Laboratory # 3

Sinusoids and Sinusoidal Correlation

3.1 Introduction

Sinusoids are important signals. Part of their importance comes from their prevalence in the everyday world, where many signals can be easily described as a sinusoid or a sum of sinusoids. Another part of their importance comes from their properties when passed through linear time-invariant systems. Any linear time-invariant system whose input is a sinusoid will have an output that is a sinusoid of the same frequency, but possibly with different amplitude and phase. Since a great many natural systems are linear and time-invariant, this means that sinusoids form a powerful tool for analyzing systems.

Being able to identify the parameters of a sinusoid is a very important skill. From a plot of the sinusoid, any student of signals and systems should be able to easily identify the amplitude, phase, and frequency of that sinusoid.

However, there are many practical situations where it is necessary to build a system that identifies the amplitude, phase, and/or frequency of a sinusoid — not from a plot, but from the actual signal itself. For example, many communication systems convey information by *modulating*, i.e. perturbing, a sinusoidal signal called a *carrier*. To *demodulate* the signal received at the antenna, i.e. to recover the information conveyed in the transmitted signal, the receiver often needs to know the amplitude, phase, and frequency of the carrier. While the frequency of the sinusoidal carrier is often specified in advance, the phase is usually not specified (it is just whatever phase happens to occur when the transmitter is turned on), and the amplitude is not known because it depends on the attenuation that takes place during transmission, which is usually not known in advance. Moreover, though the carrier frequency is specified in advance, no transmitter can produce this frequency exactly. Thus, in practice the receiver must be able to "lock onto" the actual frequency that it receives.

Doppler radar provides another example. With such a system, a transmitter transmits a sinusoidal waveform at some frequency f_o. When this sinusoid reflects off a moving object, the frequency of the returned sinusoid is shifted in proportion to the velocity of the object. A system that determines the frequency of the reflected sinusoid will also be able to determine the speed of the moving object.

How can a system be designed that automatically determines the amplitude, frequency and phase of a sinusoid? One could imagine any number of heuristic methods for doing so, each based on how you would visually extract these parameters. It turns out, though, that there are more convenient methods for doing so – methods which involve correlation. In this lab, we will examine how to automatically extract parameters from a sinusoid using correlation. Along the way, we will discover how complex numbers can help us with this task. In particular, we will make use of the *complex exponential signal* and see the mathematical benefits of using an "imaginary" signal that does not really exist.

3.1.1 "The Question"

- How can we design a system that automatically determines the amplitude and phase of a sinusoid with a known frequency?
- How can we design a system that automatically determine the frequency of a sinusoid?

3.2 Background

3.2.1 Complex numbers

Before we begin, let us quickly review the basics of complex numbers. Recall the a complex number $z = x + jy$ is defined by its *real part*, x, and its *imaginary part*, y, where $j = \sqrt{-1}$. Also recall that we can rewrite any complex number into *polar form*[1] or *exponential form*, $z = re^{j\theta}$, where $r = |z|$ is the *magnitude* of the complex number and $\theta = \text{angle}(z)$ is the *angle*. We can convert between the two forms using the formulas

$$x = r\cos(\theta) \tag{3.1}$$
$$y = r\sin(\theta) \tag{3.2}$$

and

$$r = \sqrt{x^2 + y^2} \tag{3.3}$$
$$\theta = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right), & x \geq 0 \\ \tan^{-1}\left(\frac{y}{x}\right) + \pi, & x < 0 \end{cases} \tag{3.4}$$

A common operation on complex numbers is the *complex conjugate*. The complex conjugate of a complex number, z^*, is given by

$$z^* = x - jy \tag{3.5}$$
$$= re^{-j\theta} \tag{3.6}$$

Conjugation is particularly useful because $zz^* = |z|^2$.

Euler's[2] formula is a very important (and useful) relationship for complex numbers. This formula allows us to relate the polar and rectangular forms of a complex number. Euler's formula is

$$e^{j\theta} = \cos(\theta) + j\sin(\theta) \tag{3.7}$$

[1]Sometimes the polar form is written as $z = r\angle\theta$, which is a mathematically less useful form. This form, however, is useful for suggesting the interpretation of r as a radius and θ as an angle.

[2]Pronounced "oiler's".

