# Appendix B: Creating plots in EE3150

The main purpose of this laboratory is to verify in experimental form the concepts you are learning in EE3350. Typically, in each experiment you will go from mathematical expressions to theoretical plots during the prelab, and then to circuits and experimental results during the lab. In order to make an accurate comparison between the theoretical and experimental results, it is convenient and easier to create the theoretical plots using a computer.

Just like you do to sketch analytical functions, i.e. $y = f(x)$, the procedure involves selecting the range of values for the independent variable $x$ in which you want to plot the function $y$, selecting the spacing between points (i.e. $\Delta x$), evaluating the function at each point $x_i = x_{i-1} + \Delta x$ to find the coordinates $(x_i, y_i)$, and finally plotting the corresponding points. The procedures for plotting the specific types of functions that you will encounter in EE3150 are described next.

## 1 Time domain plots

When plotting a time domain signal $g(t)$ the units must be volts and seconds, since you will be dealing with electrical signals. The procedure to plot time domain signals follows:

1. You will be given a time span ($TS$) for the plot in seconds. The time span determines the interval in which $g(t)$ must be plotted. For example, $TS = 0.001$ means the plot must show the signal from 0 to 1 ms.
2. The number of points that the plot must contain is 2000, as both the oscilloscope and computer display the signals using this number of points. The time resolution $tr$ (spacing between points) of the plot is therefore $tr = TS/2000$ sec.
3. Calculate the coordinates of each point of the plot by evaluating the time function $g(t)$ for each value of $t$ from 0 to $TS$ in increments of $tr$.
4. In summary, the *pseudocode* for the calculation of the coordinates (sec,V) of the 2000 points of a time domain plot is:

```
tr=TS/2000
t=0
for i=1 to 2000
  y=g(t)
  coordinate=(t,y)
  plot coordinate
  t=t+tr
end
```

## 2 Spectral plots

When plotting the magnitude spectrum of a signal it is convenient to express the magnitude of the frequency components in logarithmic units like *decibel-volts* (dBV). This is because the components with small amplitude are masked by components of larger amplitude when the units for the abscisa are volts. The equipment in the laboratory (oscilloscope and computer) displays *one-sided* spectra using dBV.

**In this course all the spectral functions/plots in the prelab and lab report must be computed/presented using $f$ as the frequency variable, with units in dBV and kHz and must be one-sided spectra. DO NOT use $\omega$ $(= 2\pi f)$ as the frequency variable as this will lead to results that cannot be compared to the experimental ones. A table of Fourier transform pairs in terms of $f$ is included in Appendix A**.

The procedure to plot the one-sided magnitude spectrum of $G(f)$ is as follows:

1. You will be given a *frequency span* $(FS)$ in kHz. The frequency span determines the range of frequency you must show in the plot. For example, $FS = 10$ kHz means that the plot must show the spectrum from 0 to 10 kHz.
2. You will be given also a *noise floor level* $(NF)$ in dBV. The noise floor is the minimum level of magnitude that is displayed in the plot. For example, $NF = -40$ dBV means that the plot cannot display values of magnitude lower than -40 dBV and all the values of the spectrum below this level are discarded. The reason for introducing a noise floor level is that in a logarithmic scale, components of amplitude very close to 0 volts would have a magnitude approaching $-\infty$ dBV, and therefore it is necessary to set a lower bound for the plot. You will find that in fact, the experimental results present a noise floor due to the presence of noise in the measurements.
3. The number of points that the plot must contain is 1000, as both the oscilloscope and computer display the signals using this number of points. The frequency resolution $fr$ (spacing between points) of the plot is therefore $fr = FS/1000$ kHz.
4. Calculate the coordinates of each point of the plot by evaluating the *magnitude* of the function $G(f)$ for each value of $f$ from 0 to $FS$ in increments of $fr$. Since a one-sided spectral plot is required, multiply by 2 the magnitude just found, i.e. $2|G(f)|$. This is a value in volts that must be converted to dBV. The conversion from $x$ volts to $y$ dBV is given by:

   $$y = 20\log_{10}\left(\frac{x}{\sqrt{2}}\right)$$

   From this equation, the noise floor in volts is

   $$NF_{volts} = \sqrt{2}\cdot 10^{0.05NF_{dBV}}$$

5. In summary, the *pseudocode* for the calculation of the coordinates (kHz,dBV) of the 1000 points of a one-sided spectral plot is:

```
fr=FS/1000
f=0
for i=1 to 1000
  mag=2|G(f)|
  if mag > NF_volts then
     M=20log10(mag/sqrt(2))
     coordinate=(f,M)
     plot coordinate
  end
  f=f+fr
end
```

Most of the experiments in EE3150 deal with discrete spectra (for example $G(f) = \delta(f - f_o) + \delta(f + f_o)$), for which the above procedure is simplified. Since non-zero frequency components occur only at discrete frequencies, just a few coordinates must be calculated.

