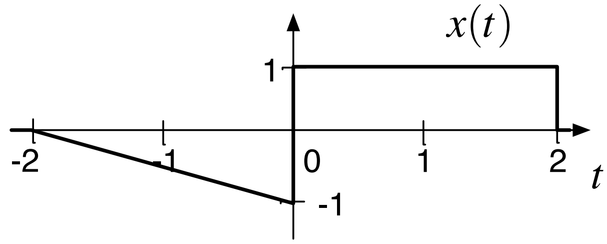
EE102A Winter 2012-13 Pauly
Signal Processing and Linear Systems I

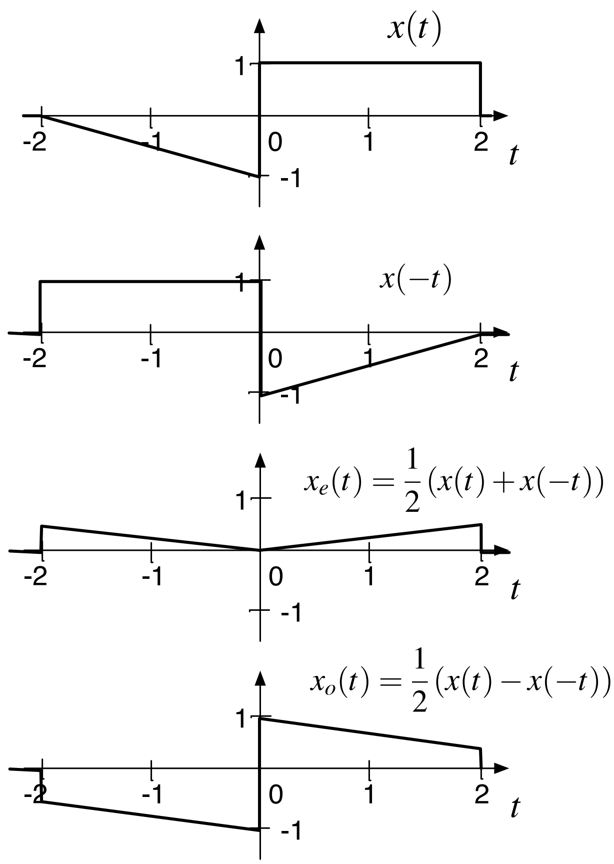
Problem Set #1 Solutions

Due: Wednesday January 16, 2013 at 5 PM.

1. *Even and Odd Symmetry*
   Find the even and odd decomposition of this signal:



Solution:



2. *Even and Odd Signal Components*

   In class we showed that any signal can be written as the sum of an even and odd component,

$$x(t) = x_e(t) + x_o(t).$$

   a) Show that the energy of x(t) is the sum of the energies of the even and odd components

$$\int_{-\infty}^{\infty} x^2(t)\ dt = \int_{-\infty}^{\infty} x_e^2(t)\ dt + \int_{-\infty}^{\infty} x_o^2(t)\ dt.$$

   Solution:

$$\begin{aligned}
\int_{-\infty}^{\infty} x^2(t)\ dt &= \int_{-\infty}^{\infty} (x_e(t) + x_o(t))^2\ dt \\
&= \int_{-\infty}^{\infty} x_e^2(t)dt + 2\int_{-\infty}^{\infty} x_e(t)x_o(t)dt + \int_{-\infty}^{\infty} x_o^2(t)\ dt \\
&= \int_{-\infty}^{\infty} x_e^2(t)dt + \int_{-\infty}^{\infty} x_o^2(t)\ dt
\end{aligned}$$

   where the last equality follows because the $x_e(t)x_o(t)$ is an odd function, and integrates to zero over symmetric limits.

   b) Now, given the same $x_e(t)$ and $x_o(t)$, define

$$x_h(t) = x_e(t) + jx_o(t).$$

   Find the value of $\int_{-\infty}^{\infty} x_h^2(t)\ dt$.