Equally important are Euler's inverse formulas:

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad (3.8)$$

$$\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j} \quad (3.9)$$

It is *strongly recommended* that you commit these three equations to memory; you will be using them regularly throughout this course.

3.2.2 Sinusoids and complex exponential signals in continuous and discrete time

Recall that a continuous-time sinusoid in *standard form*, $s(t)$, is given by the formula

$$s(t) = A\cos(\omega_0 t + \phi), \quad (3.10)$$

where A is the sinusoid's *amplitude*, ω_0 is the sinusoid's *frequency* given in *radian frequency* (radians per second), and ϕ is the sinusoid's *phase*. It is also common to represent such a sinusoid in the following form

$$s(t) = A\cos(2\pi f_0 t + \phi), \quad (3.11)$$

where f_0 is the sinusoid's frequency given in Hertz (Hz, or cycles per second). Note that $\omega_0 = 2\pi f_0$. The frequency of a sinusoid is generally restricted to be positive.

Discrete-time sinusoids are defined in a similar way. A discrete-time sinusoid in standard form, $s[n]$, is given by the formula

$$s[n] = A\cos(\hat{\omega}_0 n + \phi), \quad (3.12)$$

where $\hat{\omega}_0$ is the frequency of the sinusoid in *discrete radian frequency* (radians per sample). For now, we will restrict $\hat{\omega}_0$ for discrete-time sinusoids to be in the range $[0, \pi]$[3]. If $s[n]$ is a sampled version of a continuous-time sinusoid, then $\hat{\omega}_0 = \omega_0 T_s$ where T_s is the *sampling period*. The sampling period is simply the time that separates two samples of a sampled signal.

The notation for sinusoids also extends to a special signal known as the *complex exponential signal*[4]. Complex exponential signals are very similar to sinusoids, and have the same three parameters. We define a continuous-time complex exponential signal, $c(t)$, in standard form as

$$c(t) = Ae^{j(\omega_0 t + \phi)} \quad (3.13)$$

It is generally useful to consider that sinusoids are composed of a sum of complex exponential signals by using Euler's inverse formulas. Thus, a sinusoid in standard form can be rewritten in several different ways:

$$\begin{aligned} s(t) &= A\cos(\omega_0 t + \phi) && (3.14) \\ &= \frac{A}{2}\left[e^{j(\omega_0 t + \phi)} + e^{-j(\omega_0 t + \phi)}\right] && (3.15) \\ &= \frac{c(t) + c^*(t)}{2} && (3.16) \\ &= \text{Re}\left\{Ae^{j(\omega_0 t + \phi)}\right\} && (3.17) \end{aligned}$$

[3] Chapter 4 will provide more details about discrete-time signals and discrete radian frequency

[4] These are sometimes referred to simply as *complex exponentials*.

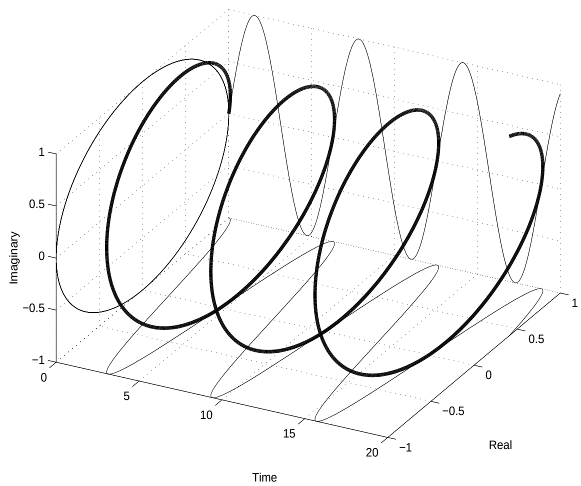

Figure 3.1: Three-dimensional plot of a complex exponential signal.

We can also interpret complex exponential signals by using Euler's formula. $c(t)$ can be viewed as the sum of a real cosine wave and an imaginary sine wave.

$$c(t) = A\cos(\omega_0 t + \phi) + jA\sin(\omega_0 t + \phi) \tag{3.18}$$

Sometimes it is useful to visualize a complex exponential signal as a "corkscrew" in three dimensions, as in Figure 3.1. Note that it is common to refer to complex exponential signals as having either positive or negative frequency. The sign of the frequency determines the "handedness" of the corkscrew.