# 3 Sample MATLAB code

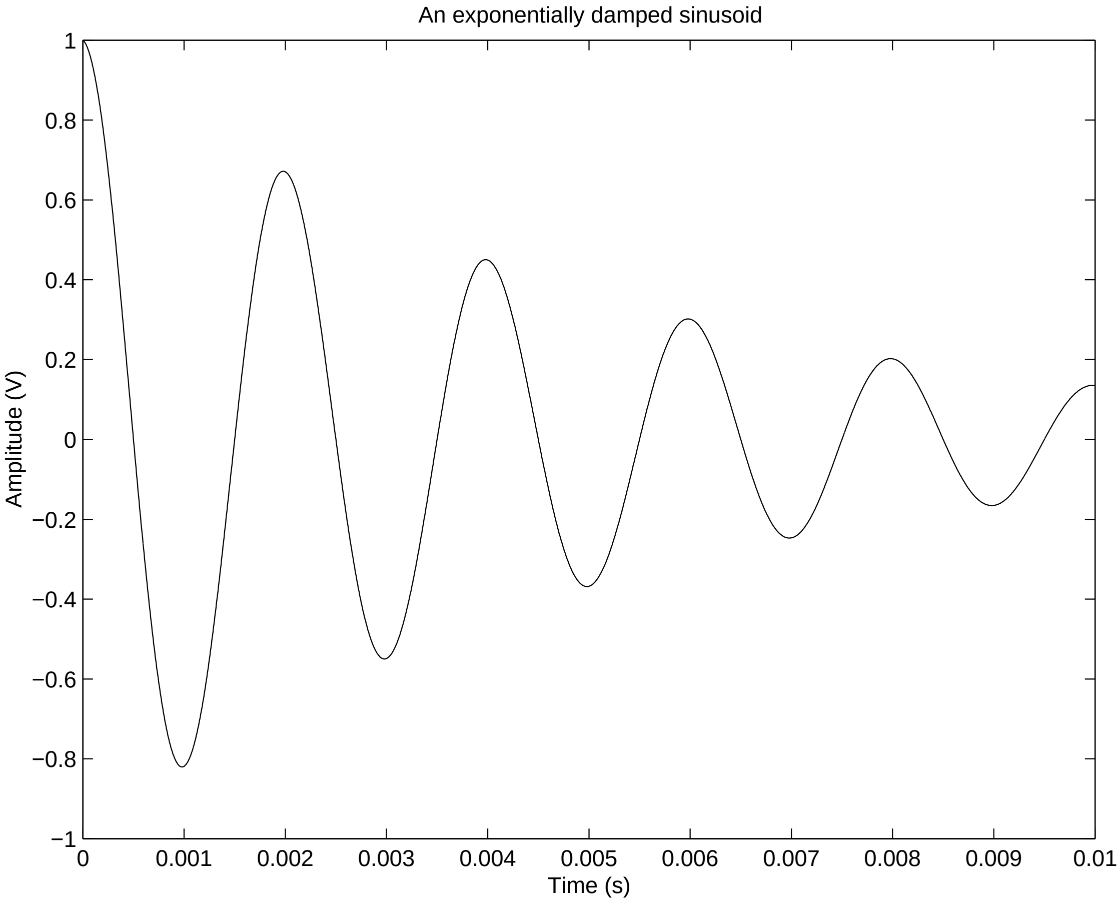
The following MATLAB scripts are provided as a hint only. This DOES NOT imply that you must use MATLAB, or that MATLAB programming is a requisite for the course. They are intended to serve as an example on how the pseudocode in previous sections can be implemented.