   Solution:

   This is similar, except for the odd component is now imaginary,

$$\begin{aligned}
\int_{-\infty}^{\infty} x_h^2(t)\ dt &= \int_{-\infty}^{\infty} (x_e(t) + jx_o(t))^2\ dt \\
&= \int_{-\infty}^{\infty} x_e^2(t)dt + 2j\int_{-\infty}^{\infty} x_e(t)x_o(t)dt + \int_{-\infty}^{\infty} j^2x_o^2(t)\ dt \\
&= \int_{-\infty}^{\infty} x_e^2(t)dt - \int_{-\infty}^{\infty} x_o^2(t)\ dt.
\end{aligned}$$

   The result is now the difference of the energy in the even and odd components.

3. Find the fundamental periods for the following signals

   (a) $\cos(2\pi t)$

   Solution

   From the definition for the period,

$$\frac{2\pi}{\omega} = \frac{2\pi}{2\pi} = 1$$

(b) $\sin(2t)$ ***Solution***
Here we have

$$\frac{2\pi}{\omega} = \frac{2\pi}{2} = \pi$$

(c) $\cos(2\pi t) + \sin(2t)$
Solution
As we described in class, we are looking for the period T_0 such that both $\cos(2\pi t)$ and $\sin(2t)$ both return to 1. This means that each experiences an integer number of cycles from time $t = 0$ to $t = T_0$,

$$T_0 = n(1) = m\pi$$

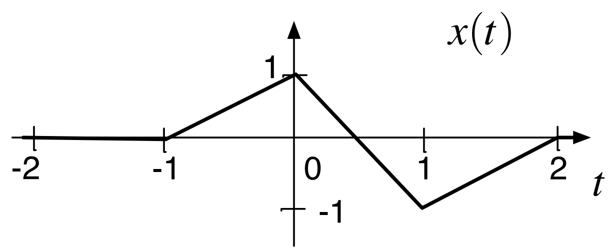
where the periods are 1 and π as we found in (a) and (b). Then to find n and m, we divide both sides by m, and find

$$\frac{n}{m} = \pi$$

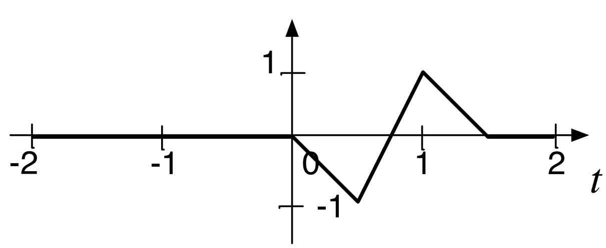
which is impossible, since π is irrational. Hence, this is an aperiodic sequence. This is a little surprising, because sinusoids are the prototypical periodic functions, and all we've done is add two of them up.

(d) $\cos(2\pi t)\sin(2t)$
Solution
The argument is the same here. The product of these two sinusoids is never exactly 1, except for at $t = 0$.

4. *Operations on Signals*
For this problem assume that the signal $x(t)$is as shown below



(a) Write an expression for the following signal in terms of $x(t)$.



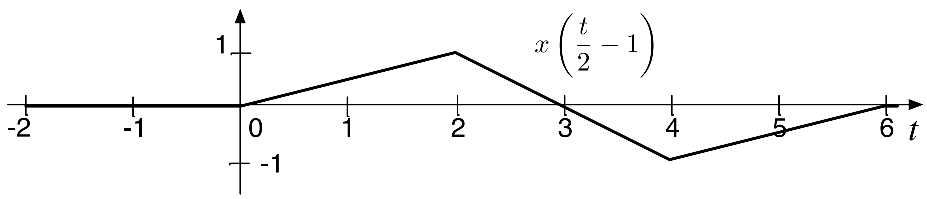
Solution: This is centered at $t = 1$, is time reversed, and compressed by a factor of 2. The answer is then $x(2(1 - t))$

(b) Sketch the signal $x\left(\frac{t}{2}-1\right)$

Solution: This is

$$x\left(\frac{t}{2}-1\right)=x\left(\frac{t-2}{2}\right)$$

which is centered at $t=2$, and is expanded by a factor of 2.