We can define discrete-time complex exponentials in much the same way as discrete-time sinusoids. If $c[n]$ is a discrete time complex exponential, we can write a formula of the form

$$c[n] = Ae^{j(\hat{\omega}_0 n + \phi)} \tag{3.19}$$

We will restrict $\hat{\omega}_0$ for discrete-time complex exponential signals to be in the range $[-\pi, \pi]$. This range is larger than the range set for discrete-time sinusoids because we allow complex exponentials with negative frequency. This is because a sinusoid of frequency $\hat{\omega}_0$, $0 \leq \hat{\omega}_0 \leq \pi$, is also the sum of two complex exponentials — one at frequency $\hat{\omega}_0$, and the other at frequency $-\hat{\omega}_0$. Once again, we can define $\hat{\omega}_0 = \omega_0 T_s$ for a sampling period T_s.

3.2.3 Calculating A and ϕ for a sinusoid with known ω_0

We've suggested that we can use *correlation* to help us determine the amplitude and phase of a sinusoid with known frequency. We will present the derivation in continuous time, since we are generally more familiar with integrals than with summations. Once our derivations are complete, we will relate the derivation to the discrete-time case.

Suppose that we have a continuous-time sinusoid $s(t)$ (the *target sinusoid*) with known frequency ω_0 but unknown amplitude, A, and phase, ϕ. We can perform in-place correlation[5]

[5] In-place correlation between two real, continuous-time signals, $x(t)$ and $y(t)$ is defined as $C(x, y) = \int_a^b x(t)y(t)dt$. The length $(b - a)$ is the *correlation length*.

between this sinusoid and a *reference sinusoid*, $u(t)$, with the same frequency and known amplitude and phase. Without loss of generality, let $u(t)$ have $A = 1$ and $\phi = 0$. Then[6],

$$C(s,u) = \int_a^b A\cos(\omega_0 t + \phi)\cos(\omega_0 t)dt \tag{3.20}$$

$$= \frac{A}{2}\int_a^b \cos(\phi) + \cos(2\omega_0 t + \phi)dt \tag{3.21}$$

$$= \frac{A}{2}\left[\cos(\phi)t + \frac{1}{4\omega_0}\sin(\omega_0 t + \phi)\right]_a^b \tag{3.22}$$

Since we know the frequency, ω_0, we can easily set the limits of integration to include an integer number of periods of our sinusoids. In this case, the second term evaluates to zero and the correlation reduces to

$$C(s,u) = \frac{A}{2}\cos(\phi)(b-a) \tag{3.23}$$

This formula is a useful first step. If we happen to know the phase, we can readily calculate the amplitude. Similarly, if we know the amplitude, we can narrow the phase down to one of two values. If both are unknown, though, we cannot uniquely determine them.

One common way to proceed is to correlate with a second reference sinusoid that is $\frac{\pi}{2}$ out of phase with the first. Here, though, we will explore a different method which is somewhat more enlightening. Notice what happens if we use a complex exponential, $c(t) = e^{j\omega_0 t}$ as our reference signal[7]:

$$C(s,c) = \int_a^b s(t)c^*(t) \tag{3.24}$$

$$= \int_a^b A\cos(\omega_0 t + \phi)e^{-j\omega_0 t}dt \tag{3.25}$$

$$= \int_a^b \frac{A}{2}\left[e^{j(\omega_0 t+\phi)} + e^{-j(\omega_0 t+\phi)}\right]e^{-j\omega_0 t}dt \tag{3.26}$$

$$= \frac{A}{2}\int_a^b e^{j\phi} + e^{-j(2\omega_0 t+\phi)}dt \tag{3.27}$$

$$= \frac{A}{2}\left[e^{j\phi}t + \frac{-1}{2j\omega_0}e^{-j(2\omega_0 t+\phi)}\right]_a^b \tag{3.28}$$

If we again assume that we are correlating over an integer number of periods of our target sinusoid, the second term goes to zero and we are left with

$$C(s,c) = \frac{A}{2}e^{j\phi}(b-a). \tag{3.29}$$

Our correlation has resulted in a simple complex number whose magnitude is directly proportional to the amplitude of the original sinusoid and whose angle is identically equal to

[6] Recall that $\cos(A)\cos(B) = \frac{1}{2}\cos(A-B) + \frac{1}{2}\cos(A+B)$.

