Stanford University
EE 102A: Signal Processing and Linear Systems I
Summer 2022
Instructor: Ethan M. Liang

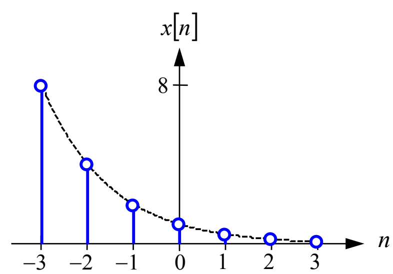
Homework 1 Solutions, due Friday, July 1

Sketching DT Signals

1. **(2 points)** Sketch the following DT real exponential signals versus time n without using a computer or calculator. The range $-3 \le n \le 3$ should be sufficient. You may sketch the envelope of the signal as a continuous curve (two curves if the signal alternates sign), as if n were a CT variable, then draw the DT samples as a "stem" plot.

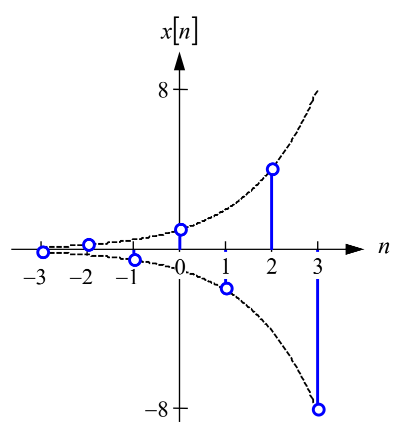
   a. **(1 point)** $x[n] = (1/2)^n$.

   Solution



   b. **(1 point)** $x[n] = (-2)^n$.

   Solution Note that $x[n] = (-2)^n = (-1)^n 2^n$, so the signal alternates sign between $-(2^n)$ and 2^n.

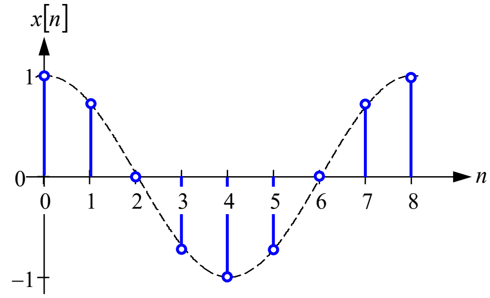


2. **(4 points)** Sketch the following DT sinusoids versus time n without using a computer or calculator. The range $0 \le n \le 8$ should be sufficient. You may sketch the signal as a continuous curve, as if n were a CT variable, then draw the DT samples as a "stem" plot. Recall from lecture that DT imaginary

exponentials or sinusoids with frequencies Ω_0 and $\Omega_0 + k2\pi$ are indistinguishable, since $e^{j(\Omega_0+k2\pi)n} = e^{j\Omega_0 n}e^{j2\pi kn} = e^{j\Omega_0 n}$. If a signal's frequency Ω_0 does not lie in the range $-\pi \le \Omega_0 < \pi$, you will find it easier to add or subtract a multiple of 2π from its frequency Ω_0 so it lies in that range before sketching it.

a. **(2 points)** $x[n] = \cos\left(\frac{\pi}{4}n\right)$.

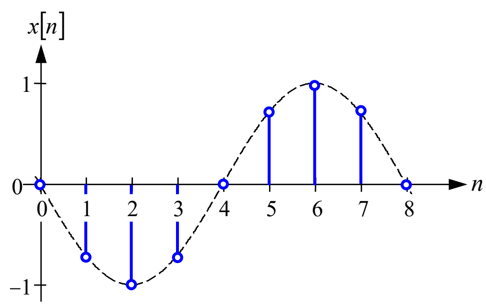
Solution Just one period of this periodic signal is shown.



b. **(2 points)** $x[n] = \sin\left(\frac{15\pi}{4}n\right)$.

Solution Subtract 4π from the frequency: $\frac{15\pi}{4} - 4\pi = -\frac{\pi}{4}$, so $x[n] = \sin\left(-\frac{\pi}{4}n\right) = -\sin\left(\frac{\pi}{4}n\right)$.

Just one period of this periodic signal is shown.



Signal Periodicity

3. **(12 points)** Determine whether each CT signal is periodic and, if so, find its period.

a. **(3 points)** $x(t) = \cos(2\pi t)$.

Solution This is obviously periodic, with period

$$T_0 = \frac{2\pi}{\omega_0} = \frac{2\pi}{2\pi} = 1.$$

b. **(3 points)** $x(t)=\cos^2(2\pi t)$. *Hint*: express the signal as a sum of sinusoids at different frequencies, possibly including zero.

Solution First we express it as a sum of sinusoids:

$$x(t)=\cos^2(2\pi t)=\frac{1}{2}\left[1+\cos(4\pi t)\right].$$

The first term is a zero-frequency sinusoid (constant) and does not affect periodicity. The second term, a sinusoid with frequency $\omega_0=4\pi$, determines the period of the overall signal $x(t)$:

$$T_0=\frac{2\pi}{\omega_0}=\frac{2\pi}{4\pi}=\frac{1}{2}.$$

Squaring the cosine has doubled its frequency and halved its period.

c. **(3 points)** $x(t)=\cos(2\pi t)+\sin(2t)$.

Solution This signal is not periodic.

