

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 determines 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’s² 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

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

Equally important are Euler’s inverse formulas:

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

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

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

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”.

3.2.2 Sinusoidal and complex exponential signals in continuous 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 $A > 0$ 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.

The notation for sinusoids also extends to a special signal known as the *complex exponential signal*³. 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

$$c(t) = Ae^{j(\omega_0 t + \phi)} \quad (3.12)$$

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:

$$s(t) = A \cos(\omega_0 t + \phi) \quad (3.13)$$

$$= \frac{A}{2} \left[e^{j(\omega_0 t + \phi)} + e^{-j(\omega_0 t + \phi)} \right] \quad (3.14)$$

$$= \frac{A}{2} (c(t) + c^*(t)) \quad (3.15)$$

$$= \operatorname{Re} \left\{ Ae^{j(\omega_0 t + \phi)} \right\} \quad (3.16)$$

Using Euler's formula, we can also interpret a complex exponential signal $c(t)$ as the sum of a real cosine wave and an imaginary sine wave:

$$c(t) = A \cos(\omega_0 t + \phi) + jA \sin(\omega_0 t + \phi) \quad (3.17)$$

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 permit complex exponential signals to have either positive or negative frequency. The sign of the frequency determines the “handedness” of the corkscrew.

3.2.3 Finding the amplitude and phase of a sinusoid with known frequency

We've suggested that we can use *correlation* to help us determine the amplitude and phase of a sinusoid with known frequency. Suppose that we have a continuous-time sinusoid (the *target sinusoid*)

$$s(t) = A \cos(\omega_0 t + \phi) \quad (3.18)$$

with known frequency ω_0 , but unknown amplitude A and phase ϕ , which we would like to find. We can perform in-place correlation⁴ between this sinusoid and a *reference sinusoid*, $u(t)$, with the same frequency and known amplitude

³These are sometimes referred to simply as *complex exponentials*.

⁴In-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*.

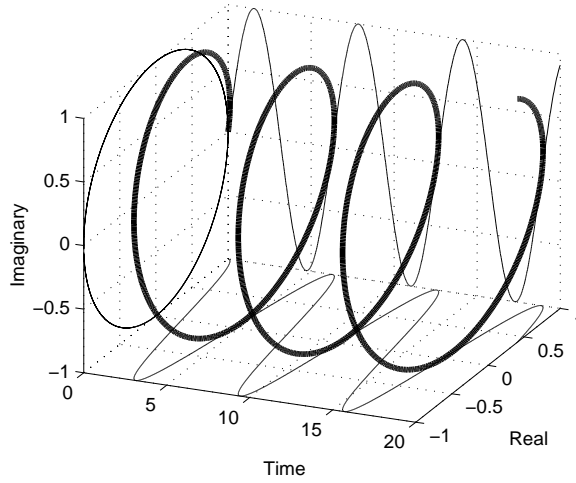


Figure 3.1: Three-dimensional plot of a complex exponential signal.

and phase. Without loss of generality, let $u(t)$ have $A = 1$ and $\phi = 0$. Then⁵,

$$C(s, u) = \int_{t_1}^{t_2} A \cos(\omega_0 t + \phi) \cos(\omega_0 t) dt \quad (3.19)$$

$$= \frac{A}{2} \int_{t_1}^{t_2} \cos(\phi) + \cos(2\omega_0 t + \phi) dt \quad (3.20)$$

$$= \frac{A}{2} \left[\cos(\phi)t + \frac{1}{2\omega_0} \sin(2\omega_0 t + \phi) \right]_{t_1}^{t_2} \quad (3.21)$$

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

$$C(s, u) = \frac{A}{2} \cos(\phi)(t_2 - t_1) \quad (3.22)$$

This formula is a useful first step. If we happen to know the phase ϕ , then we can readily calculate the amplitude A of $s(t)$ from $C(s, u)$. Similarly, if we know the amplitude A , we can narrow the phase ϕ down to one of two values. If both amplitude and phase are unknown, though, we cannot uniquely determine them.

If the interval over which we correlate is not a multiple of the fundamental period of $u(t)$, then the second term in equation (3.21) may be nonzero. However, if as commonly happens ω_0 is much greater than one, then the second term will be so small that it can be ignored, and equation (3.22) holds with approximate equality.

To resolve the ambiguity when both amplitude and phase are unknown, one common approach 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,

$$c(t) = e^{j\omega_0 t} \quad (3.23)$$

⁵Recall that $\cos(A) \cos(B) = \frac{1}{2} \cos(A - B) + \frac{1}{2} \cos(A + B)$.

