Chapter 4

Sinusoids

Most analog signals are either sinusoids, or a combination of sinusoids (or can be approximated as a combination of sinusoids). This makes combinations of sinusoids especially interesting. It is easy to add sinusoids together; pressing keys on a piano or strumming a guitar adds several sinusoids together (though they do decay, unlike the sinusoids we usually study). In this chapter, we will investigate sinusoids and see how they can be added together to model signals. The goal is to gain a better understanding of sinusoids and get ready for the Fourier transform.

4.1 Review of Geometry and Trigonometry

Greek letters are nothing to be afraid of. Mathematics uses variables, much like we do in computer programming. These variables are one letter, which allows things to be described concisely. Greek letters are often used simply because we tend to run out of our alphabetic letters. Also, there are letters that are good to avoid, like l, and o, since they look too much like the numbers 1 and 0.

Consider a right triangle (see Figure 4.1). We know, thanks to Pythagoras (or at least his school [15]) that $a^2 + b^2 = c^2$, so if we happen to know only two sides, we can calculate the third[1]. Also, with this information, we can figure out the angles θ_1 and θ_2. Since it is a right triangle, the third angle is 90 degrees by definition.

The cosine function gives us the ratio of the length of the adjacent side (a) to the length of the hypotenuse (c). In other words,

$$\cos(\theta_1) = \frac{a}{c}.$$

[1] According to [16] the Babylonians knew this, but the Greeks were first to prove it.

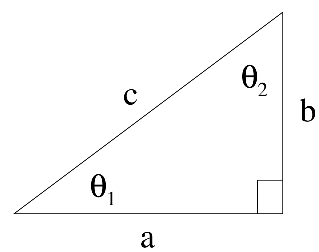

Figure 4.1: A right triangle.

The sine function gives us the ratio of the length of the opposite side (b) to the length of the hypotenuse (c). In other words,

$$\sin(\theta_1) = \frac{b}{c}.$$

The $\cos^{-1}$ and $\sin^{-1}$ functions (also called *arccos* and *arcsin*) give us the angle θ_1 from the ratio of the sides.

4.2 The Number π

It is impossible to talk about sine and cosine functions without talking about the constant π, the ratio of a circle's diameter to its circumference (see Figure 4.2). You probably remember the equation:

circumference $= 2\pi r$, where r is the circle's radius.

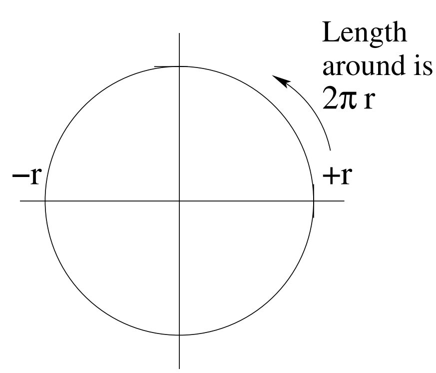

Figure 4.2: An example circle.

Pi (π) is an irrational number. No matter how many digits you calculate it out to be, someone else could go a digit further. To date, no repeating pattern has been found, so unlike rational numbers, you cannot represent π in a compact form. For example, with a rational number such as $\frac{1}{3}$, you could write it as $0.3\overline{3}$ to indicate that it repeats forever when represented as a decimal number, or at least it can be represented as one number (1) divided by another (3).

Ancient peoples had various approximations for π. The Egyptians used 256/81 for π, while the Babylonians calculated with $3\frac{1}{8}$ [17]. We will use the approximation of 3.1415927 for π.

We will specify sinusoid components with the format

$$a \times \cos(2\pi f t + \phi)$$

where a is the amplitude, f is the frequency, and ϕ is the phase angle. The variable t represents time. Sine could be used in place of cosine, if we add $\pi/2$ to the phase angle, since $\cos(\theta) = \sin(\theta + \pi/2)$.

Amplitude (a), phase shift (ϕ) (also called the phase angle), and frequency completely specify a sinusoid. *Amplitude* describes a quantity without assigning it units. *Magnitude* is also a unitless description of a quantity, with one difference: amplitude can be positive or negative, while magnitude is always positive [11]. $|x(t)|$ refers to magnitude, and as you may expect, the `abs` function in MATLAB returns the magnitude. Many people in DSP use magnitude and amplitude interchangeably, for good reason. Viewing a sinusoid component as a phasor, where we draw it in its initial position without concern about rotation with time, we would draw the phasor along the positive x-axis, then rotate it counterclockwise according to the phase angle. A negative length means we would draw the phasor along the negative x-axis, then rotate (again counterclockwise) according to the angle. However, this is exactly the same as drawing the length along the positive x-axis, and rotating it an additional 180 degrees. In other words, $-a \times \cos(2\pi f t + \phi)$ is exactly the same as $a \times \cos(2\pi f t + \phi + \pi)$. This is why we typically use a positive value for the amplitude. The main exception is for the DC component, where frequency equals zero. The phase angle also is assumed to be zero, and thus the amplitude is the only value left to determine the sign of the DC component.