$\cos(2\pi t)$ is periodic with period $T_1=\frac{2\pi}{2\pi}=1.$

$\sin(2t)$ is periodic with period $T_2=\frac{2\pi}{2}=\pi.$

If the signal $x(t)$ were periodic with period T_0, T_0 would need to be a multiple of both T_1 and T_2 :

$$T_0=mT_1=nT_2,$$

where *m* and *n* are integers. This would require that *m*/*n* satisfy

$$\frac{m}{n}=\frac{T_2}{T_1}=\pi,$$

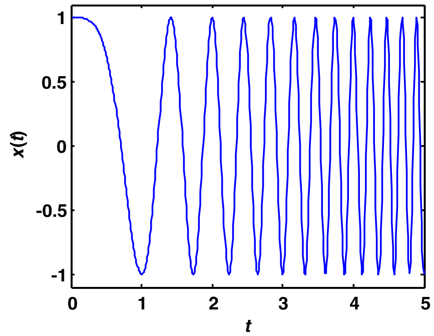
which is impossible, since π is irrational.

d. **(3 points)** $x(t)=\cos(\pi t^2)$. This is a “chirped” sinusoid and can be thought of as a sinusoid whose instantaneous frequency $\omega(t)=d\left(\pi t^2\right)/dt=2\pi t$ increases linearly with time.

Solution This signal is not periodic. If the signal $x(t)$ is periodic, then for all *t* and some *T*, we must have

$$\cos\left(\pi(t+T)^2\right)=\cos\left(\pi\left(t^2+2tT+T^2\right)\right)=\cos\left(\pi t^2\right).$$

This is not satisfied for any *T*. Although you weren’t asked to sketch the signal, a plot is shown.



Energy and Power of Signals

4. **(12 points)** Classify each CT or DT signal as an energy signal ($0 \le E < \infty, P = 0$), a power signal ($0 < P < \infty, E = \infty$) or neither. Support your classification by an explicit calculation or well-reasoned argument. You don't need to state the precise energy or power.

a. **(3 points)** $x(t) = e^{-2|t|},\ -\infty < t < \infty.$

Solution Energy signal. Compute the energy 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. **(3 points)** $x(t) = e^{-t},\ -\infty < t < \infty.$

Solution Neither. It grows too rapidly as $t \to -\infty$. The energy is infinite:

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

Also, the power is infinite:

$$P = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \left| e^{-t} \right|^2 dt = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} e^{-2t} dt = \lim_{T\to\infty} -\frac{1}{4T} e^{-2t} \Big|_{-T}^{T} = \lim_{T\to\infty} \frac{1}{4T}\left(e^{2T} - e^{-2T}\right) = \infty.$$

c. **(3 points)** $x(t) = \cos^2(2\pi t),\ -\infty < t < \infty.$

Solution Power signal. The signal can be written as

$$x(t) = \frac{1}{2}\left[1 + \cos(4\pi t)\right].$$

The constant and the scaled cosine are power signals, and their sum is a power signal.

d. **(3 points)** $x[n] = \left(\frac{1}{2}\right)^{|n|},\ -\infty < n < \infty.$

Solution Energy signal. Compute the energy directly by summing geometric series:

$$E=\sum_{n=-\infty}^{\infty}\left|\left(\frac{1}{2}\right)^{|n|}\right|^2=\sum_{n=-\infty}^{\infty}\left(\frac{1}{4}\right)^{|n|}=\sum_{n=-\infty}^{-1}\left(\frac{1}{4}\right)^{-n}+\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^{n}=\sum_{n=1}^{\infty}\left(\frac{1}{4}\right)^{n}+\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^{n}=-1+2\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^{n}=-1+\frac{2}{1-\frac{1}{4}}=\frac{5}{3}$$

5. **(9 points)** *Even and odd components and signal energy*. Recall that any CT or DT signal can be expressed as the sum of its even and odd components:

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

$$x[n]=x_e[n]+x_o[n]\ .$$

Here we show that the energy of any signal is the sum of the energies of its even and odd parts:

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

$$E=\sum_{n=-\infty}^{\infty}|x[n]|^2=\sum_{n=-\infty}^{\infty}|x_e[n]|^2+\sum_{n=-\infty}^{\infty}|x_o[n]|^2\ .$$

Prove this for the CT case. *Hint*: complex conjugation of a signal preserves its evenness or oddness.

Solution The energy is

$$\begin{aligned}E&=\int_{-\infty}^{\infty}|x(t)|^2dt=\int_{-\infty}^{\infty}\left[x_e(t)+x_o(t)\right]\left[x_e(t)+x_o(t)\right]^*dt\\&=\int_{-\infty}^{\infty}x_e(t)x_e^*(t)dt+\int_{-\infty}^{\infty}x_o(t)x_e^*(t)dt+\int_{-\infty}^{\infty}x_e(t)x_o^*(t)dt+\int_{-\infty}^{\infty}x_o(t)x_o^*(t)dt\\&=\int_{-\infty}^{\infty}|x_e(t)|^2dt+\int_{-\infty}^{\infty}|x_o(t)|^2dt\end{aligned}$$

In the next-to-last line, the second and third integrals vanish because the integrands, $x_o(t)x_e^*(t)$ and $x_e(t)x_o^*(t)$, are odd functions of *t* and the integrals are taken over symmetric intervals. You may recognize these integrals as *inner products* $\langle x_o(t),x_e(t)\rangle$ and $\langle x_e(t),x_o(t)\rangle$, which vanish because the even and odd components are *mutually orthogonal*.