[7] Notice that we conjugate our complex exponential here. This is because correlation between two *complex* signals is defined as $\int x(t)y^*(t)dt$ in continuous-time or $\sum x[n]y^*[n]$ in discrete-time.

its phase! We can easily turn the above formula inside out to obtain

$$A = \frac{2}{b-a}|C(s,c)| \tag{3.30}$$
$$\phi = \text{angle}(C(s,c)) \tag{3.31}$$

We can also see from equation (3.29) that in correlating with a complex exponential signal, we have effectively calculated the phasor[8] representation of our sinusoid.

Discrete-time sinusoids and non-ideal cases

We can derive the above result for discrete-time sinusoids as well. This is because the sum of any number of samples from *exactly one period* of a uniformly sampled sinusoid or complex exponential will always equal zero, just as the integral goes to zero in the continuous-time case. Thus, if we correlate over an integral number of periods of a discrete-time sinusoid,

$$C(s,c) = \sum_{n=a}^{b} s[n]c^*[n] \tag{3.32}$$
$$= \sum_{n=a}^{b} A\cos(\hat{\omega}_0 n + \phi)e^{-j\hat{\omega}_0 n}, \tag{3.33}$$

where a and b are now integers, the result, after considerable simplification, is

$$C(s,c) = \frac{A}{2}e^{j\phi}(b-a+1). \tag{3.34}$$

Recall that $b-a+1$ is the number of samples over which the correlation is calculated. Again, we can solve for A and ϕ as

$$A = \frac{2}{b-a+1}|C(s,c)| \tag{3.35}$$
$$\phi = \text{angle}(C(s,c)) \tag{3.36}$$

There is one additional caveat to a discrete-time implementation. In order to achieve the exact result presented here, the frequency of the input sinusoid must be a rational multiple[9] of 2π. If it is not, then the sampled sinusoid is not actually periodic! In this case, we clearly cannot correlate over an integral number of periods of the sinusoid. In some cases, though, we can relax the "integral number of periods" assumption.

In particular, note what happens if we do not integrate over an integer number of periods in continuous-time. In this case, our calculation of $C(s,c)$ we will also include the second term from equation (3.28). This additional "error" term can have a magnitude no larger than $\frac{A}{2\omega_0}$, which is independent of the length of the correlation interval. The "true" correlation value given by equation (3.29), on the other hand, is proportional to the length of the correlation interval. The "error" term for the discrete-time case is more complicated, but is still independent of the length of the correlation interval. Thus, in both cases we can minimize the effects of the "error" by simply correlating over a sufficiently long time period.

[8]When we represent a sinusoid with amplitude A and phase ϕ as the complex number $Ae^{j\phi}$ to simplify the calculation of a sum of two or more sinusoids, this complex number is known as a *phasor*.

[9]That is, $\hat{\omega}_0 = 2\pi\frac{N}{M}$ for any integers N and M.

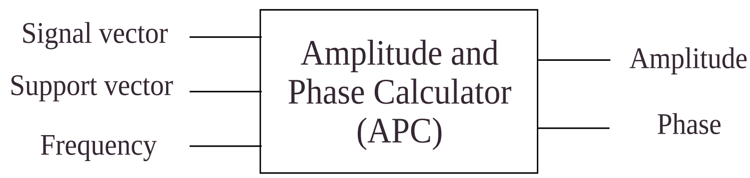

Figure 3.2: System diagram for the "amplitude and phase calculator."

Implementation Notes

In the laboratory assignment, we will be implementing an "amplitude and phase calculator" (APC) as a MATLAB function. A diagram of this system is shown in Figure 3.2. The system takes three input parameters. The first is the *signal vector* which contains the sinusoid itself. The second is the *support vector* for the sinusoid. Note that the support vector can contain either integers (for a purely discrete-time signal) or real numbers (for a sampled continuous-time signal). The third input parameter is the frequency of the sinusoid in radians per sample (for a discrete-time sinusoid) or radians per second (for a sampled sinusoid). Note that for the system's output to be exact, the input sinusoid must be defined over exactly an integer number of periods.

The system outputs the sinusoid's amplitude and its phase in radians. The system calculates these outputs by first computing the in-place correlation given by equations (3.32) and (3.33). Then, this correlation value is used with equations (3.35) and (3.36) to compute the amplitude and phase.