We use f to represent the cyclic frequency, also called simply *frequency*, though sometimes the radian frequency (ω) is used for convenience. It relates to f with the equation $\omega = 2\pi f$. The units of f are in Hz, cycles per second, named for Heinrich Hertz (1857–1894), a German physicist whose experiments confirmed Maxwell's electromagnetic wave theories [18]. Since the unit Hz comes from a person's name, the first letter should always be capitalized.

4.3 Unit Circles

A *unit circle* is of particular interest. For a unit circle, the radius r equals 1. Therefore, the circumference, or length of the circle if it were cut and unrolled, is 2π. As a result, we can talk about an angle in terms of the length of the arc between two line segments joined at a point. We call this angle *radians* (rad for short). In Figure 4.3, angle θ is only a fraction of the circle's circumference, therefore, it is a fraction of 2π. Figure 4.4 shows the arc lengths on a unit circle for several common angles.

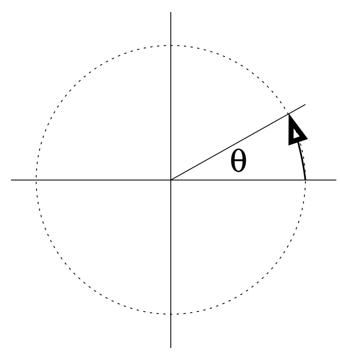

Figure 4.3: An angle specified in radians.

No doubt you are also familiar with an angle measured in degrees. When dealing with sinusoidal functions (such as sine, cosine, etc.), it is important to know whether the function expects the argument to be in degrees or radians. On a scientific calculator, you are likely to have a button labeled DRG, to allow you to specify degrees or radians, though the default is probably degrees. For MATLAB, C/C++ and Java, the arguments to the sin() and cos() functions are assumed to be radians.

To convert between degrees and radians, we recall that one complete rotation around the unit circle's edge is 2π radians, or $360°$. To convert from degrees to radians, therefore, we multiply by $2\pi/360$. To convert from radians to degrees, we multiply by $360/(2\pi)$. For example, suppose the angle shown in Figure 4.3 is $30°$,

$$\begin{aligned}(30)(2\pi/360) &= 60\pi/360 \\ &= \pi/6 \; radians.\end{aligned}$$

Converting $\pi/6$ back to degrees can be done in a similar fashion:

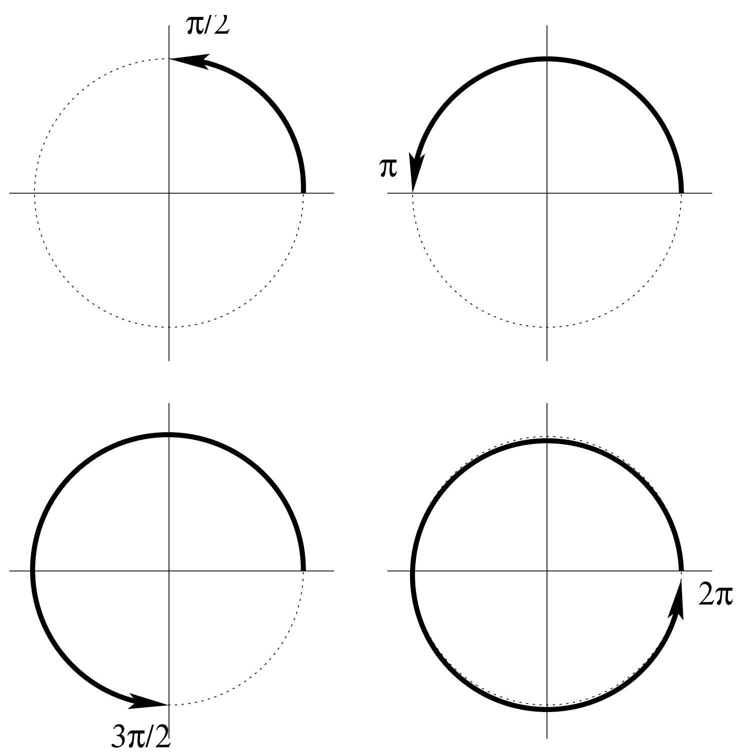

Figure 4.4: An angle specified in radians.

$$\begin{aligned}(\pi/6)(360/(2\pi)) &= 360\pi/(12\pi)\\ &= 360/12\\ &= 30^\circ.\end{aligned}$$