Complex Number Review (not to be turned in)

6. Simplify each expression and express each number in Cartesian (real part plus *j* times imaginary part) and polar (magnitude times an imaginary exponential function of the phase).

a. $(1+2j)(1-3j)$.

Solution Cartesian

$$(1+2j)(1-3j)=1+2j-3j-6j^2=7-j.$$

Polar

$$\sqrt{7^2+(-1)^2}e^{-j\tan^{-1}(1/7)}=\sqrt{50}e^{-j\tan^{-1}(1/7)}.$$

b. $e^{j\pi/2}+e^{j\pi/3}$.

Solution Cartesian

$$\begin{aligned} e^{j\pi/2}+e^{j\pi/3} &= \cos(\pi/2)+j\sin(\pi/2)+\cos(\pi/3)+j\sin(\pi/3) \\ &= 0+j+1/2+j\sqrt{3}/2 \\ &= 1/2+\left(1+\sqrt{3}/2\right)j \end{aligned}.$$

Polar

$$\begin{aligned} \sqrt{(1/2)^2+\left(\left(2+\sqrt{3}\right)/2\right)^2}e^{j\tan^{-1}\left(2+\sqrt{3}\right)} &= \frac{1}{2}\sqrt{1+4+3+4\sqrt{3}}e^{j\tan^{-1}\left(2+\sqrt{3}\right)} \\ &= \sqrt{2+\sqrt{3}}e^{j\tan^{-1}\left(2+\sqrt{3}\right)} \end{aligned}.$$

c. $\sqrt{e^{j\pi}}$.

Solution Since $e^{j\pi}=-1$, we know that the answers are j and $-j$ in Cartesian form and $e^{j\pi/2}$ and $e^{-j\pi/2}$ in polar form. Alternatively, we can use the method of the next problem.

d. Express $\sqrt[3]{e^{j\pi}}$ in polar form.

Solution To start, we write

$$e^{j\pi}=e^{j(\pi+k2\pi)},$$

where *k* is an integer, since we can add any multiple of 2π to the phase of a complex number without changing the number. The cube root is then

$$e^{j\pi/3}=e^{j(\pi/3+k2\pi/3)}.$$

While this expression is true for any *k*, it yields only 3 distinct values, which are given by choosing any three consecutive values of *k*, such as *k* = 0, 1, 2:

$$e^{j(\pi/3+0\times2\pi/3)}=e^{j\pi/3}$$

$$e^{j(\pi/3+1\times2\pi/3)}=e^{j\pi}$$

$$e^{j(\pi/3+2\times2\pi/3)}=e^{j5\pi/3}=e^{-j\pi/3}.$$

We can use this approach to solve for the *N* different N^{th} roots of any complex number.

e. Find an expression for $\cos^3\theta$ in terms of $\cos\theta$ and $\cos 3\theta$.

Solution We can write

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

Then

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

f. Simplify the expression $j\sqrt{2}\dfrac{e^{j(\omega t+\varphi)}}{1+j}$ and leave the result in polar form.

Solution

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

g. 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

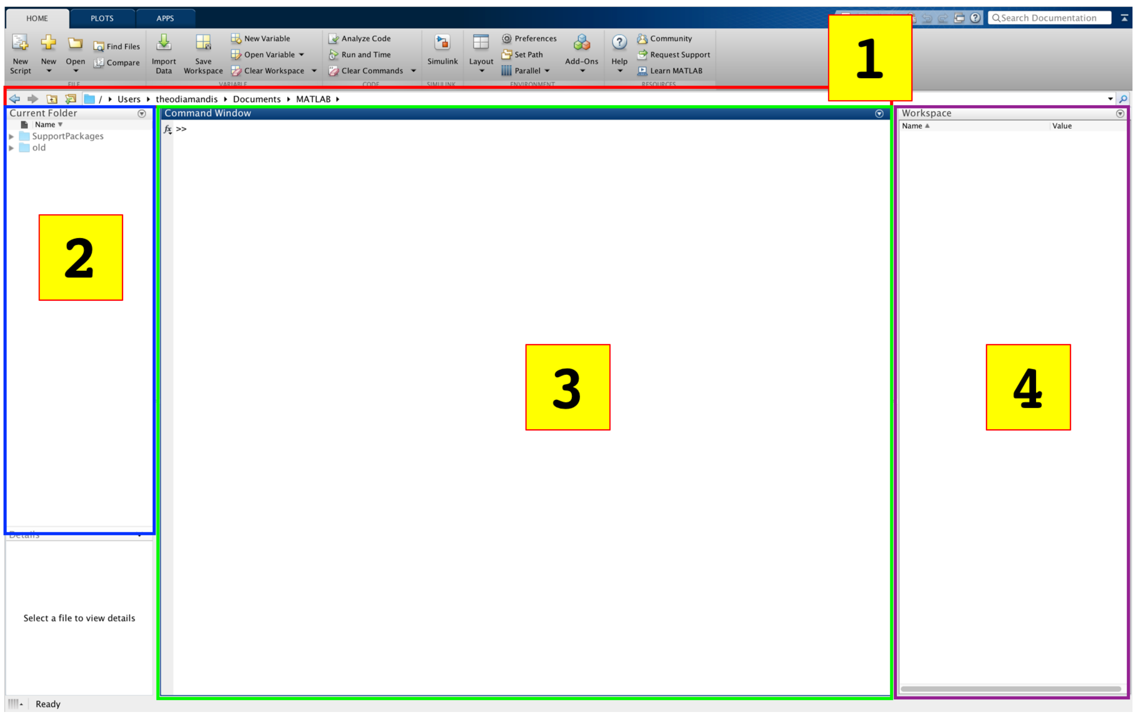
$$(\cos\omega t + j\sin\omega t)(\cos 2\omega t - j\sin 2\omega t) = e^{j\omega t}e^{-j2\omega t} = e^{-j\omega t}.$$