## 3.1 A time domain plot

The following script creates a plot of the function $y = \exp(-200t)\cos(2\pi 500t)$ in the interval $0 \le t \le 0.01$.

```
% MATLAB script to plot a time function in EE3150
close all
clear all
TS=0.01;    % time span
tr=TS/2000; % time increment
t=0:tr:TS;  % time vector from 0 to TS in increments of tr

y=exp(-200*t).*cos(2*pi*500*t);    % time function
plot(t,y);
title('An exponentially damped sinusoid');
ylabel('Amplitude (V)');
xlabel('Time (s)');
```

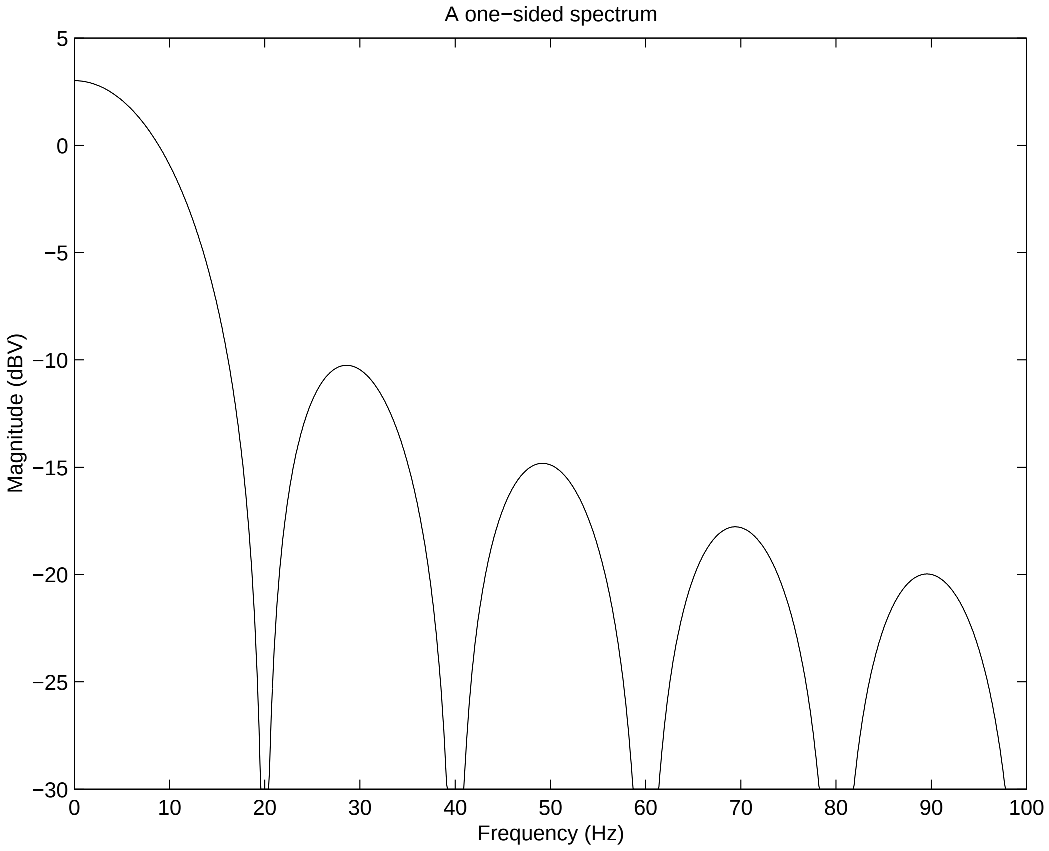


## 3.2 A continuous frequency domain plot

The following script creates a plot of the spectrum $G(f) = \mathrm{sinc}(f/20)$ in the interval $0 \leq f \leq 100$.

```
% MATLAB script to create a one-sided spectral plot
clear;
close all;
FS=100;                     % Frequency span in Hz
NF=-30;                     % Noise floor in dBV
fr=FS/1000;                 % frequency increment
f=0:fr:FS;                  % Freq. vector from 0 to FS in increments of fr
nfv=sqrt(2)*10^(0.05*NF);   % Noise floor in volts

mag=2*abs(sinc(f/20));     % evaluating 2|G(f)|
for i=1:length(f),          % selecting the points above NF
    if mag(i)<nfv,
        M(i)=NF;            % points below NF are made equal to NF
    else
        M(i)=20*log10(mag(i)/sqrt(2));    % conversion to dBV
    end
end
plot(f,M);                  % plotting the coordinates
title('A one-sided spectrum');
ylabel('Magnitude (dBV)');
xlabel('Frequency (Hz)');
```



### 3.3 A discrete frequency domain plot

The following script creates a plot of the spectrum

$$G(f) = 10[\delta(f-3)+\delta(f+3)] + 7[\delta(f-5)+\delta(f+5)] + 4[\delta(f-7)+\delta(f+7)]$$