3.2.3 Finding the amplitude and phase of a sinusoid with known frequency

as our reference signal⁶:

$$C(s, c) = \int_{t_1}^{t_2} s(t)c^*(t) dt \quad (3.24)$$

$$= \int_{t_1}^{t_2} A \cos(\omega_0 t + \phi) e^{-j\omega_0 t} dt \quad (3.25)$$

$$= \int_{t_1}^{t_2} \frac{A}{2} [e^{j(\omega_0 t + \phi)} + e^{-j(\omega_0 t + \phi)}] e^{-j\omega_0 t} dt \quad (3.26)$$

$$= \frac{A}{2} \int_{t_1}^{t_2} e^{j\phi} + e^{-j(2\omega_0 t + \phi)} dt \quad (3.27)$$

$$= \frac{A}{2} \left[e^{j\phi} t + \frac{-1}{2j\omega_0} e^{-j(2\omega_0 t + \phi)} \right]_{t_1}^{t_2} \quad (3.28)$$

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

$$C(s, c) = \frac{A}{2} e^{j\phi} (t_2 - t_1). \quad (3.29)$$

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

$$A = \frac{2}{t_2 - t_1} |C(s, c)| \quad (3.30)$$

$$\phi = \text{angle}(C(s, c)) \quad (3.31)$$

We can also see from equation (3.29) that in correlating with a complex exponential signal, we have effectively calculated the phasor⁷ representation of our sinusoid.

As with the case of correlating with a sinusoid, we note that when the interval over which we correlate is not a multiple of the fundamental period of $c(t)$, then the second term in equation (3.28) is not zero. However, if as commonly happens ω_0 is much greater than 1, then the second term will again be small enough that it can be ignored, and equations (3.29), (3.30), and (3.31) hold with approximate equality.

The Amplitude and Phase Calculator

In this lab we will implement a system that estimates the amplitude and phase of a sinusoid with a known frequency. Since we will do this using a computer and MATLAB, we must necessarily work with sampled version of the signals $s(t)$ and $c(t)$. Specifically, if T_s denotes the sampling period, then we work with the discrete-time signals

$$s[n] = s(nT_s) = A \cos(\omega_0 T_s n + \phi) \quad (3.32)$$

$$c[n] = c(nT_s) = e^{j\omega_0 T_s n} \quad (3.33)$$

As shown below, when T_s is small, the correlation between $s(t)$ and $c(t)$ can be approximately computed from the correlation between $s[n]$ and $c[n]$. Let $\{n_1, \dots, n_2\}$ denote the discrete-time interval corresponding to the continuous-time interval $[t_1, t_2]$, and let $N = n_2 - n_1 + 1$ denote the number of samples taken in the interval $[t_1, t_2]$, so that

⁶Notice that we conjugate our complex exponential here. This is because correlation between two *complex* signals is defined as $\int x(t)y^*(t)dt$.

⁷When 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*.

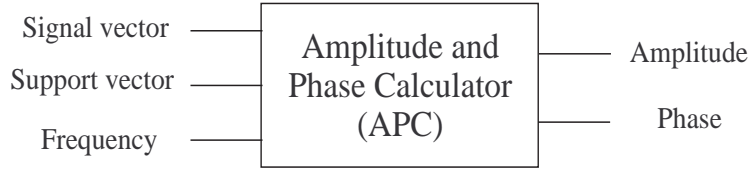


Figure 3.2: System diagram for the “amplitude and phase calculator.”

$t_2 - t_1 \approx NT_s$. Then,

$$C(s, c) = \int_{t_1}^{t_2} s(t) c^*(t) dt \quad (3.34)$$

$$= \sum_{n=n_1}^{n_2} \int_{nT_s}^{(n+1)T_s} s(t) c^*(t) dt \quad (3.35)$$

$$\approx \sum_{n=n_1}^{n_2} \int_{nT_s}^{(n+1)T_s} s(nT_s) c^*(nT_s) dt \quad (3.36)$$

$$= \sum_{n=n_1}^{n_2} s(nT_s) c^*(nT_s) T_s \quad (3.37)$$

$$= \sum_{n=n_1}^{n_2} s[n] c^*[n] T_s \quad (3.38)$$

$$= C_d(s, c) T_s \quad (3.39)$$

where the approximation leading to the third relation is valid because T_s is small, and consequently the signals $s(t)$ and $c(t)$ change little over each T_s second sampling interval, and where we use $C_d(s, c)$ to denote the correlation between the discrete-time signals $s[n]$ and $c[n]$, to distinguish it from the correlation between continuous time signals $s(t)$ and $c(t)$. We see from this derivation that the continuous-time correlation is approximately the discrete-time correlation multiplied by the sampling interval, i.e.

$$C(s, c) \approx C_d(s, c) T_s \quad (3.40)$$

We will use this value of correlation in equations (3.30) and (3.31) to estimate the amplitude and phase of a continuous-time sinusoid.