Laboratory 1

In this lab, we'll create, manipulate, and plot vectors in MATLAB. Before beginning, please read the "Matrices and Arrays" section of the "Getting Started" guide https://www.mathworks.com/help/matlab/getting-started-with-matlab.html. For future reference, documentation and examples can be found at https://www.mathworks.com/help/.

1. (11 points) Getting Started

When you open MATLAB, you should see a screen similar to the screenshot below.



Region 1 (red) shows you the current folder you are in. Here, we are currently working in `Users/theodiamandis/Documents/MATLAB` (this will differ in a different operating system).

Region 2 (blue) shows you the contents of this folder.

Region 3 (green) is the command window, which allows you to execute short bits of code. Try typing in 2 + 2.

Region 4 (purple) shows you the variables you current have defined. We'll come back to this in a moment.

a. Your first script: Plotting Basics

Your lab report will consist of the scripts (MATLAB programs) you write, the figures you generate, and any output. To start, navigate to the folder you wish to keep your work in using the address bar. You can navigate by typing in the path ("tab" shows autocomplete suggestions) or by clicking on your Documents folder in the address bar and navigating via the current folder interface. We recommend using a different folder for each homework assignment to keep things tidy as complexity increases. Then hit the new script button, in the top left. The new file will open in the middle of your screen.

Now, at the top of your script, write a comment with your name, the homework, and the problem, as done below. The % symbol makes a comment in MATLAB. Save your script.

```
% Your Name
% EE102A HW1
% Problem 1
clear all; close all;
```

The last line clears all saved variables (none yet) and closes any figures that MATLAB produced. You should always include this at the beginning of a program. Now, let's create a section for Part (a) of lab Problem 1 with a `%%`, as shown below. A section allows us to run part of a script without running the entire thing.

```
%% 1a
```

Finally, let's plot a discrete time sinusoid $x[n] = \sin(\Omega_0 n)$, $\Omega_0 = -\frac{\pi}{4}$. First we have to define the independent variable axis in our script:

```
n = 0:10;
```

This creates a vector `[0 1 … 10]` and stores it in the variable **n**. Run the program (big green play button), then try typing **n** in the command window (Region 3).

```
>> n
```

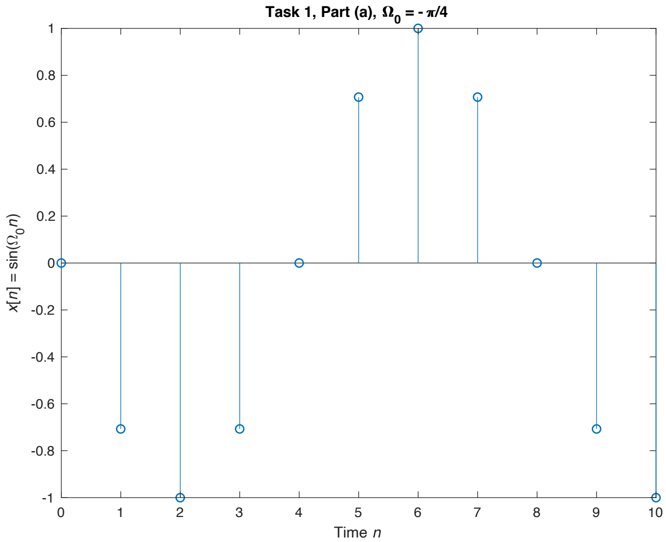
Now, let's define the sinusoid in terms of n and plot it.

```
Omega0 = -pi/4;
xn = sin(Omega0 * n);
figure; stem(n, xn);
```

The first line creates a new vector **xn** by evaluating the expression for each value of n. The second line creates a stem plot of **xn** with respect to **n**. We use figure to avoid accidentally overriding any previously made plot (not necessary here, but always good practice). Now, label the graph and its axes:

```
xlabel('Time \itn');
ylabel('\itx\rm[\itn\rm] = sin(\Omega_{0}\itn\rm)');
title('Task 1, Part (a), \Omega_{0} = -\pi/4');
```

The command **\it** makes the text that follows it *italics* and **\rm** returns the text to normal. Note that using italics and Greek letters is optional. Your graph should look something like the one below:



When you include this graph in your report, use the **copy figure** option under the Edit menu. **Do not use a screenshot**.