3.2.4 Determining the frequency of a target sinusoid

Suppose now that we are given the task of automatically determining the frequency of a particular target sinusoid. It turns out that correlation can help us with this problem as well. Consider the following case. Let the target sinusoid be defined by $s(t) = A\cos(\omega_s t + \phi)$, where ω_s, A, and ϕ are all unknown. We correlate $s(t)$ with a complex exponential signal, $c(t) = e^{j\omega_c t}$, with frequency ω_c, where $\omega_s \neq \omega_c$:

$$C(s,c) = \int_a^b s(t)c^*(t) \quad (3.37)$$

$$= \int_a^b A\cos(\omega_s t + \phi)e^{-j(\omega_c t)}dt \quad (3.38)$$

$$= \int_a^b \frac{A}{2}\left[e^{j(\omega_s t+\phi)} + e^{-j(\omega_s t+\phi)}\right]e^{-j(\omega_c t)}dt \quad (3.39)$$

$$= \frac{A}{2}\int_a^b e^{j[(\omega_s-\omega_c)t+\phi]} + e^{-j[(\omega_s+\omega_c)t+\phi]}dt \quad (3.40)$$

$$= \frac{A}{2}\left[\frac{1}{\omega_s-\omega_c}e^{j[(\omega_s-\omega_c)t+\phi]} + \frac{1}{\omega_s+\omega_c}e^{-j[(\omega_s+\omega_c)t+\phi]}\right]_a^b \quad (3.41)$$

Here, let us make a simplifying assumption and assume that $(\omega_s + \omega_c)$ is sufficiently large that we can neglect the second term. Then, we have

$$C(s,c) \quad \approx \quad \frac{A}{2(\omega_s - \omega_c)}\left[e^{j[(\omega_s-\omega_c)b+\phi]} - e^{-j[(\omega_s-\omega_c)a+\phi]}\right] \tag{3.42}$$

The resulting equation is primarily dependent upon the frequency difference between the target sinusoid and our reference signal. Though it is not immediately apparent, the value of this correlation converges to the value of equation (3.29) as the frequency difference $(\omega_s - \omega_c)$ approaches zero.

Consider the length-normalized correlation, $\widetilde{C}(s,c)$, defined as

$$\widetilde{C}(s,c) = \frac{C(s,c)}{b-a}. \tag{3.43}$$

The length-normalized correlation is constant if the reference and target signals have the same frequency, as we can see from equation (3.29). However, when the signals have different frequencies, the magnitude of the length-normalized correlation becomes smaller as we correlate over a longer period of time. (This happens more slowly as the frequency difference becomes smaller.) In the limit as the correlation length goes to infinity, *the length-normalized correlation goes to zero unless the frequencies match exactly*. This is a very important theoretical result in signals and systems.

Another special case occurs when we correlate over a *common period* of the target and reference signals. This occurs when our correlation interval includes an integer number of periods of *both* the target signal and reference signal. In this case, the correlation in equation (3.42), for signals of different frequencies, is identically zero[10]. Of course, the correlation is *not* zero when the frequencies match. Note that this is the same condition required for equation (3.29) to be exact.

It is worthwhile to note that all of these results apply to the discrete-time case as well. In this case, we define the length-normalized correlation, $\widehat{C}(s,c)$, as

$$\widehat{C}(s,c) = \frac{C(s,c)}{b-a+1}. \tag{3.44}$$

$\widehat{C}(s,c)$ again goes to zero as the quantity $(b-a+1) \to \infty$ when the reference and target signals' frequencies do not match. When the frequencies do match, $\widehat{C}(s,c)$ remains constant. In the discrete-time case as well, the correlation goes to zero when correlating over a common period of both signals unless the frequencies match exactly.

How does all of this help us to determine the frequency of the target sinusoid? The answer is perhaps less elegant than one might hope; basically, we "guess and check". If we have no prior knowledge about possible frequencies for the sinusoid, we need to check the correlation with complex exponentials having a variety of frequencies. Then, whichever complex exponential yields the highest correlation, we take the frequency of that complex exponential as our estimate of the frequency of the target signal. In the next section, we will formalize this algorithm for the discrete-time case.

