Laboratory # 2

Complex Numbers, Complex Exponentials, and Sinusoids

This lab focuses on the use of complex numbers and complex exponentials. Complex numbers are crucial to an analytic understanding of signals and systems, so this laboratory will give you experience working with them. Note that appendix A of the text has a very complete discussion of complex numbers. You should read this appendix before beginning this laboratory and refer back to it as needed.

1. [21] Perform the following complex arithmetic by hand. Make sure you show your work. Also, use MATLAB to check your answers. For each problem, plot the complex numbers used in the calculation on the same plot using the function `ax_plot.m` (located on the course webpage). `ax_plot.m` plots multiple (x, y) pairs with axis lines and the unit circle. The first parameter is a vector of x components (the real parts of your complex numbers) and the second is a vector of y components (the imaginary parts). (Note: you can use `subplot` to put the plots from several of these problems into a single figure, but make sure you label the plots appropriately).

   (a) Write $3 + j6$ in polar form.

   (b) Write $2\angle\frac{3\pi}{4}$ in rectangular form.

   (c) Add $1 - j2$ and $2\angle\pi/3$.

   (d) Multiply $1\angle - \frac{\pi}{6}$ by $-5 - 3j$.

   (e) Take the reciprocal of $4 + j4$.

   (f) Take the complex conjugate of $2 - j$.

   (g) Take the complex conjugate of $3\angle\frac{2\pi}{3}$.

2. [12] Consider the general complex number $z = a + jb$.

   (a) Compute z^2.

   (b) Compute zz^*.

   (c) What do you notice about your answers when z is either purely real or purely imaginary (i.e., either a or b is zero)?

   (d) Does the same hold when both a and b are nonzero?

3. [10] Complex numbers may arise when taking the roots of polynomials. Recall that the roots of a parabola, $y = ax^2 + bx + c$ are given by the quadratic formula,

$$roots = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Also recall that we can factor $y = ax^2 + bx + c$ into the form $y = (x - r_1)(x - r_2)$ where r_1 and r_2 are roots of the polynomial.

The MATLAB function `parabola_plot.m`, which is available on the course webpage, plots parabolas of the form $y = x^2 + bx + c$ in one subplot and their roots on the complex plane in another subplot. The function takes two parameters. The first parameter is the b coefficient, and the second parameter is the c coefficient. If the parameters are vectors, the function will plot several parabolas on the same graph.

(a) Use `parabola_plot.m` to find coefficients for which
   i. The roots are real and distinct
   ii. The roots are repeated
   iii. The roots are complex

(b) What do you notice about the complex roots? (Hint: is y ever complex?)

(c) Set b equal to one and look at values of c ranging from negative to positive. What happens to the roots on the complex plane as c increases?

4. [8] Execute the following commands, and measure the frequency (in Hz) and phase (in radians, relative to cosine phase) of the resulting sinusoids by inspection. Make sure you note where zero occurs on the time axis. Print the two signals.

   (a) `plot(linspace(-.5,2,1000),cos(linspace(-10,32,1000)));`

   (b) `plot(linspace(-2,1,1000),cos(linspace(0,100,1000)));`

5. [17] The following is an examination of phasor addition.

   (a) Generate three sinusoids, each with a frequency of 300 Hz. Use `0:1/8192:0.01` as your time axis. The sinusoids should have the following amplitudes and phases:
      i. The first should have an amplitude of 2.5 and a phase shift of $\frac{\pi}{5}$.
      ii. The second should have an amplitude of 0.75 and a phase shift of $-\frac{\pi}{8}$.
      iii. The third should have an amplitude of 1.5 and a phase shift of $\frac{\pi}{2}$.

   (b) Generate a fourth sinusoid by summing the three sinusoids you've just generated.

   (c) Plot all four sinusoids on the same figure (use `subplot`).

   (d) Measure the phase shift, amplitude, and frequency of the fourth sinusoid by inspection, annotating your graph sufficiently.

   (e) Now calculate what the amplitude and phase shift should be using phasor addition. Do you get the same value?