This plot shows the signal, but we can improve on it. Let's explore some additional plotting functionality. First, we can superimpose a curve showing what **xn** would look like if **n** were a continuous variable. We'll call this continuous variable **t**. (You did something similar in sketching discrete time signals in analytical Problems 1 and 2 above.) All signals in MATLAB must be in discrete time, since truly describing continuous-time signals would require an infinite amount of information. However, we can often approximate continuous-time signals well by using a small step size between time points. When we plot this signal, it will look like a CT signal.

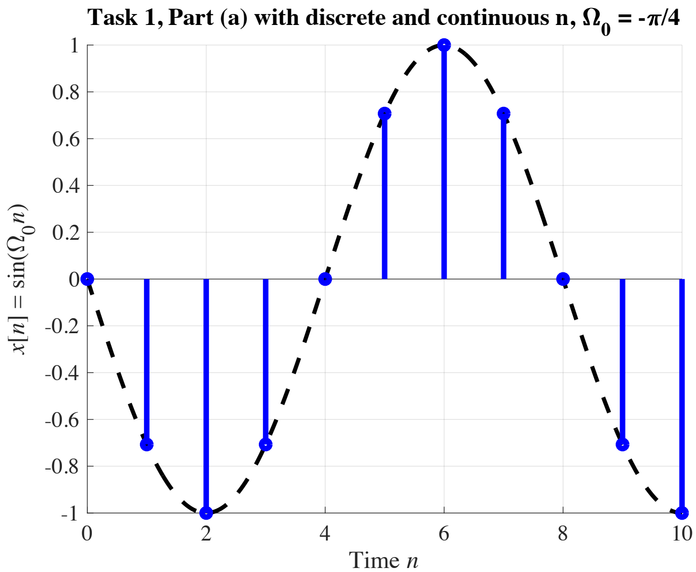
```
t = 0:.01:10;
xn_ct = sin(Omega0 * t);
```

This creates a time vector from 0 to 10 with a step size of 0.01 between points. We also could have used **t = linspace(0,10,1001)** for the same result. Now let's make a better plot.

```
figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(t, xn_ct,'k--', 'LineWidth', 3);
stem(n, xn, 'b', 'LineWidth', 4);
xlabel('Time \itn');
ylabel('\itx\rm[\itn\rm] = sin(\Omega_{0}\itn\rm)');
title('Task 1, Part (a) with discrete and continuous n, \Omega_{0} = -
\pi/4')
grid on;
```

Again, we create a new figure. **hold on** tells MATLAB that we want multiple signals on the same plot. Next, we tweak a lot of variables to make the plot look a little nicer. **gca** returns the current plot, which we use to make the axes, labels, and titles easier to read. The **'k--'** setting makes the line black and dotted.

Similarly `'b'` sets the stem plot to be blue. `'LineWidth'` is fairly self-explanatory, but note that the value follows it. Finally, `grid on` displays the grid lines on the plot. Feel free to check out other customizations in the documentation. The final result is below.



b. Your first script: indistinguishable sinusoids:

Create a new section and then define $y[n] = \sin(\Omega_1 n)$, $\Omega_1 = \Omega_0 + 2\pi = \frac{7\pi}{4}$, with the same n used in Part (a). Also define the same signal with a continuous n, as we did in Part (a). Call this **yn_ct**. Note that you can use the variables you already defined in Part (a) in this section, and you can use the "Run Section" button to only run this section of your script.

(2 points) Solution:

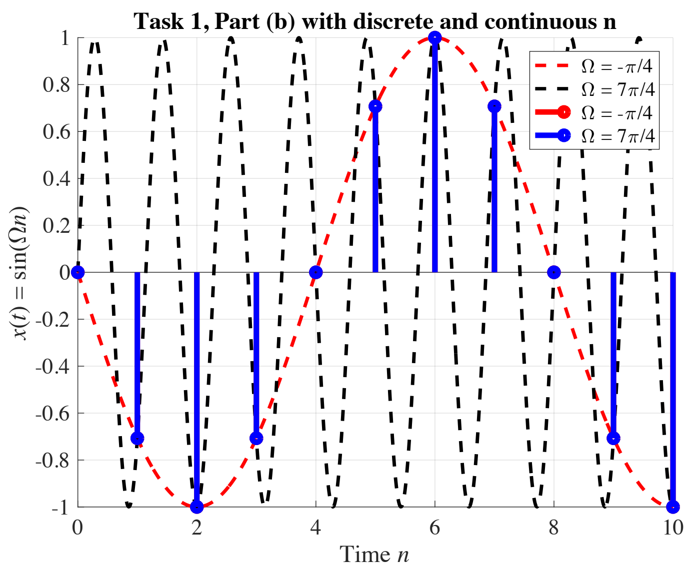
```
Omega1 = Omega0 + 2*pi;
yn = sin(Omega1 * n);
yn_ct = sin(Omega1 * t);
```

In lecture, we learned that CT sinusoids at different values of frequency ω are distinguishable, but not all DT sinusoids at different values of frequency Ω are distinguishable. We can see this by plotting the continuous-time versions of $x[n]$ and $y[n]$ on the same plot:

```
figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(t, xn_ct,'r--', 'LineWidth', 2.5, 'DisplayName', '\Omega = -
\pi/4');
plot(t, yn_ct,'k--', 'LineWidth', 2.5, 'DisplayName', '\Omega =
7\pi/4');
stem(n, xn, 'r', 'LineWidth', 4, 'DisplayName', '\Omega = -\pi/4');
stem(n, yn, 'b', 'LineWidth', 4, 'DisplayName', '\Omega = 7\pi/4');
xlabel('Time \itn');
```

```
ylabel('\itx\rm(\itt\rm) = sin(\Omega\itn\rm)');
title('Task 1, Part (b) with discrete and continuous n')
legend('show');
grid on;
```

Here, we use the `'DisplayName'` parameter to name each signal, so later we can create a legend. The final plot appears below.