Quiz: What is larger, $\cos(0)$ or $\cos(1,000,000)$? At first glance, one unfamiliar with the cosine function might assume that the latter is larger, simply because the argument is larger. But since sinusoids repeat, there is no reason for this to be the case. In fact, one might remember that $\cos(0)$ is 1, and that $\cos(\textit{any number})$ always generates a value between $+1$ and -1. Therefore, $\cos(0)$ is going to be greater than or equal to $\cos(\textit{any number})$. In fact, $\cos(1,000,000) \approx 0.9368$, correct? If not, it is likely due to your calculator being in degrees mode! Since it was not specified that the angle is in degrees, one should assume that it is in radians.

Another thing to consider when dealing with sinusoid functions is that they are repetitive. The angle specified by 0 radians is the same as that specified by 2π radians. Thus, you can always add or subtract 2π to/from an angle. In fact, you can add or subtract any integer multiple of 2π to or from an angle. Sometimes this makes things easier.

4.4 Principal Value of the Phase Shift

A sinusoid stretches out infinitely in both directions. If we want to view it in terms of a time axis, we can use the positive peaks of the sinusoid to relate to our time axis. Figure 4.5 shows an example sinusoid, $\cos(2\pi 60t)$, with a frequency of 60 Hz. Therefore, the period of time before this sinusoid repeats is $\frac{1}{60}$, or 0.0167 seconds. Vertical lines show where one repetition starts and ends, at $t = -0.0167/2$ and $t = +0.0167/2$. This corresponds to arguments of $-\pi$ and $+\pi$ for the cos function, that is, let $t = (\frac{1}{60})(\frac{-1}{2})$ to make $\cos(2\pi 60t)$ equal $\cos(2\pi 60(1/60)(-1/2)) = \cos(-\pi)$.

But WHICH peak do we normally use? In Figure 4.5 it is very easy, but a sinusoid function often has a nonzero phase angle. Since it repeats every 2π, we could pick any peak that we want to classify the signal. This means that there will always be a positive peak between -π and $+\pi$, in other words, $-\pi < \phi \leq +\pi$ where ϕ is the phase shift [6].

So why not use the positive peak closest to zero? This makes a lot of sense, and we give this peak a special name: we say it is the *principal value* of the phase shift. To find this peak, we can always add/subtract 2π from the phase shift, until it is in the range we desire.

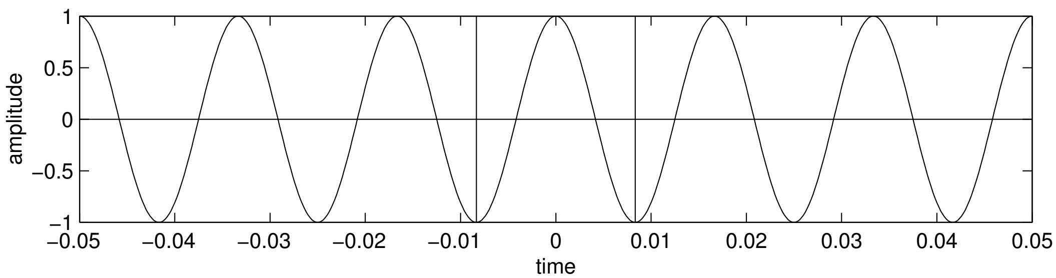

Figure 4.5: A 60 Hz sinusoid.

4.5 Amplitudes

Usually, amplitude values are positive, and for good reason. Consider two sinusoids of the same frequency, $x_1 = a \times \cos(2\pi ft + \phi_1)$ and $x_2 = -a \times \cos(2\pi ft + \phi_2)$. We know that the minimum and maximum values for the cosine function are -1 and +1. Therefore, x_1 varies between $+a$ and $-a$, and x_2 varies between $-a$ and $+a$. Given that the two sinusoids have the same frequency, they must look a lot alike. In fact, the only difference between them is their phase angles, a difference of π. That is, $\phi_2 = \phi_1 + \pi$. This makes a lot of sense considering the conversion of a complex value of $x + jy$ to polar form $r\angle\theta$: the equation $(\sqrt{x^2 + y^2})$ leads to a positive value for r. A vector with a negative length would be the same as a vector with a positive length, rotated a half-revolution, as demonstrated in Figure 4.6.