5. If $y(t)$ is an even function, and $y(t-1)$ is also even, is $y(t)$ periodic? If so, what is the fundamental period T_0? Ignore the case where $y(t)$ is constant, and assume that $T=1$ is the smallest shift for which $y(t-T)$ is even.

   Solution:

   Since $y(t)$ is even

   $$y(t)=y(-t)$$

   This also means that

   $$y(t-1)=y(-(t-1))=y(-t+1)$$

   Next, what does it mean for $y(t-1)$ to be even? It means that if we replace t by $-t$, we should get the same value,

   $$y(t-1)=y((-t)-1)=y(-t-1)$$

   If we combine the two expressions for $y(t-1)$

   $$y(-t-1)=y(-t+1)$$

   Since this is true for any t, $y(t)$ must be periodic with a period 2.

6. Assume that the signal $x(t)$ is periodic with period T_0, and that $x(t)$ is odd (*i.e.* $x(t)=-x(-t)$). What is the value of $x(T_0)$?

   Solution

   It first it seems we don't have any information about $x(T_0)$. However, since $x(t)$ is periodic with fundamental period T_0, we know that

   $$x(T_0)=x(nT_0)$$

   for any n. Are than any values of $x(nT_0)$ that are determined by the odd symmetry of $x(t)$?

   The answer is that if $x(t)$ is odd, then $x(0)$ must be zero, since odd symmetry requires

   $$x(0)=-x(0)$$

and this can only be true of $x(0) = 0$.

Since $x(t)$ is periodic, then

$$x(nT_0) = 0$$

for any n, and in particular,

$$x(T_0) = 0.$$

7. *Power and Energy Signals*

   Plot these signals, and classify them as energy or power signals. Support your classification with an explicit calculation or an argument. In each case $-\infty < t < \infty$.

   (a) $x(t) = e^{-2|t|}$

   (d) $x(t) = e^{|-t|}$

   (c) $x(t) = e^{-|t|}\cos(2\pi t)$

   (d) $x(t) = \cos^2(2\pi t)$

   Solution

   a) This is energy, since it you can integrate it directly

   $$E = \int_{-\infty}^{\infty} \left| e^{-2|t|} \right|^2 dt = 2\int_0^{\infty} e^{-4t} dt = -\frac{1}{2}e^{-4t}\Big|_0^{\infty} = \frac{1}{2}$$

   b) This signal grows quickly so lets first check to see if it is even a power signal,

   $$\begin{aligned} \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} e^{|-t|} dt &= \lim_{T\to\infty} \frac{2}{2T}\int_{-T}^{T} e^{t} dt \\ &= \lim_{T\to\infty} \frac{e^T - e^{-T}}{T} \\ &= \infty \end{aligned}$$

   so this grows faster than a power signal, so it is not a power or energy signal.

   c) This signal is bounded by $e^{-|t|}$, which is an energy signal. So $x(t)$ is also an energy signal.

   d) This signal can be written as

   $$\cos^2(2\pi t) = \frac{1}{2}\left(1 - \cos(4\pi t)\right).$$

   The constant and the $\cos(4\pi t)$ are power signals, so this is a power signal. Alternatively, you can simply note that this is a bounded periodic signal, so this is a power signal, as we mentioned in class.

8. *Review of Complex Numbers*

(a) Simplify the following expression

$$(j\sqrt{2})\frac{e^{j(\omega t+\phi)}}{1+j}$$

and leave the result in polar form.

Solution

This is

$$j\sqrt{2}\frac{e^{i(\omega t+\phi)}}{1+j} = e^{j\pi/2}\sqrt{2}\frac{e^{i(\omega t+\phi)}}{\sqrt{2}e^{j\pi/4}} = e^{i(\omega t+\phi+\pi/4)}.$$

(b) Simplify

$$(\cos\omega t + j\sin\omega t)(\cos 2\omega t - j\sin 2\omega t)$$

and leave the result in polar form.