A frequency identification algorithm

Suppose that we have a discrete-time target sinusoid $s[n]$ with unknown amplitude, frequency, and phase. Let the length of $s[n]$ be N; we will calculate the length-normalized

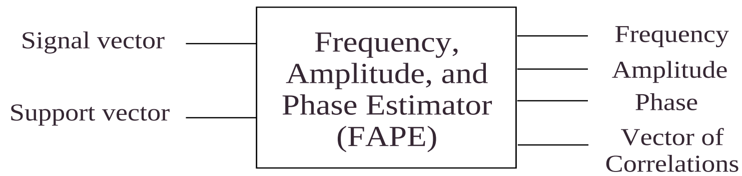

Figure 3.3: System diagram for the "frequency, amplitude, and phase estimator."

correlation over this entire length. Since we would like to be correlating over an integral number of periods of our reference signal (so that we can take advantage of the "common period" property if possible), we'll select frequencies of the form $\hat{\omega} = \frac{2\pi k}{N}$ where[11] $k = 1, 2, 3, \ldots, \lfloor \frac{N}{2} \rfloor$. This selection of frequencies provides us with uniform coverage of all possible frequencies in the range $[0, \pi]$, while not exceeding this range. Further, N contains an integral number of periods (k periods, in fact) of the periodic signals with these frequencies.

For simplicity, let us store the correlations that we calculate in another discrete-time signal, X. Specifically, let $X[k]$ be the length-normalized correlation for the reference complex exponential with frequency $\frac{2\pi k}{N}$, calculated as

$$X[k] = \frac{1}{N} \sum_{n=0}^{N} s[n] e^{-j\left(\frac{2\pi k}{N}\right)n}, \qquad k = 1, 2, 3, \ldots, \left\lfloor \frac{N}{2} \right\rfloor \tag{3.45}$$

Remember that $X[k]$ will generally be complex. To estimate the frequency of the target sinusoid, we simply identify value, k_{max}, of k for which $|X[k]|$ is highest. Then, our estimated frequency, $\hat{\omega}_{est}$, is given by

$$\hat{\omega}_{est} = \frac{2\pi k_{max}}{N} \tag{3.46}$$

Now that we have an estimated frequency, we should also be able to estimate the amplitude and phase as well. In fact, we have almost calculated these estimates already. They are:

$$A \approx 2|X[k_{max}]| \tag{3.47}$$

$$\phi \approx \text{angle}(X[k_{max}]) \tag{3.48}$$

There is one problem here, however. Previously, we needed to know the frequency exactly to determine the amplitude and phase; now, we only know the frequency approximately. In the laboratory assignment, we will see the effect of this approximation.

In the laboratory assignment, we will be developing a system that can automatically estimate the amplitude, phase, and frequency of a sinusoid. A block diagram of the "frequency, amplitude, and phase estimator" (FAPE) system is given in Figure 3.3. Unlike the APC, this system takes only two input parameters: a signal vector and the corresponding support vector. Further, the support vector must contain only integers; that is, this system does *not* accept sampled continuous-time signals. The system has four output parameters. The first three are the estimates of the frequency, amplitude, and phase of the input sinusoid.

[11] $\lfloor \cdot \rfloor$ indicates the *greatest integer* or *floor* function. Basically, we want k to be an integer, but it cannot exceed $\frac{N}{2}$.

The fourth is the vector of correlations examined by the function, $X[k]$. It is often useful to examine this vector to get a sense of what the system is doing.

3.3 Some Matlab commands for this lab

- **Constructing complex numbers:** Matlab represents all complex numbers in rectangular form. To enter a complex number, simply type `5 + 6*j` (for instance). Note that both `i` and `j` are used to represent $\sqrt{1}$ (unless you have used one or the other as some other variable). To enter a complex number in polar form, type `2*exp(j*pi/3)` (for instance).

  You may be wondering how Matlab actually works with complex numbers, given that complex numbers are, in general, the sum of a real number and an imaginary one. The point is that the imaginary component of a complex number is in fact a *real number*, which Matlab stores in the usual way. It thinks of a complex number as a pair of floating point numbers, one to be interpreted as the real part and the other to be interpreted as the imaginary part. And it knows the rules of arithmetic to apply to such pairs of numbers in order to do what complex arithmetic is supposed to do.