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. The third input parameter is the frequency of the reference sinusoid in radians per second. Note that for the system’s output to be exact, the input sinusoid must be defined over exactly an integer number of fundamental 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 (3.37) or (3.38). Then, this correlation value is used with equations (3.30) and (3.31) to compute the amplitude and phase. Note that in equation (3.30), we need to replace $t_2 - t_1$ with $N = n_2 - n_1 + 1$ when implementing in discrete time.

3.2.4 Determining the frequency of a target sinusoid

Suppose now that we are given the task of making a system that automatically determines the frequency of a 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 $\omega_s \neq \omega_c$:

$$C(s, c) = \int_{t_1}^{t_2} s(t)c^*(t) dt \quad (3.41)$$

$$= \int_{t_1}^{t_2} A \cos(\omega_s t + \phi) e^{-j(\omega_c t)} dt \quad (3.42)$$

$$= \int_{t_1}^{t_2} \frac{A}{2} \left[e^{j(\omega_s t + \phi)} + e^{-j(\omega_s t + \phi)} \right] e^{-j(\omega_c t)} dt \quad (3.43)$$

$$= \frac{A}{2} \int_{t_1}^{t_2} e^{j[(\omega_s - \omega_c)t + \phi]} + e^{-j[(\omega_s + \omega_c)t + \phi]} dt \quad (3.44)$$

$$= \frac{A}{2} \left[\frac{1}{\omega_s - \omega_c} e^{j[(\omega_s - \omega_c)t + \phi]} + \frac{1}{\omega_s + \omega_c} e^{-j[(\omega_s + \omega_c)t + \phi]} \right]_{t_1}^{t_2} \quad (3.45)$$

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

$$C(s, c) \approx \frac{A}{2(\omega_s - \omega_c)} \left[e^{j[(\omega_s - \omega_c)t_2 + \phi]} - e^{-j[(\omega_s - \omega_c)t_1 + \phi]} \right] \quad (3.46)$$

The resulting equation depends primarily on the frequency difference $(\omega_s - \omega_c)$ 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 $(\omega_s - \omega_c)$ approaches zero.

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

$$\tilde{C}(s, c) = \frac{C(s, c)}{t_2 - t_1}. \quad (3.47)$$

One can see from equation (3.29) that when the reference and target signals have the same frequency, the length-normalized correlation does not depend on the length of the signal. However, when the signals have different frequencies, one can see from equations (3.46) and (3.47) that 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 (3.46), for signals of different frequencies, is identically zero⁸. 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.

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.

A frequency estimation algorithm

Suppose that we have a continuous-time target sinusoid $s(t)$ with support $[0, T]$ with unknown amplitude, frequency, and phase. To estimate these parameters, we'll calculate the length-normalized correlation between this signal and reference complex exponentials with various frequencies over the signal's T second length. Then, we look for the frequency that produces the largest correlation.

We choose the frequencies of these complex exponentials to be multiples $1/T$ so that the correlation is over an integer number of periods of each complex exponential. That is, the frequencies will be $1/T, 2/T, \dots$. As in the previous subsection, we'll need to approximately compute the correlation from samples of $s(t)$ and each reference exponential, taken with some small sampling interval T_s . For convenience we'll take N samples and choose $T_s = T/N$ for some large even integer N . With samples taken at intervals of T_s seconds, we cannot hope to do a good job of correlating with complex exponentials with very high frequency. The rule of thumb that you will learn in Chapter 4 is that, at the very least, two samples are needed from each period of the signal being sampled. Therefore, the highest frequency with which we will correlate is, approximately, $1/(2T_s)$. Specifically, we will correlate $s(t)$ with complex exponentials at frequencies

$$\frac{1}{T}, \frac{2}{T}, \dots, \frac{N}{2T} = \frac{1}{2T_s} \quad (3.48)$$

Then, for $k = 1, 2, \dots, N/2$, the length normalized correlation of $s(t)$ with the complex exponential at frequency $\frac{k}{T}$ is (using equations (3.37), (3.38) and (3.47))