Solution Using complex exponentials, then this is

$$e^{j\omega t}e^{-j2\omega t} = e^{-j\omega t}$$

(c) We can represent a cosine as the sum of two complex exponentials

$$\cos\theta = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right).$$

If we raise a cosine to a power, we generate new frequencies. This will be of use later in the course. Assume that a signal $s(t)$ is generated by

$$s(t) = (\cos\omega t)^3.$$

What frequencies are in $s(t)$? **Solution**

This is

$$\begin{aligned}\left(\frac{1}{2}\left(e^{j\omega t} + e^{-j\omega t}\right)\right)^3 &= \frac{1}{4}\left(e^{j2\omega} + 2 + e^{-j2\omega t}\right)\left(\frac{1}{2}\left(e^{j\omega t} + e^{-j\omega t}\right)\right) \\ &= \frac{1}{8}\left(e^{j3\omega t} + 3e^{j\omega t} + 3e^{-j\omega t} + e^{-j3\omega t}\right)\end{aligned}$$

This means there are three frequencies, $\pm 3\omega$ and $\pm\omega$.

Laboratory 1

For this lab we only need the basic abilities of matlab to create and manipulate vectors, and plot the results. The documentation for matlab exists in many places, including the book that comes with student matlab, the online help (the question mark on the user interface), and the Mathworks web site. We'll assume you are using the online help, but this is also all in your book.

The basic introduction to matlab is given in the "Getting Started" section, which is the first entry under "MATLAB" in the documentation. If this is your first time using matlab, which will be true for many of you, first read the "Matrices and Arrays" section of "Getting Started".

Preparing Your Lab Report Your lab report will consist of a record of the matlab commands, matlab responses, and the plots you generate. To save a copy of your matlab interaction, type

```
>> diary lab1task1.txt
```

This will save everything you type, and all of matlab responses. To add comments, use a % sign at the beginning of the line.

```
>> % This is a comment
```

If you want to stop recording while you try something out, type

```
>> diary off
```

and then, when you want to start recording again,

```
>> diary on
```

Finally, when you are done, you can edit the diary file in your favorite text editor to add comments, or eliminate the text from when you asked for "help", or forgot a semicolon, and printed the entire array. For each task below, the diary file should only contain a few lines.

For the report, submit the (cleaned up) diary file, and the plots that go with it.

Matlab Plotting Basics One of the simplest uses of matlab is as a graphing calculator. We'll go over a simple plotting example. Suppose we want to make plots of the waveforms $w(t) = e^{-t}$, $x(t) = te^{-t}$ and $y(t) = e^{-t} + te^{-t}$ from $t = 0$ to $t = 10$. We want to plot them on both separate and common axes. We follow these steps:

1. Create a time vector:

   ```
   >>t=0:.1:10;
   ```

   This creates a vector, with name `t`, whose elements range from 0 to 10 in intervals of .1. You can type `t` to see the (101) entries of the vector `t`. When you finish the line with a semicolon, Matlab won't print out the results of the operation. You can see what happens if you type

```
>>t=0:.1:10
```

(no semicolon).

2. Now form a vector whose i^{th} component is $w(t_i)$:

```
>>w=exp(-t);
```

This creates a 101-vector `w` whose elements are the values of w at times specified in the time vector `t`. Here we are using a nice feature of Matlab syntax: when a function that normally applies to a real or complex number (such as $\exp$) is applied to a vector, then Matlab applies the function to every entry of the vector.

3. Do the same thing to create x:

```
>>x=t.*exp(-t);
```

The command `.*` specifies element-by-element multiplication of the vectors `t` and `exp(-t)`. (Without the period, Matlab would try to interpret the multiplication as matrix multiplication, and you'll get an error.)

4. Create the vector y:

```
>>y=exp(-t) + t.*exp(-t);
```

Addition of vectors in Matlab is always element by element.

5. Plot w versus time:

```
>>plot(t,w);
```

This makes a plot with t and w on the horizontal and vertical axes, respectively.