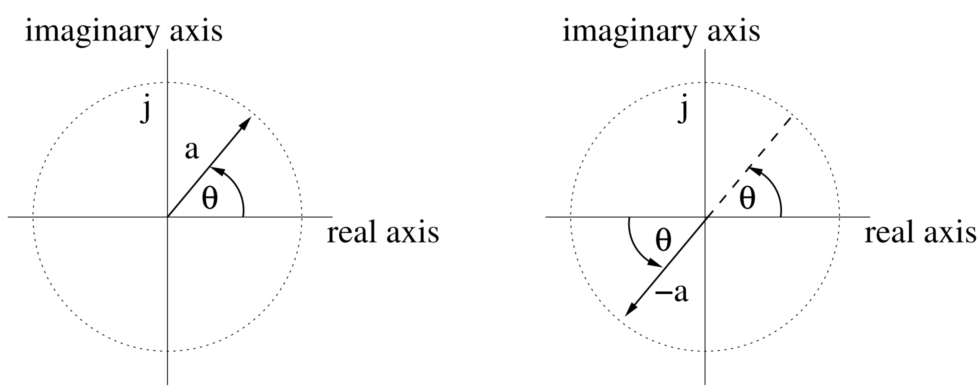

Figure 4.6: A vector of $-a$ at angle $\theta = a$ at angle $(\theta + \pi)$.

Signals are often represented as a sum of sinusoids. When these sinusoids are related, we call the signal a harmonic.

4.6 Harmonic Signals

Real-world signals can be decomposed into a set of sinusoids added together. A *harmonic* signal is composed of several sinusoids added together, each with an integer multiple of some base frequency, called the *fundamental frequency*. For example, the following signal is a harmonic:

$$x(t) = 0.4\cos(2\pi 15t + \pi/5) + 0.7\cos(2\pi 30t) + \cos(2\pi 45t - \pi/4).$$

Signal $x(t)$ is of the form

$$a_1 \times \cos(2\pi(1)f_0 t + \phi_1) + a_2 \times \cos(2\pi(2)f_0 t + \phi_2) + a_3 \times \cos(2\pi(3)f_0 t + \phi_3)$$

where, for $x(t)$, $a_1 = 0.4$, $a_2 = 0.7$, $a_3 = 1$, $\phi_1 = \pi/5$, $\phi_2 = 0$, $\phi_3 = -\pi/4$, and $f_0 = 15$ Hz. Any signal of this form is a harmonic.

If an amplitude value is 0, then the corresponding sinusoid will not contribute to the function. In other words, $0 \times \cos(anything)$ evaluates to 0. If we have a function like

$$x_2(t) = 0.1\cos(2\pi 100t - \pi/6) + 1.3\cos(2\pi 300t + \pi) + 0.5\cos(2\pi 400t + 2\pi/3)$$

then $x_2(t)$ is a harmonic with $a_1 = 0.1$, $a_2 = 0$, $a_3 = 1.3$, $a_4 = 0.5$, $\phi_1 = -\pi/6$, $\phi_2 = 0$, $\phi_3 = \pi$, $\phi_4 = 2\pi/3$, and $f_0 = 100$ Hz.

The following MATLAB code is a function to plot harmonic signals.

```
%
%  Given magnitudes, phases, and the fundamental frequency,
%    plot a harmonic signal.
%
%  Usage: plotharmonic(freq, mag, phase)
%
%  where  freq = the fundamental frequency (a single value),
%          mag = the amplitudes (a list),
%        phase = the phases (a list).
%
function plotharmonic(freq, mag, phase)

num_points = 200;   % Number of points per repetition

% Check parameters
if (size(mag) ~= size(phase))
```

```
    sprintf('Error - magnitude and phase must be same length')
end

% We want this for 2 repetitions, and num_points per rep.
step = 2/(freq*num_points);
t = 0:step:2*(1/freq);

clear x;
x = 0;
for i=1:length(mag)
    x = x + mag(i)*cos(2*pi*i*freq*t + phase(i));
end

my_title = sprintf('Harmonic signal: %d sinusoids',length(mag));
plot(t,x);
title(my_title);
xlabel('time (sec)');
ylabel('Amplitude');
```

For example, the following commands plot $x_2(t)$ from above. It is assumed that the code from above is saved with "plotharmonic.m" as the filename.

```
Mag = [ 0.1  0  1.3  0.5];
Phase = [ -pi/6  0  pi  2*pi/3];
plotharmonic(100, Mag, Phase);   % Fundamental freq. is 100 Hz
```

The resulting plot can be seen in Figure 4.7.

One additional piece of information is if there is a vertical shift of the signal. That is, what if the graph shown in Figure 4.7 was not centered around the x-axis? This can be accomplished by adding a constant to the signal. For example, if we add 100 units to the signal, it would look the same, but it would be centered around the line $y = 100$. This piece of information can be added to the above harmonic signal format, simply by including the term $a_0 \cos(2\pi(0)f_0 t + \phi_0)$. Starting the multiples of f_0 at zero means that the first component is simply a constant. There is no need to specify a ϕ_0 term, since $\cos(\phi_0)$ would just be a constant, and a_0 already specifies this. Therefore, we will assume ϕ_0 is zero. This term is often called the *DC component*, for Direct Current.