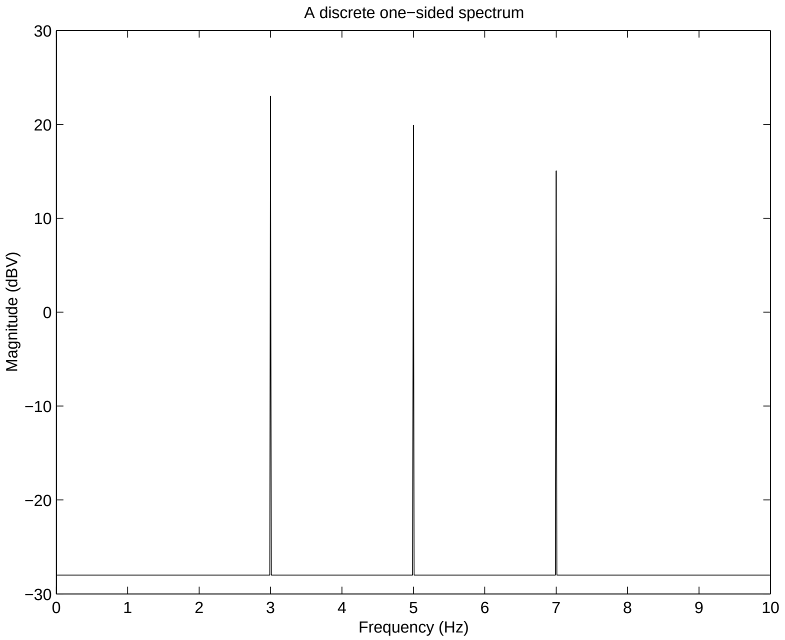
in the interval $0 \leq f \leq 10$.

```
% Matlab script to generate a spectral plot that contains impulses only
clear
close all
FS=10;          % Frequency span in Hz
NF=-28;         % Noise floor in dBV
fr=FS/1000;     % Frequency resolution, spacing between points

% The terms in the positive side of the spectrum are
% 10*d(f-3)+7*d(f-5)+4*d(f-7)

fq=[3 5 7];      % Frequencies where the impulses occur (in Hz)
A=[10 7 4];      % Amplitude of the impulses in volts > NF_volts
                 % Both of these vectors can be extended to any number of impulses

mag=20*log10(2*A/sqrt(2));      % Conversion to dBV and multiply by 2 for a one-sided spectrum
f=[0];          % Initialize frequency vector
M=[NF];         % Initialize magnitude vector
for i=1:length(fq),     % loop to create the necessary points for the plot
    f=[f fq(i)-fr fq(i) fq(i)+fr];
    M=[M NF mag(i) NF];
end
f=[f FS];
M=[M NF];
plot(f,M);      % plotting the coordinates
title('A discrete one-sided spectrum');
ylabel('Magnitude (dBV)');
xlabel('Frequency (Hz)');
```



## 3.4 Plotting the experimental results

During the experiments you record sets of data, a virtual instrument writes the data points of the plots to a text file each time it is executed with the `Save data` switch set to `ON`. You can open these data files with any text editor.

### 3.4.1 Files from TIMEFREQ.VI

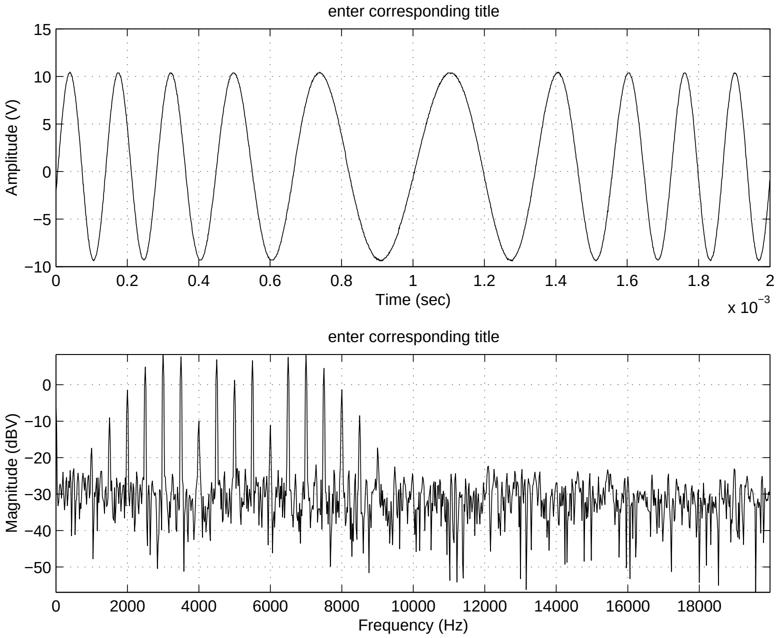
In the data files saved by TIMEFREQ.VI you will find that there are two columns. The first column contains the time domain signal points and the second one contains the spectrum points. The first element of each column is the spacing between points, i.e. $tr$ and $fr$ respectively. The second element of each column is the number of points for each plot, usually 2000 for the time plot and 1000 for the frequency plot. However, the second column is filled with zeroes in the last 1000 entries of the file, pay attention to this fact if you are using Excel to plot the graphs so the frequency scale is correct. The amplitude (in V) or magnitude (in dBV) values are listed from the third element onward.