6. Label the graph:

```
>>grid on;
>>xlabel('t (sec)');
>>ylabel('w (volts)');
>>title('Sample waveform w vs. time');
```

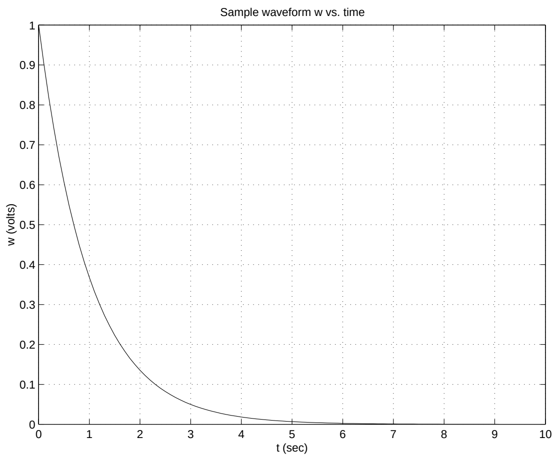
The command `print` will print the current plot, or save it to a file. You can type `help print` to get more information about printing a plot.

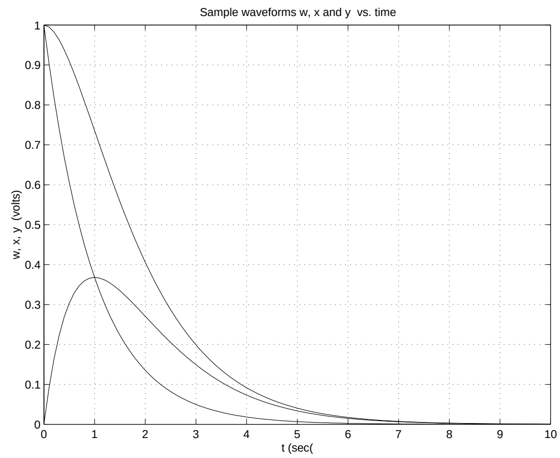
7. Plot w, x and y vs time, on a common set of axes.

```
>> plot(t,w,t,x,t,y);
```

8. Label graph, just as in the previous plot

At this point, we will have the two plots shown below.



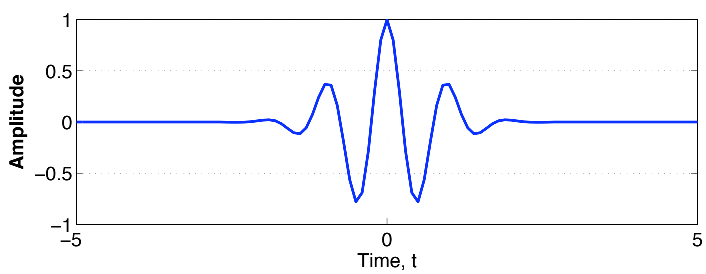


Task 1 Matlab can plot a function verses time, as illustrated above, or plot one function verses another.

A) Plot the waveform

$$x(t) = e^{-t^2} \cos(2\pi t)$$

for $-5 \leq t \leq 5$, at a step size of 0.1. Label the time axis.
Solution:



B) When we analyze the relationship between two different harmonically related frequencies, it is often useful to plot one waveform against the other in a parametric plot. We will do this in the next lab. When the two waveforms are sinusoids, the result is called a *Lissajous curve*. One example is

$$\mathrm{x} = \cos 2\pi \mathrm{At}, \quad \mathrm{y} = \sin 2\pi \mathrm{Bt}$$

To find out more about Lissajous curves, check out

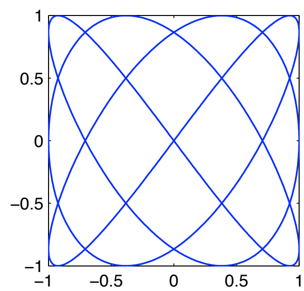
`http://en.wikipedia.org/wiki/Lissajous_curve`

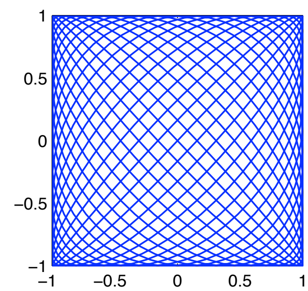
The values of parameters 'A' and 'B' result in curves can have different appearances. Make a plot for $0 \leq t \leq 1$, $A = 3$, $B = 4$ with a step size of 0.001. The Lissajous curve can be created by first defining x and y and then using the command

```
>> plot(x,y);
```

Remember to label the axes and title the figure. Repeat for $A = 19$, $B = 20$
Solution:

These are plotted below.