Example:

For the signals below, say which ones are harmonic and which are not. If they

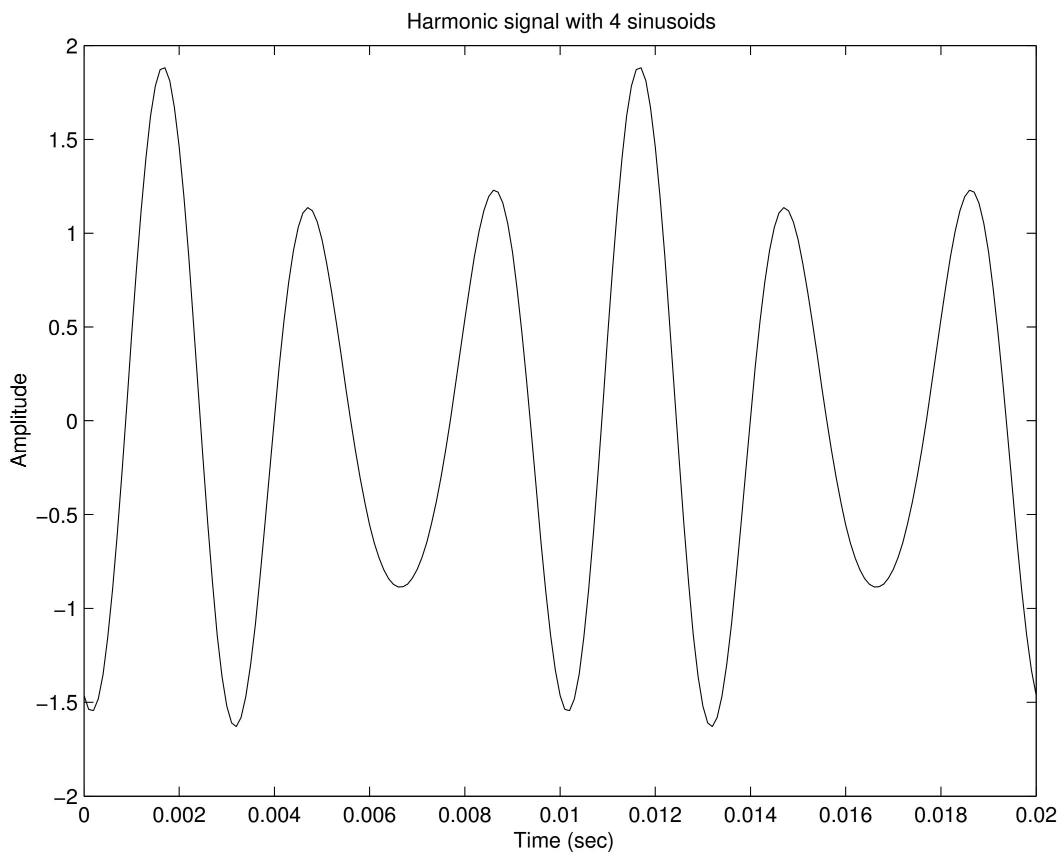

Figure 4.7: A harmonic signal.

are harmonic, say what the fundamental frequency is.

$x_1(t) = 2\cos(2\pi 7t) + 3\cos(2\pi 35t)$
$x_2(t) = 2\cos(2\pi^2 t) + 3\cos(2\pi t)$
$x_3(t) = 2\cos(2\pi 7t + \pi) + 3\cos(2\pi t - \pi/4)$

Answer:

To be harmonically related, each sinusoid must have a frequency that is an integer multiple of some fundamental frequency. That is, they should be in the following form. Remember that k is an integer.

$$x(t) = \sum_{k=1}^{N} a_k \cos(2\pi k f_0 t + \phi_k)$$

So we put the above signals in this form, and see if the k values are integers, which they must be for the sinusoids to be harmonically related. Looking at the arguments of the cosine function for x_1, let k_1 and k_2 be values of k.

$2\pi 7t = 2\pi k_1 f_0 t + \phi_k$, Let $\phi_k = 0$
$f_0 = 7/k_1$

$2\pi 35t = 2\pi k_2 f_0 t + \phi_k$, Let $\phi_k = 0$
$f_0 = 35/k_2$

Substituting in f_0, we get

$35/k_2 = 7/k_1$
$k_1/k_2 = 7/35$
$k_1/k_2 = 1/5$

So if $k_1 = 1$, then $k_2 = 5$

$f_0 = 35/k_2 = 35/5 = 7$ Hz.