We see that the continuous-time versions of these signal are very different, but their discrete time versions are identical. In fact, discrete sinusoids $\sin(\Omega_0 n)$ and $\sin(\Omega_1 n)$, $\Omega_1 = \Omega_0 + 2\pi k$ for any integer k are indistinguishable from each other. Later in the course, we will learn about sampling CT signals and the minimum rate at which we need to sample a CT signal to unambiguously characterize it.

c. Your first script: more vector manipulation

Let's now plot the function $w[n] = x^2[n]$. Start by defining it:

```
%% 1c
wn = xn.*xn;
```

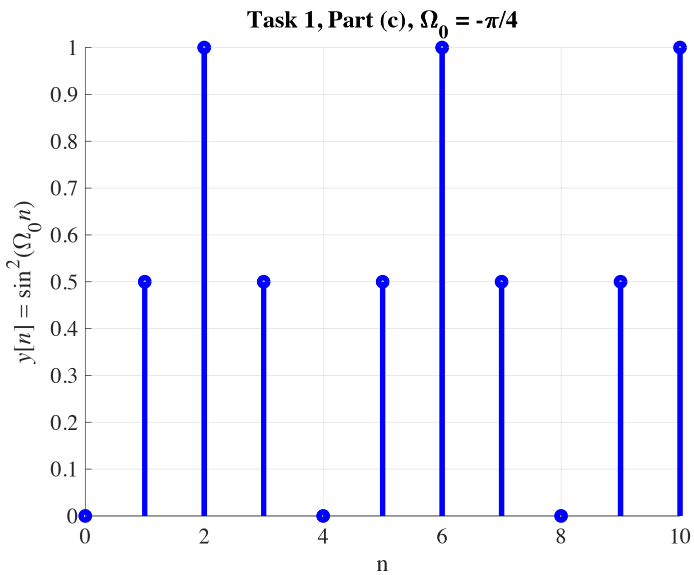
We use `.*` for element-wise multiplication. On its own, `*` performs matrix multiplication when the arguments are both vectors/matrices. Again, note that even though we are working in a new section, we can use previously defined variables, and you can use the "Run Section" button to only run this part of your script.

Now, plot this signal with a stem plot. Be sure to make it look nice and readable. Is the signal periodic?

(3 points) Solution:

```
wn = xn.*xn;

figure; hold on;
set(gca,'FontName','times','FontSize',16);
stem(n, wn, 'b', 'LineWidth', 4);
title('Task 1, Part (c), \Omega_{0} = -\pi/4');
xlabel('n'); ylabel('\ity\rm[\itn\rm] = sin^2(\Omega_0\itn\rm)');
grid on;
```



d. Your first script: the imaginary exponential

Now consider the imaginary exponential

$$e^{j\theta} = \cos\theta + j\sin\theta$$

In MATLAB, define $z(t) = e^{-t+j2\pi t}$ where $t \in [0,5]$ is a continuous-time variable. The exponential function in MATLAB is **exp()**. We can extract the real and imaginary components of $z(t)$ with the code below:

```
re_zt = real(zt);
im_zt = imag(zt);
```

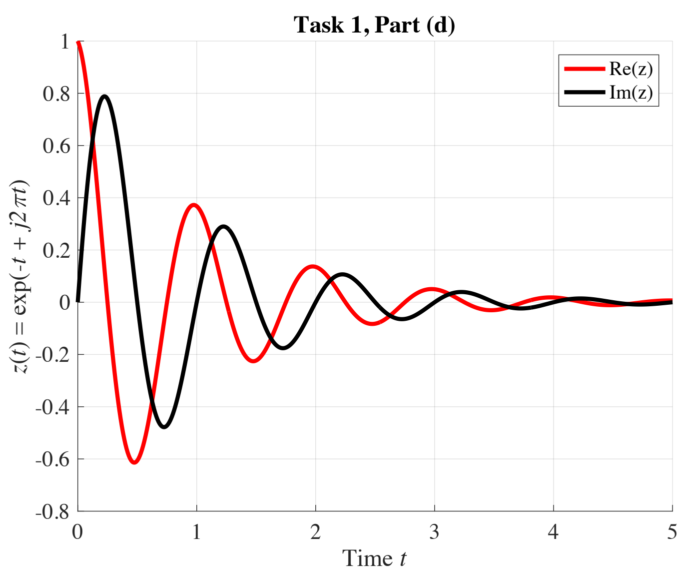
Plot the real and imaginary components of $z(t)$ over time on the same plot. Don't forget to include a legend on your plot. Comment on what you see.