- **Extracting parts of complex numbers:** If `z` contains a complex number (or an array of complex numbers), you can find the real and imaginary parts using the commands `real(z)` and `imag(z)`, respectively. You can obtain the magnitude and angle of a complex number (or an array of complex numbers) using the commands `abs(z)` and `angle(z)`, respectively.

- **Complex conjugation:** To compute the complex conjugate of a value (or array) `z`, simply use the Matlab command `conj(z)`.

- **Building discrete-time sinusoids:** To construct a discrete-time sinusoid in Matlab, you first need to define a *support vector*. For instance, you might want the support vector to extend from 0 to 99, in which case you can use the command

  ```
  >> n = 0:99;
  ```

  Then, you can create the sinusoid's *signal vector*. If you want $A = 2$, $\hat{\omega}_0 = \pi/8$, and $\phi = 2\pi/3$, for instance, you would use the command:

  ```
  >> s = 2*cos(pi/8*n + 2*pi/3);
  ```

- **Building discrete-time complex exponential signals:** The process for creating a complex exponential in Matlab is basically identical to the process for creating discrete time sinusoids:

  ```
  >> n = 0:99;
  >> c = 2*exp(j*(pi/8*n + 2*pi/3));
  ```

- **Finding the index of the maximum value in a vector:** Sometimes we don't just want to find the maximum value in a vector; instead, we need to know where that maximum value is located. The `max` command will do this for us. If `v` is a vector and you use the command


```
>> [max_value, index] = max(v);
```

the variable `max_value` will contain largest value in the vector, and `index` contains position of `max_value` in `v`.

- Matlab **commands to help you visually determine the amplitude, frequency, and phase of a sinusoid:** Sometimes you may need to determine the frequency, phase, and amplitude of a sinusoid from a Matlab plot. In these cases, there three commands that are quite useful. First, the command `grid on` provides includes a reference grid on the plot; this makes it easier to see where the sinusoid crosses zero (for instance). The `zoom` command is also useful, since you can drag a zoom box to zoom in on any part of the sinusoid. Finally, you can use `axis` in conjunction with `zoom` to find the period of the signal. To do so, simply zoom in on exactly one period of the signal and type `axis`. Matlab will return the current axis limits as `[x_min, x_max, y_min, y_max]`.

3.4 Demonstrations in the Lab Section

- Complex Numbers in Matlab
- Sinusoids and complex exponentials in Matlab
- Sinusoidal correlation: matching frequencies
- Sinusoidal correlation: different frequencies
- FAPE: the Frequency, Amplitude, and Phase Estimator

3.5 Laboratory Assignment

1. Execute the following commands:

```
>> t = linspace(-0.5, 2, 1000);
>> plot(t,cos(linspace(-7.5,27,1000)),'k:');
```

   (a) Visually identify the amplitude, frequency, and phase of the continuous-time (sampled) sinusoid that you've just plotted.
      - [4] Include your estimated values in your report. Reduce your answers to decimal form.
      - [3] What is the phasor that corresponds to this sinusoid? Write it in both rectangular and polar form. (Again, keep your answers in decimal form. You should use Matlab to perform these calculations.)

   (b) Verify your answers in the previous problem by generating a sinusoid using those parameters and plotting them on the above graph using `hold on`. Use `t` as your time axis/support vector. The new plot should be close to the original, but it does not need to be exactly correct.
      - [3] Include the resulting graph in your report. Remember to include a `legend`.

2. Download the file `apc.m`. This is a "skeleton" M-file for the function `apc`, which implements the "amplitude and phase calculator" described in Section 3.2.3. Also, generate the following discrete-time sinusoid (`s_test`) with its support vector (`n_test`):

```
>> n_test = 0:99;
>> s_test = 1.3*cos(n_test*pi/10 + 2.8);
```

   (a) What are the amplitude, frequency, and phase of `s_test`?
      - [2] Include your answers in your lab report.

   (b) Complete the function `apc`. You can use the signal `s_test` to test the operation of your function. You may also wish to use the compiled function `apc_demo.dll`[12] to test your results on other sinusoids.
      - [10] Include the code for `apc` in your MATLAB appendix.