$$X[k] \approx \frac{1}{T} \sum_{n=0}^{N-1} s(nT_s) e^{-j2\pi \frac{k}{T} nT_s T_s} \quad (3.49)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-j2\pi \frac{k}{N} n} \quad (3.50)$$

where we have used the fact that $T_s/T = 1/N$ and where we have denoted the result $X[k]$ because this is the notation used in future labs for the last formula above. Thus, the output output of these correlations is the set of $N/2$ numbers $X[1], \dots, X[N/2]$. Remember that $X[k]$ will generally be complex. To estimate the frequency of the target sinusoid, we simply identify the value of k for which $|X[k]|$ is largest. With k_{max} denoting this value, our estimated frequency, ω_{est} , is

$$\omega_{est} = 2\pi \frac{k_{max}}{T} = 2\pi \frac{k_{max}}{NT_s} \quad (3.51)$$

Now that we have estimated the frequency, we should also be able to estimate the amplitude and phase as well. In fact, we have almost calculated these estimates already. From equations (3.30) and (3.31), they are:

$$A_{est} = 2|X[k_{max}]| \quad (3.52)$$

$$\phi_{est} = \text{angle}(X[k_{max}]) \quad (3.53)$$

There is one potential problem here, however. Previously, we assumed the frequency was known exactly when determining 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. The system has four output parameters. The first three are the estimates of the frequency, amplitude, and phase of the input sinusoid. The fourth is the vector of correlations $X[1], \dots, X[N/2]$ produced by the correlations. It is often useful to examine this vector to get a sense of what the system is doing.

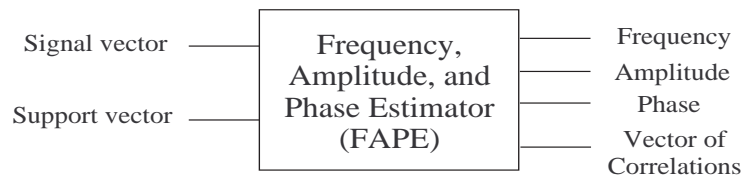


Figure 3.3: System diagram for the “frequency, amplitude, and phase estimator.”

Estimating doppler shift

When a sinusoidal transmitted signal reflects off an object moving towards the transmitter at speed v_o , the signal returned to the transmitter is again a sinusoid, but with the higher frequency

$$\omega_r = \omega_t * \frac{v}{v - v_o} \quad (3.54)$$

where ω_t is the original frequency of the transmitted signal and v is the speed of propagation in the given medium. This is called the *Doppler shift* phenomenon. If one measures ω_r , e.g. with FAPE, then one can use equation (3.54) to compute the speed of the object v_s , assuming of course, that ω_t and v are known. Indeed, this is how radar systems are able to measure the speed of automobiles, airplanes, baseballs, wind, etc. In the lab assignment you will be asked to estimate the velocity of an underwater object from a reflection of a sonar signal.

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 fact 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)`.
- **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);
```

Laboratory 3. Sinusoids and Sinusoidal Correlation

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]`.
- **Calling `apc`:** The function `apc`, which you will be writing in this laboratory, estimates amplitude and phase of a continuous-time target sinusoid from its samples. The input parameters are a (sampled) target sinusoid `s`, the sinusoid's support vector `t`, and the continuous-time frequency `w0` in radians per second. We call `apc` like this:

```
>> [A,phi] = apc(s,t,w0);
```

Note that a compiled version of this function, called `apc_demo.dll`, is also available.

- **Calling `fape`:** The function `fape`, which you will be writing in this laboratory, implements the frequency, amplitude, and phase estimator system. This function accepts the samples of a target continuous-time sinusoid `s` and it's support vector `t`, like this:

```
>> [frq,A,phi,X] = apc(s,t);
```

where `frq` is the estimated frequency in radians per second, `A` is the estimated amplitude, `phi` is the estimated phase, and `X` is the vector of correlations, `X[1], . . . , X[N/2]` between `s` and each reference complex exponential. Note that a compiled version of this function called, `fape_demo.dll`, is also available.

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. (Understanding sinusoids) Execute the following commands:

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

- (a) (Extracting sinusoid parameters) Visually identify the amplitude, continuous-time 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) (Checking your parameters) 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. (The Amplitude and Phase Calculator) In this problem we will complete and test a function which implements the "Amplitude and Phase Calculator", as described in Section 3.2.3. Download the file `apc.m`. This is a "skeleton" M-file for the "amplitude and phase calculator". Also, generate the following sinusoid (`s_test`) with its support vector (`t_test`):