Therefore, $x_1(t)$ has integer k values, and is harmonic. $x_3(t)$ is, too. $x_2(t)$ does not have integer k values (due to the frequency of π) and is, therefore, not harmonic. The third signal, x_3, is a bit tricky because the frequency components are not lined up as we may expect.$f_0 = 1$ Hz, $a_0 = 1$, $a_k = \{3, 0, 0, 0, 0, 0, 2\}$, and $\phi_k = \{-\pi/4, 0, 0, 0, 0, 0, \pi\}$.

We could consider harmonic a signal with a DC component, too. Without loss of generality, we can have index k start at 0, implying that we have a frequency component of $0 * f_0$, or 0 Hz. Since designating a value for the phase angle would only result in the amplitude a_0 being multiplied by a constant, we should set the phase angle to 0. Thus, a_0 completely specifies the DC component.

Representing a signal that has discontinuities, such as a square wave, with a sum of harmonically related sinusoids is difficult. Many terms are needed [19]. Called *Gibbs' phenomenon*, we cannot perfectly represent a discontinuous signal as a sum of sinusoids, but our approximation will have overshoot, undershoot, and ripples.

Fourier analysis works best with a *periodic* signal. We implicitly expect that the first and last sample are linked, that if we were to take one more reading of the signal, that this reading would be the same as the very first sample, or at least a value that we have seen before. Whether or not this $N+1$ value matches value 0 depends upon how many samples we have taken. Also, we assume that a sufficient number of samples has been recorded, given by the sampling period. If the signal truly is periodic, and we have a good sampling rate, and we take exactly one period's worth of samples, then the sample at $N+1$ would be equal to sample 0.

Sometimes we do not have a periodic signal. In this case, we may have a discontinuity between the last sample and the first. For example, consider the signal below, Figure 4.8:

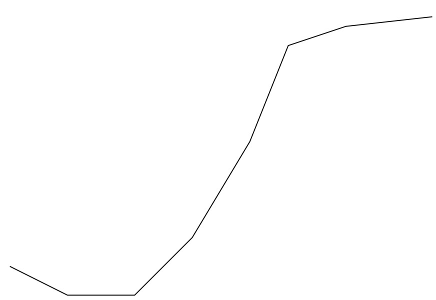

Figure 4.8: A short signal.

If we treat this signal as being periodic, then we expect that it repeats forever, and that we can copy and paste this signal to itself, such as in Figure 4.9.

Here we see that there is a discontinuity in the middle, and this is difficult for a Fourier series to handle (remember Gibb's phenomenon). Thus, we should consider the Fourier analysis to give us an *approximation* of the original signal. We would expect that the approximation is not as good around a discontinuity as it is for the rest of the signal. We call this difficulty *edge effects*.

Even with edge effects, Fourier analysis will return a good approximation to the

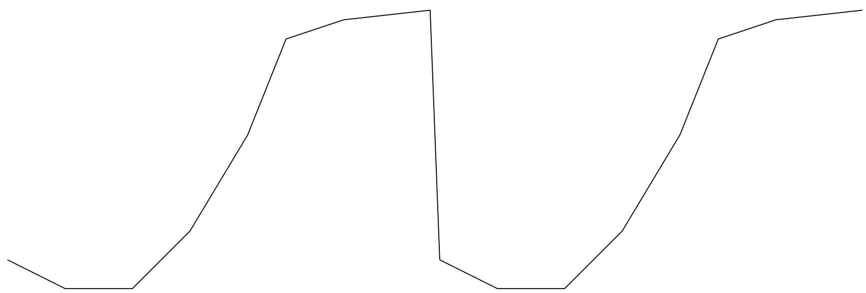

Figure 4.9: The short signal, repeated.

original signal. In fact, *any nonrandom signal can be approximated with Fourier analysis.* If we take a random signal and find the Fourier series of it, we will find that it approximates the signal for the values that we have seen. However, we may be tempted to use this to predict future values of the signal. If it is truly random, though, any matching between the signal and a prediction would be coincidence.

4.7 Representing a Digital Signal as a Sum of Sinusoids

A digital signal can be represented as a sum of sinusoids. The following three figures show how this works. We can take a digital signal and find the Fourier transform of it (the Fourier transform is covered a bit later in this book). The important thing to understand is that the Fourier transform gives us a frequency-domain version of the signal: a list of amplitudes and phase angles corresponding to a harmonically related set of sinusoids.

We explore the fundamental frequency of these sinusoids later; for the present discussion, we will ignore it. The timing of these simulated samples determines the real fundamental frequency, so we will just stick to sample numbers instead of time for the x-axis of the plots.