   (c) Download the file `lab3_data.mat`. This `.mat` file contains the support vector (`t_samp`) and signal vector (`s_samp`) for a sampled continuous-time sinusoid with frequency $\omega_0 = 200\pi$.
      - [2] From `t_samp`, determine the sampling frequency of this signal.
      - [3] Use `apc` to determine the amplitude and phase of the sinusoid exactly.

   (d) Now we would like to investigate the behavior of `apc` in cases when we do not correlate over exactly an integral number of periods of the target sinusoid. Generate the following sinusoid:

```
>> apc_support = 0:80;
>> apc_test = cos(apc_support*pi/15);
```

   This is a sinusoid with a frequency of $\hat{\omega}_0 = \frac{\pi}{15}$, unit amplitude, and zero phase shift.
      - [2] Plot `apc_test` and include the plot in your report.
      - [2] What is the fundamental period of `apc_test` in samples?
      - [2] Approximately how many periods are included in `apc_test`?
      - [2] Use `apc` to estimate the amplitude and phase of this sinusoid. What are the amplitude and phase errors for this signal?

   (e) Now we wish to examine a large number of different lengths of this sinusoid. You will do this by writing a `for` loop that repeats the previous part for many different values of the length of the incoming sinusoid. Specifically, write a `for` loop with loop counter `support_length` ranging over `10:500`. In each iteration of the loop, you should

      i. Set `apc_support` equal to `0:(support_length`1)-,
      ii. Recalculate `apc_test` using the new `apc_support`,
      iii. Use `apc` to estimate the amplitude and phase of `apc_test`, and
      iv. Store these estimates in two separate vectors.

   Put your code in an M-file script so that you can run it easily.
      - [8] Include your code in the MATLAB appendix.

[12] Note that `apc_demo.dll` will not work on inputs with non-integer support vectors.

- [4] Use subplot to plot the amplitude and phase estimates as a function of support length in two subplots of the same figure. You should be able to see both local oscillation of the estimates and a global decrease in error with increased support length.
- [3] At what support lengths are the amplitude estimates correct (i.e., equal to 1)?
- [3] What minimum support length do we need to be sure that the phase error is less than 0.01 radians?

3. Download the file fape.m. This is a "skeleton" M-file for the function fape, which implements the "frequency, amplitude, and phase estimator" system described in Section 3.2.4.

   (a) Complete the fape function. You can use n_test and s_test from Problem 2 along with the compiled fape_demo.dll to check your function's results.

   - [16] Include the completed code in your report's MATLAB appendix.
   - [2] What are the frequency, amplitude, and phase estimates returned by fape for n_test and s_test? Are these estimates correct?
   - [3] Use stem and abs to plot the magnitude of the vector of correlations returned by fape versus the associated frequencies.
   - [3] What do you notice about this plot? What can you deduce from this fact? (Hint: Consider what this plot tells you about the input signal and the returned estimates.)

   (b) lab3_data.mat contains the variables fape_test_n (a support vector) and fape_test_s (its associated sinusoidal signal). Run fape on this signal.

   - [3] What are the frequency, amplitude, and phase estimates that are returned?
   - [3] Use stem and abs to plot the magnitude of the returned vector of correlations.
   - [3] Plot fape_test_s and a new sinusoid that you generate from the parameter estimates returned by FAPE on the same figure (using hold on). Use fape_test_n as the support vector for the new sinusoid. Make sure you use different line types and include a legend.
   - [2] What can you say about the accuracy of estimates returned by FAPE?

   (c) Finally, we would like to examine the error characteristics of fape in more detail, as we did with apc. We will use the following sinusoid:

   ```
   >> fape_support = 0:N;
   >> fape_test = cos(fape_support*0.4);
   ```

   where we will vary N throughout the problem. Note in particular that fape_test is not truly periodic in discrete-time.

   Generate fape_test for N = 50, 100, 170, and 275. Run fape on each signal.

   - [4] Use stem and subplot to plot the resulting correlation vectors for each signal in separate subplots of the same figure. (Make sure you indicate which subplot is associated with which value of N.)

- [4] What are the corresponding frequency, amplitude, and phase estimates? (Put your estimates into a table in your report.)
- [4] What can you say about the accuracy of `fape` for each type of parameter? (Hint: There is a strong relationship between the error for each of the three parameters. Do the errors generally decrease as `N` increases? What range of values do the estimates take on? You should examine some other value of `N` or, better yet, consider plotting the parameters versus `N` over some range.)