(4 points) Solution:

```
%% 1d
t = 0:.01:5;
zt = exp(-t + j*2*pi*t);

re_zt = real(zt);
im_zt = imag(zt);

figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(t, re_zt,'r', 'LineWidth', 3, 'DisplayName', 'Re(z)');
plot(t, im_zt,'k', 'LineWidth', 3, 'DisplayName', 'Im(z)');
xlabel('Time \itt');
ylabel('\itz\rm(\itt\rm) = exp(\it-t + j2\pit\rm)');
title('Task 1, Part (d)')
legend('show');
grid on;
```



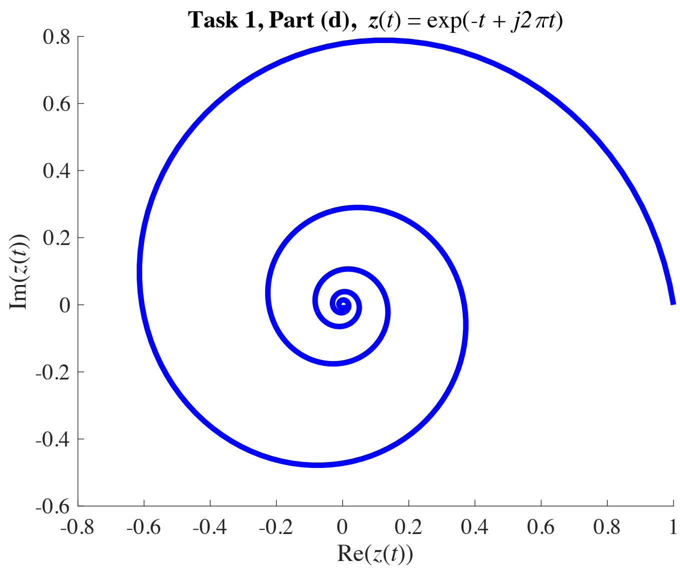
Decaying sinusoids 90 degrees out of phase.

Next we are going to plot $z(t)$ in the complex plane (a phasor plot), as shown below.

```
figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(re_zt, im_zt,'b', 'LineWidth', 4);
xlabel('Re(\itz\rm(\itt\rm))');
ylabel('Im(\itz\rm(\itt\rm))');
title('Task 1, Part (d),  \itz\rm(\itt\rm) = exp(-\itt + j2\pit\rm)')
```

Comment on what you see. Explain it in terms of the behavior of the real and imaginary exponentials. Be sure to explain why we see a spiral instead of a circle.

(2 points) Solution:



The imaginary exponential is a circle in the complex plane as theta varies. $z(t)$ is an imaginary exponential multiplied by a decaying exponential. We get a spiral in the complex plane due to the decay.

2. (12 points) Guitar Note (tuning parameters to mimic a guitar)

A digitized sound file **`guitar_note.mat`** is available in **`Homework_1_MATLAB_Files.zip`** in the Homework Assignments folder on the course Canvas website. Place this file in the directory you're working in (same directory as the script in which you'll use it). Create a new script, and in the header load the guitar note with **`load`**.

```
% Your Name
% EE102A HW1
% Problem 2
clear all; close all;
load guitar_note.mat;
```

After loading the file, you will have three MATLAB variables defined in your workspace. These are **`note`** the note data itself, **`fs`**, the sampling rate, and **`dt`**, the time between samples $\left(fs = \frac{1}{dt}\right)$. You can play the note by typing

```
>> sound(note,fs)
```

into the command window. This tells MATLAB to play the waveform **note** on the computer's speakers at a sampling rate of **fs**. If you don't specify **fs**, MATLAB will assume a default sampling rate which may or may not be correct.

We will tune parameters to approximate the guitar note with an exponentially decaying sinusoid, and then we will compare the two sounds produced. Specifically, our generated guitar note will be of the form

$$\alpha e^{\sigma t} \sin(2\pi f_{note} t)\, u(t - t_0).$$

Since $\sigma < 0$, this is an exponentially decaying sinusoid that is nonzero only after time t_0.

a. Approximating the envelope

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

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

This is time in seconds. Then plot

```
figure; plot(t(1:8:end),note(1:8:end));
xlabel('time (s)');
ylabel('Amplitude');
title('Guitar note waveform');
```

This plots every 8th sample. 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). You do not have to turn in this plot.

Now, we will approximate the envelope of the waveform as a decaying exponential that is nonzero only once we reach time **t0**. We can do this by using a step function $u(t - t_0)$. In MATLAB, this is **double(t>=t0)** for now (in the future, we'll learn how to create functions in MATLAB). Thus, we approximate the envelope as

$$\text{envelope}(t) = \alpha e^{\sigma t}\, u(t - t_0).$$

In MATLAB, this is

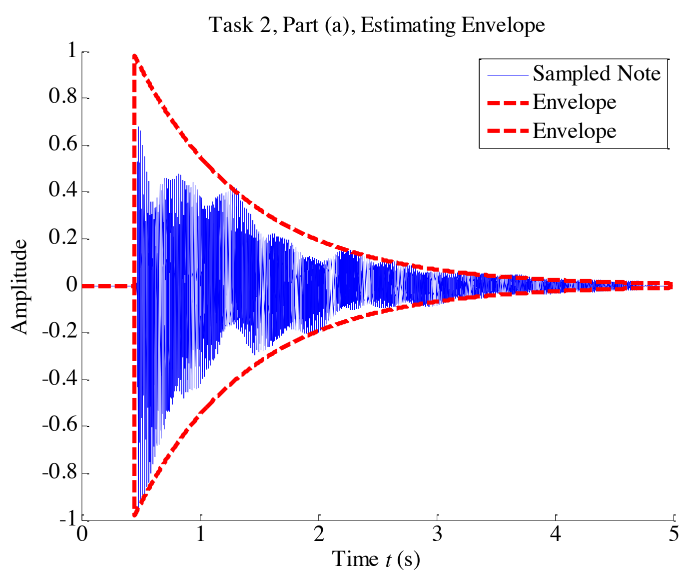
```
envelope = alpha*exp(sigma*t).*double(t>=t0);
```