Refer to Figure 4.10. Here we see the original signal, a list of discrete sample points. Under it, we have a graph of a signal made by adding several sinusoids (8, to be precise). Notice how this composite signal passes through the same points as the original digital signal. If we were to read values from this composite signal with the same timing as the sampling of the digital signal, we would have the same data.

In Figure 4.10, we see that the sum of sinusoids representation is continuous. But we should be careful *not* to conclude that the underlying analog signal, which we are trying to represent as a set of discrete points, actually looks like the composite

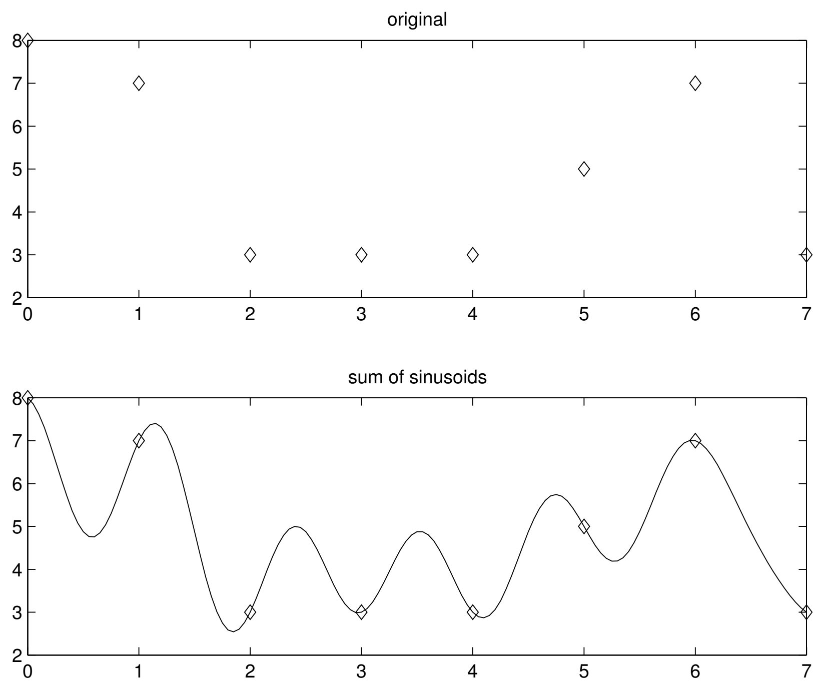

Figure 4.10: A digital signal (top) and its sum of sinusoids representation (bottom).

signal. We may not know what it looks like, and the hill we have between sample 2 and sample 3 (as well as between sample 3 and sample 4) could mislead us.

The next two graphs, Figures 4.11 and 4.12, show the sinusoids that we added together to get the composite one. Yes, the first one shown is a sinusoid of zero frequency! The asterisks show the values that would be found if the sinusoids were sampled with the same timing as the original signal. Notice how the number of sinusoids (8) returned from the Fourier transform matches the number of original points (8).

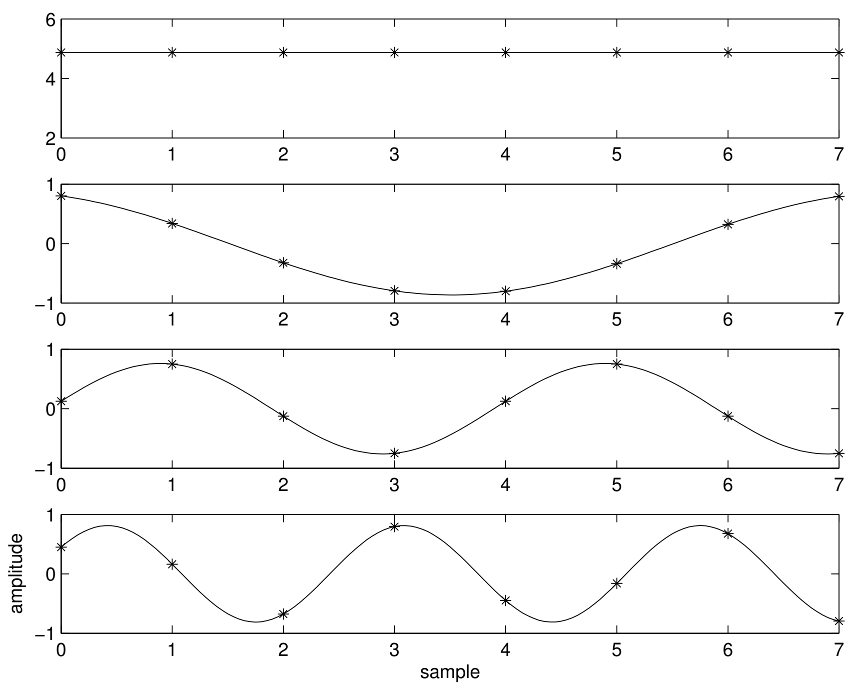

Figure 4.11: The first four sinusoids in the composite signal.