Task 2 A digitized sound file `guitar_note.mat` is available on the course web site. This is a single note sampled from my daughter Rachel's electric guitar. Load the file with

```
>> load guitar_note
```

after making sure it is in your local directory (use `pwd` to tell where you are, and `cd` to change to another directory).

After loading the file, you will have three matlab variables defined in your workspace. These are `d`, the data itself, `fs`, the sampling rate, and `dt`, the sampling time. You can play the sound in matlab with the

```
>> sound(d,fs)
```

command. This tells matlab to play the waveform `d` on the computer's speakers at a sampling rate of `fs`. If you don't specify `fs` matlab assumes a sampling rate of 8 kHz, which is not correct for this case. The data is actually sampled at 44.1 kHz, which is the sampling rate for CDs.

In this task we are going to approximate the guitar note by a decaying sinusoid, and compare the sounds that the two produce.

A) First, plot the sound waveform as a function of time. Generate a time vector by

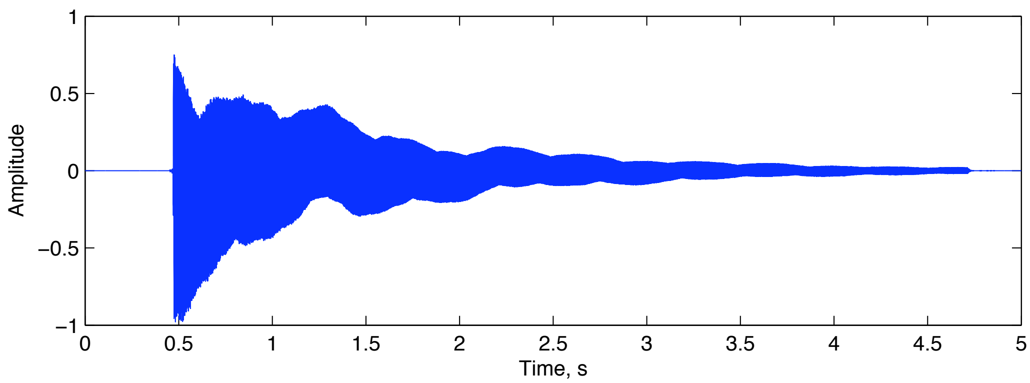
```
>> t = [1:length(d)]*dt;
```

This is time in seconds. Then plot

```
>> plot(t(1:8:end),d(1:8:end));
```

and label the axes. This plots every 8th sample, which saves you time and printer toner. Include this plot in your report. At this scale, all you can really see is the envelope of the waveform (the abrupt start of the waveform, and the slow fading of the sound).
Solution:

```
>> subplot(211)
>> plot(t,d)
>> xlabel('Time, s')
>> ylabel('Amplitude')
```



B) Second, approximate the envelope of the waveform as a delay, followed by a decaying exponential. If `ld = length(d)`, and `ns` is the number of samples in the delay,

```
>> de = [zeros(1,ns) a*exp(sigma*t(ns+1:ld))];
```

Matlab concatenates the two vectors, to make a single, longer vector. Here `sigma` will be negative, since the envelope is decaying. Make an initial estimate of `ns`, `a`, and `sigma`, and check it with

```
>> plot(t,abs(d),t,de);
```

Adjust the parameters until the fit is reasonable (it won't be perfect). Include the plot in your report. What parameters did you choose?
Solution:

```
>> ld=length(d);
>> % Based on the graph, the sound starts at around .45 seconds.
>> ns=round(0.45/dt);

    ns =   3969

>> % at t=.45 the absolute peak is 0.98 (look at the negative peak) and
>> % at t=2.25 the amplitude drops to 0.15. We can calculate sigmal as follows:
```