```
>> t_test = 0:0.01:.99;
>> s_test = 1.3*cos(t_test*10*pi + 2.8);
```

- (a) (Identify sinusoid parameters by hand) What are the amplitude, frequency in radians per second, and phase of `s_test`?
- [2] Include your answers in your lab report.
- (b) (Write the APC) Complete the function `apc`. You should use the signal `s_test` to test the operation of your function. You may also wish to use the compiled function `apc_demo.dll` to test your results on other sinusoids.
- [10] Include the code for `apc` in your MATLAB appendix.
- (c) (Test APC on a sinusoid with unknown parameters)
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 a continuous-time frequency of $\omega_0 = 200\pi$ radians.
- [4] From `t_samp`, determine the sampling period, T_s , of this signal.
 - [3] Use `apc`⁹ to determine the amplitude and phase of the sinusoid exactly.
- (d) (APC in a non-ideal case) What happens if we use `apc` to correlate over a non-integral number of periods of our target sinusoid? We will investigate this question in this problem and the next. First, let's examine a single non-integral number of periods. Generate the following sinusoid:

```
>> apc_support = 0:0.1:8;
>> apc_test = cos(apc_support*2*pi/3);
```

This is a sinusoid with a frequency of $\omega_0 = \frac{2\pi}{3}$ radians per second, 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`?
- [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?

⁹If you failed to correctly complete `apc.m`, you may use `apc_demo.dll` for the following problems. If you use the demo function, please indicate this in your lab report.

Laboratory 3. Sinusoids and Sinusoidal Correlation

- (e) (APC in many non-ideal cases) 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 values of `1:0.1:50` seconds. In each iteration of the loop, you should
- Set `apc_support` equal to `0:0.1:(support_length-0.1)`,
 - Recalculate `apc_test` using the new `apc_support`,
 - Use `apc` to estimate the amplitude and phase of `apc_test`, and
 - 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.
- [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. (The Frequency, Amplitude, and Phase Estimator) In this problem, we'll explore the frequency, amplitude and phase estimator, as described in Section 3.2.4. Download the file `fape.m`. This is a "skeleton" M-file for the "frequency, amplitude, and phase estimator" system.

- (a) (Write the FAPE) Complete the `fape` function. You can use `t_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.
- [3] What are the frequency (in radians per second), amplitude, and phase estimates returned by `fape` for `t_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 returned estimates.)

- (b) (Running FAPE on in a non-ideal case) In this problem, we'll see what happens to FAPE when the target sinusoid does not include an integral number of periods. `lab3_data.mat` contains the variables `fape_test_t` (a support vector) and `fape_test_s` (its associated sinusoidal signal). Run `fape` on this signal.

- [3] What are the frequency in radians per second, amplitude, and phase estimates that are returned?
- [3] Use `stem` and `abs` to plot the magnitude of the returned vector of correlations.
- [4] 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_t` as the support vector for the new sinusoid. Make sure you use different line types and include a legend.
- [3] What can you say about the accuracy of estimates returned by FAPE?
- [3] Compare the plot of the correlations generated in this problem and in Problem 3a. What do these different plots tell you?

Food for thought: Investigate the error characteristics of `fape` as you did with `apc` in problem 2e. Do the frequency, amplitude, and phase estimates improve as we use longer support lengths? Which parameter is exhibits the most error? What does the vector of correlations, $X[k]$, tell you about these estimates?

- (c) Measuring speed via Doppler shift. A sonar transmitter in the ocean emits a sinusoidal signal with frequency 1000 Hz, and the signal reflects off an object moving toward the transmitter. The received signal can be found in the MATLAB workspace `lab3_data.mat`. The signal vector is called `s_sonar` and the support vector is `t_sonar`. The speed of sound in salt water is approximately 1450 meters/second. (Note: because the signal is rather long, it may take a little while for FAPE to run.)

- [4] Estimate the speed of the object.

Food for thought: Use `randn` to add some random noise to `s_sonar` and observe how your estimate changes. How much noise do you need to add to produce an error? Does the system degrade gracefully? (That is, is the amount of error proportional to the amount of noise?)

4. On the front page of your report, please provide an estimate of the average amount of time spent outside of lab by each member of the group.