A MATLAB script that creates the plots from the experimental data files is provided here:

```
% MATLAB script to plot experimental results produced by TIMEFREQ.VI
% Before printing or copying the figure, maximize the plot window
% to separate the label from the title in the middle of the plot
clear
close all
% Substitute 'xformpair.txt' for your file name here:
R=load('xformpair.txt');
subplot(211)
ts=R(1,1); % sampling period
Ns=R(2,1); % number of samples
t=0:ts:(Ns-1)*ts;
plot(t,R(3:Ns+2,1));
title('enter corresponding title') % Enter a title for the time plot
xlabel('Time (sec)')
ylabel('Amplitude (V)')
grid

subplot(212)
fs=R(1,2); % freq increment
Nf=R(2,2); % number of samples
f=0:fs:(Nf-1)*fs;
plot(f,R(3:Nf+2,2));
title('enter corresponding title') % Enter a title for the spectral plot
xlabel('Frequency (Hz)')
ylabel('Magnitude (dBV)')
grid
```

The next figure shows an actual time-frequency plot taken in the lab.



### 3.4.2 Files from RESPONSE.VI

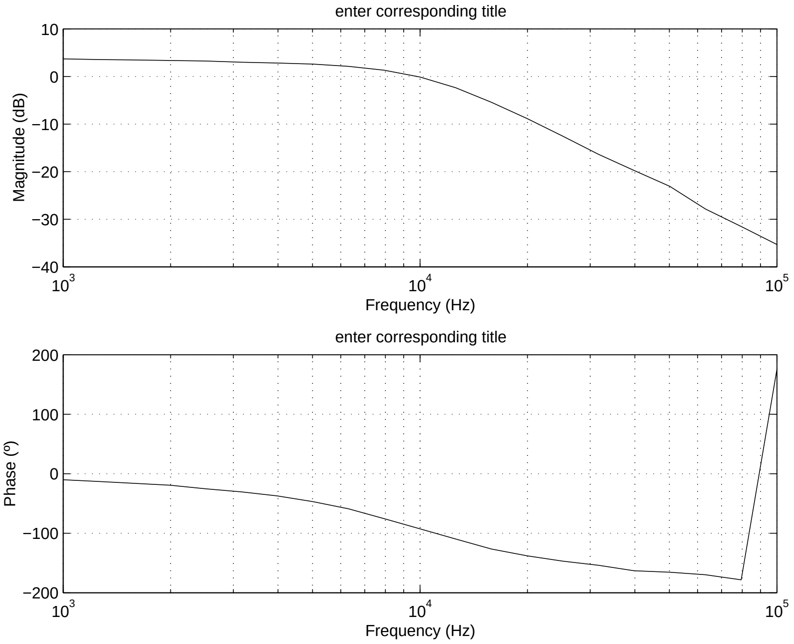
In the data files saved by RESPONSE.VI you will find that there are three columns. The first column contains the frequency values (in Hz) at which the frequency response was measured. The second column contains the magnitude of the response in decibels (dB). The third column contains the values of the phase response in degrees. Pay attention to the fact that the spacing between points is not linear but logarithmic, therefore you must use a semilog scale for the frequency axis.

A MATLAB script that creates the plots from the experimental data files is provided here:

```
% MATLAB script to plot the experimental results produced by RESPONSE.VI
% Before printing or copying the figure, maximize the plot window
% to separate the label from the title in the middle of the plot
clear
close all
% Substitute 'response.txt' for your file name here:
R=load('response.txt');
subplot(211)
semilogx(R(:,1),R(:,2));
title('enter corresponding title') % Enter a title for the magnitude response
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
grid

subplot(212)
semilogx(R(:,1),R(:,3));
title('enter corresponding title') % Enter a title for the phase response
xlabel('Frequency (Hz)')
ylabel('Phase ()')
grid
```

The next figure shows an actual frequency response plot taken in the lab.


enter corresponding title
Magnitude (dB)
10
0
−10
−20
−30
−40
$10^3$
$10^4$
$10^5$
Frequency (Hz)
enter corresponding title
Phase (º)
200
100
0
−100
−200
$10^3$
$10^4$
$10^5$
Frequency (Hz)