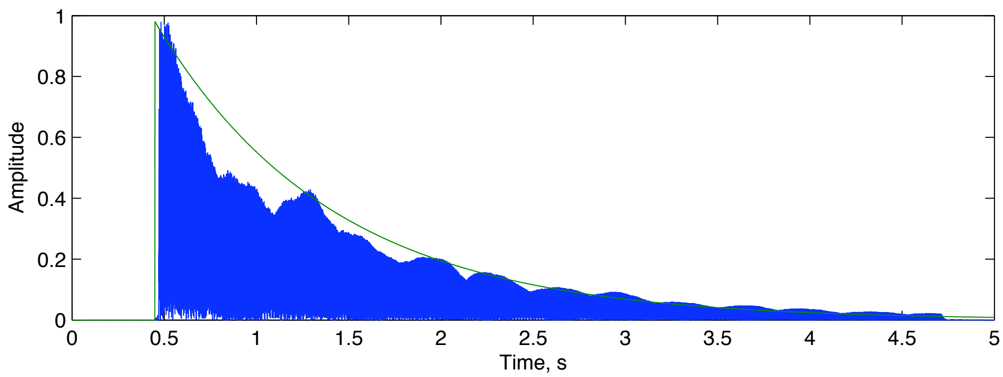
```
>> sigma=log(0.15/0.98) / (2.25 - .45)

    sigma = -1.0427

>> % Now that we have sigma we can find a:
>> a=0.98/exp(.45*sigma)

    a = 1.5668

>> de = [zeros(1,ns) a*exp(sigma*t(ns+1:ld))];
>> subplot(211);
>> plot(t,abs(d),t,de);
>> xlabel('Time, s')
>> ylabel('Amplitude');
```



C) Next, estimate the frequency of the sinusoid. Extract about 1/16th of a second of data from the waveform right around 2 seconds, and call it `ds`. If the starting and ending samples of the segment you want to extract are `n1` and `n2`, the data for the extracted segment is

```
>> ds = d(n1:n2);
```

Create a corresponding time vector, and plot this waveform, label the axes, and turn it in with your report. Estimate the period of this waveform by counting cycles over this interval. Since the waveform is approximately periodic, if we delay the waveform by this time, and subtract it from the non-delayed waveform, we should get something very close to zero. We can use this to refine our estimate of the period. If `tp` is our estimate of the delay, this corresponds to `np = round(tp/dt)` samples. Plot

```
>> lds = length(ds);
>> plot(ds(1:lds-np)-ds(1+np:lds));
```

Adjust the value of `np` until the difference is minimized. The period of the waveform is then `dt*np` seconds. What is the frequency in Hz? If you are interested, you can search the web to figure out what note this corresponds to.

Solution:

```
>> %Estimating the frequency

>> n1=1*fs
>> n2=round((1+1/16)*fs)
>> ds = d(n1:n2);

>> % according to the plot np is about 34 samples
>> fp=fs/34

    fp =  259.4118
```

This is middle ”C”.

D) Simulate the guitar note waveform as an the exponential envelope multiplied by a sinusoid. If `fp` is the frequency you found in C above, then the simulated waveform is

```
>> dsim = de.*sin(2*pi*fp*t);
```

where `de` is the envelope waveform you found in B. Compare the sound of this waveform to the original. Are they the same note? Do they fade at the same rate? If not, go back and check B and C. Plot `dsim` and label the axes, to make sure it looks like the plot from A. Turn this in with your report.
Solution:

```
>> %Reconstruction

>> dsim = de.*sin(2*pi*fp*t);
>> sound(dsim,fs)
```

Although the note is clearly the same, the character of the sounds are quite different. This is due to the fact that there are other frequency components in the guitar note. In a subsequent lab we will estimate these other components, do a better job of approximating the actual waveform, and get a better understanding of why a guitar sounds the way it does.