6. [16] The complex exponential signal, $e^{j(\omega t+\phi)}$, is extremely important to the study of signals and systems. Euler's identity establishes a close relationship between complex exponential signals and sinusoids. In MATLAB, we create complex exponentials roughly the same way as sinusoids.

   (a) Generate two complex exponentials, one with a frequency of 3 Hz and another with a frequency of -3 Hz. Both should have zero phase shift, an amplitude of 1, and should be generated over the time range from $t = 0$ to $t = 1$. Use a sampling frequency of 8192 Hz.

   (b) Take the sum and difference of the two complex exponential signals.

   (c) In a figure with four subplots, plot the real and imaginary parts of both the sum signal and the difference signal.

   (d) What do you notice about the resulting plots? Is this what you would expect? Why might the plots turn out like this? (Hint: notice the relative amplitudes of the real and imaginary parts of the two signals.)

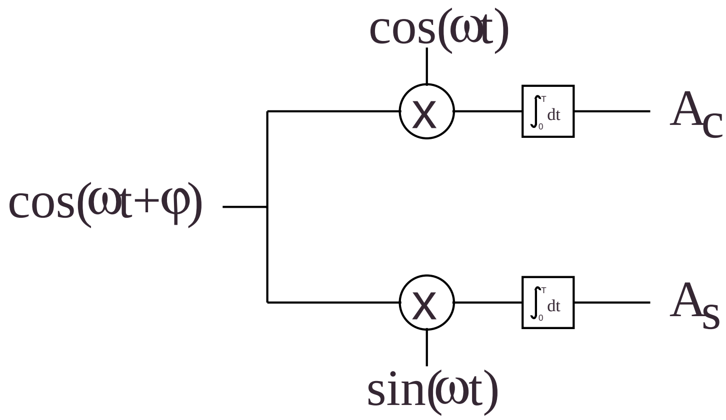

Figure 2.1: Phase detection system

7. [16] Consider the following problem. We have a signal composed of a sinusoid with known frequency corrupted by *additive noise* (that is, a noise signal has been added to the sinusoid) , and we would like to know the phase of the sinusoid with respect to some reference. For small amounts of noise, this is a trivial problem that can be solved by inspection. When the amount of noise becomes significant, however, this becomes a much more difficult problem. One way to approach this problem is with the system shown in Figure 1.

   In this system, we multiply the signal by a sine-phase sinusoid and a cosine-phase sinusoid and integrate both of the resulting product signals from zero to some time T. Such multiply-then-integrate procedures are known as *correlations*. The values produced by these two correlators tell us how similar the original signal is to the sine wave and the cosine wave, respectively.

   The Matlab m-file `phase_detect.m` (available on the course web page) implements this system. The first parameter is the phase of the input signal. The second parameter provides the power of the noise added to the signal. The third parameter indicates how many fundamental periods the system should correlate (integrate) over.

   (a) Call `phase_detect` with various phase values input. Leave the second two parameters blank. What happens to the two correlation values as the input signal's phase changes? (Hint: What happens at phase shifts of 0, $\pi/2$, π, and $3\pi/2$?)

   (b) How might we use these correlation values to find the phase of the input signal? (Hint: assuming there is no noise, find formulas for A_c and A_s in terms of ω, T, and ϕ.)

   (c) For a fixed phase value of $\pi/3$ and correlation over only one period (the default value for the third parameter), examine the error between your phase estimate and the true phase as we increase the signal noise power from .1 to 3. Determine and plot the square of the error values versus noise power for at least 100 different values of noise in this range. What happens as we increase the noise amplitude?

   (d) Suppose we were to increase the amplitude of our input sinusoid by a factor of 100, thus increasing the *signal-to-noise ratio*. What would happen to the errors in the phase estimates? Why does this happen?

   (e) For a fixed phase of $\pi/3$ and a fixed noise power of 3, examine the effects as we correlate over more periods of the signal. Plot the square of the phase error as we increase the number of correlation periods from 1 to 100. What do you notice as we correlate over a greater number of periods?