Make an initial estimate of alpha, sigma, and t0, and check it by plotting **note**, **envelope**, and **-envelope** on the same graph. Adjust parameters until the fit is reasonable (it won't be perfect). Make sure to include this figure and your chosen parameters in your report.

```
(4 points) Solution:
t = [1:length(note)]*dt;


t0 = .45;
alpha = 1.5668;
sigma = -1.0427;
envelope = alpha*exp(sigma*t).*double(t>=t0);

figure; hold on;
set(gca,'FontName','times','FontSize',14);
plot(t(1:8:end),note(1:8:end), 'DisplayName', 'Sampled Note');
plot(t(1:8:end),envelope(1:8:end), 'r--', 'LineWidth', 3,
'DisplayName', 'Envelope');
plot(t(1:8:end),-envelope(1:8:end), 'r--', 'LineWidth', 3,
'DisplayName', 'Envelope');
xlabel('Time \itt \rm(s)');
ylabel('Amplitude');
title('Task 2, Part (a), Estimating Envelope');
legend('show');
```



b. Estimating the frequency

Next, estimate the frequency of the note, f_{note}. We extract about 1/16th of a second of data from the waveform starting at 1 second and call it **note_sample**. We first get the index of the sample of interest by multiplying the sampling rate by the desired start and end times. Then we use this index vector to extract the desired parts of the time and note vectors.

```
index_sample = 1*fs:round((1+1/16)*fs);
t_sample = t(index_sample);
note_sample = note(index_sample);
```

Plot this waveform, label the axes, and turn it in with your report. Make sure you are plotting over the correct time interval.

Estimate the frequency of this waveform by counting cycles over this interval.

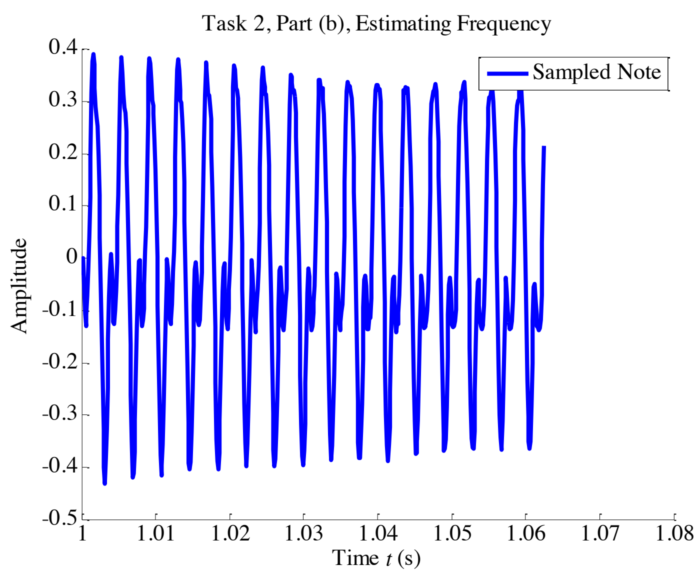
$$f_{note} = \frac{\text{number of cyles}}{\text{time}}$$

In this case, time = $1/16$. Report your value. Look up the frequency that this note corresponds to here: https://en.wikipedia.org/wiki/Piano_key_frequencies.

(4 points) Solution:

```
figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(t_sample,note_sample,'LineWidth',3,'DisplayName','Sampled Note');
xlabel('Time \itt \rm(s)');
ylabel('Amplitude');
title('Task 2, Part (b), Estimating Frequency');
legend('show');

f_note = 16.5*16. This corresponds to middle C.
```



c. The simulated note

Simulate the guitar note waveform as an exponential envelope multiplied by a sinusoid. If **`f_note`** is the frequency you found in Part (b) above, then the simulated waveform is

```
simulated_note = envelope.*sin(2*pi*f_note*t);
```

where **envelope** is the envelope waveform you found in Part (a). 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 Parts (a) and (b).

Finally, plot your simulated note and label the axes to make sure it looks something like the plot from Part (a). Plot every 16th sample (instead of every sample).

(4 points) Solution:

```
%% c
simulated_note = envelope.*sin(2*pi*f_note*t);
figure; hold on;
set(gca,'FontName','times','FontSize',16);
plot(t(1:16:end),simulated_note(1:16:end), 'k', 'LineWidth', .5,
'DisplayName', 'Simluated Note');
xlabel('Time \itt \rm(s)');
ylabel('Amplitude');
title('Task 2, Part (c) Simulating Note');
legend('show');
```