
```
function [m_ctr,cm_ctr_win,yy] = fft_hanning_ifft(t,y,N,Mwin)
  % Calculate, display F(m).
  % NOTE: Matlab fft() returns N times spectrum so N is divided out
  %       Matlab ifft() used later will scale it back up by N
  m_ctr=-N/2:N/2-1;
  cm_ctr = fftshift(fft(y,N)/N);
  make_stem(m_ctr,abs(cm_ctr),'shifted spectrum','m(center)','abs(cm)');

  win=zeros(size(cm_ctr));
  win(find(m_ctr==-Mwin):                    )=hanning(2*Mwin+1)';
  cm_ctr_win=                                  ;

  % Reconstruct y (called yy) using inverse FFT (IFFT).
  % NOTE: Matlab fft() returns N times spectrum so N is was divided out
  %       Matlab ifft() now expects fft() scale up by N
  yy = real(ifft(N*fftshift(cm_ctr_win))); % scrub imaginary vestiges
  make_plot(t,yy,'Reconstructed Waveforms','seconds','reconstructed y');
end
```

Signals Systems and Transforms
EEET-332
Lab 10

7) Edit the script section4.m, so it calls the new windowing fft function (new code below). Rerun the script and verify the plot below.

```
Mwin=128; %2*Mwin+1 terms kept in freq domain after windowing
[m_ctr,cm_ctr,yy] = fft_hanning_ifft(t,y,N,Mwin);
```

The reconstructed waveform uses +/- 128 point instead of 1024. Compare the original waveform to the reconstructed waveform. Loosing 768 data points (1024 – 128*2) makes slight differences, especially near fast-changing areas like the peak and start of the sine waves. Sending the transformed data and not the original waveform will save time and is nearly lossless.



Prepare Figure 4 with N = 1024 and Mwin = 128 for sign-off.

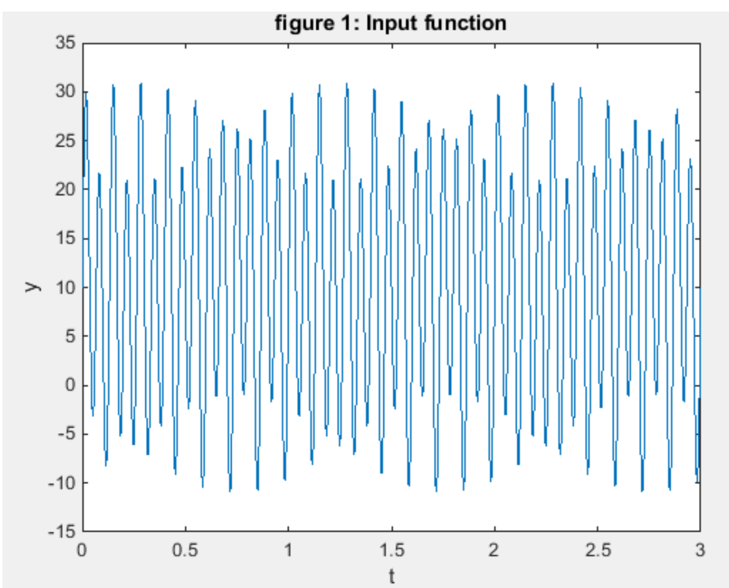
Signals Systems and Transforms
EEET-332
Lab 10

Section 5: Reading the spectrum

1) Create a new script named section5.m, insert and complete the code below to generate the y function. The function repeats every 3 sec (T=3).

$$y = 10 + 5\sin(\omega 1 * t) + 16\sin(\omega 2 * t) \; for\ all\ t, \omega 1 = 8 * 2\pi \; and\ \omega 2 = 15 * 2\pi$$

```
init();
N=1024; %number of samples in time and freq domain
n=0:N-1; %index for freq domain.
T= __________; %signal period
Ts=T/N; %sample period
t=0:Ts:T-Ts;
y=zeros(size(t));

w1 = 8*2*pi;
w2 = ____________;

y = 10 + 5* sin(w1*t)+___________________;

make_plot(t,y,'Input function','t','y');
Mwin=128;
[m_ctr,cm_ctr,yy] = fft_hanning_ifft(t,y,N,Mwin); %Shifted spectrum
omega = m_ctr *2 * pi/T;
make_stem(omega,cm_ctr,'spectrum','ang freq (rad/s)','abs(cm)');
```



Signals Systems and Transforms
EEET-332
Lab 10

Answer the following questions.

2) What is the sample period? $\frac{T}{N} = Ts =$ 0.0029 second
3) What is the sample frequency? $f_s = \frac{1}{Ts} =$ 341.333 Hz or Samples/second
4) What is the fundamental frequency? $fo = \frac{1}{T} =$ 0.333 Hz
5) The frequency of the input sine waves: 8 Hz and 15 Hz.
6) (True/False) This system's Nyquist frequency is much greater than the frequency of either of the input sine waves. True
   The maximum frequency the system can sample without aliasing is the Nyquist frequency. The system Nyquist frequency is: $\frac{f_s}{2}$.

7) Examine the spectrum.

   a. What is the frequency of the 24th harmonic? $24fo =$ 8 Hz

   b. What is m equal to at each of the spikes in the spectrum? Execute section5.m, observe the shifted spectrum plot (Figure 2) created by section5.m and fill in the table.

-45	-24	0	24	-45

   c. Convert these sample numbers to frequency. Fill in the table.

-15 Hz	-8 Hz	0 Hz	8 Hz	15 Hz

Submit the following in your report:

a) Screenshots of the Section5.m code and plots.
b) Answers to questions 2-7 (handwritten is acceptable).

Signals Systems and Transforms
EEET-332
Lab 10

Report:

Create your own cover page.

Submit your cover page, the requested screenshots (sections 1 and 5), and this sign-off sheet on the second page.

Sign-offs

Name ______________________

Section 2: Other waveforms

	/ /
Signature	Date

Section 3: Windowed spectrum

	/ /
Signature	Date

Section 4: Reconstructed waveform from the windowed spectrum

	/ /
Signature	Date