As an exercise, let's approximate the value of the second-to-last sample. In Figure 4.11, we see that sample #6 of each sinusoid has the following approximate values: 4.9, 0.3, -0.1, and 0.7. Turning our attention to Figure 4.12, we see this next-to-last sample's values are roughly: 0.4, 0.7, -0.1, and 0.3. If we add these up together, we get 7.1. Looking back at the original Figure 4.10, we see that the value for sample #6 is right around 7. In fact, the only reason we did not get exactly the same number as in the first figure was due to the error induced by judging the graphs by eye.

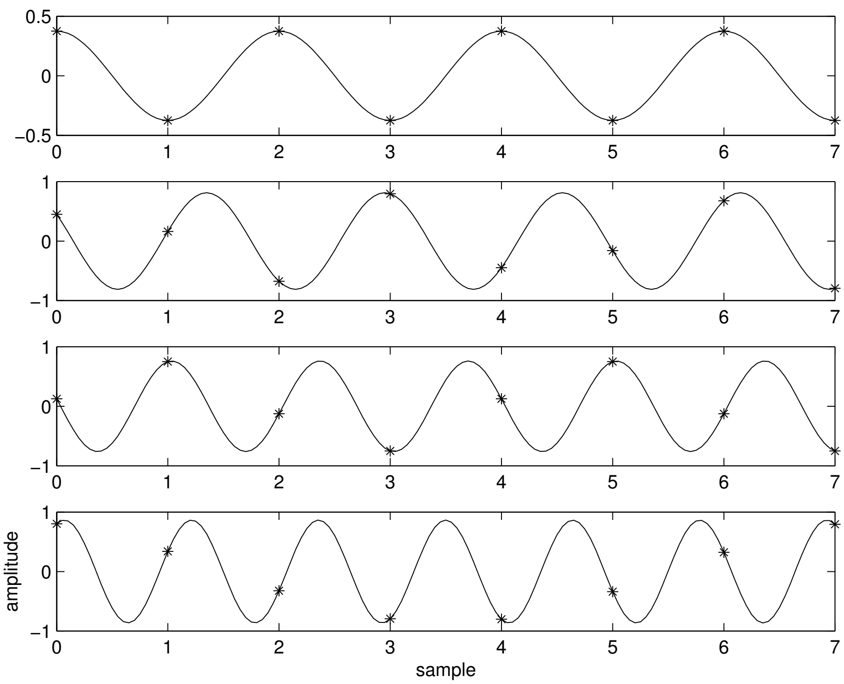

Figure 4.12: The last four sinusoids in the composite signal.

In a similar exercise, we can note the maximum values for each of the sinusoids in Figures 4.11 and 4.12. These are: 4.88, 0.86, 0.76, 0.81, 0.38, 0.81, 0.76, and 0.86. They correspond to the amplitudes of each of the sinusoids, the a_k values that we would need for a harmonic representation. The other pieces of information (besides the fundamental frequency) would be the phase angles. These are a bit more difficult to read off of a graph, but we can make approximations for them. Remembering that $\cos(0) = 1$, we notice that the value returned from the cosine function decreases as the angle gets larger, until it reaches -1 at π, then it increases until it gets to 1 again at 2π. We can use this observation to get a rough idea about the phases. Looking at the second graph of Figure 4.11, we see that the sinusoid has its maximum point right at sample 0. The maximum point would correspond to the amplitude, or more exactly $a_1 \times \cos(2\pi 1 f_0 t + \phi_1)$ when the argument to the cosine function is 0, as it would be at time 0. So we would have $2\pi 1 f_0 0 + \phi_1 = 0$, or simply $\phi_1 = 0$. Now, to find the next phase angle, ϕ_2. At sample 0 (third graph on Figure 4.11), we notice that our value would be about 0.1. Solving for ϕ_2, $a_2 \times \cos(2\pi 1 f_0 t + \phi_2)$ at time $0 \approx 0.1$. This means that $a_2 \times \cos(\phi_2) \approx 0.1$, and we already have an approximation of a_2 as 0.8. So we can solve to find:

$$0.8 \cos(\phi_2) \approx 0.1$$
$$\cos(\phi_2) \approx \frac{0.1}{0.8}$$
$$\phi_2 \approx arccos\left(\frac{0.1}{0.8}\right)$$
$$\phi_2 \approx 83^\circ$$
$$\phi_2 \approx \frac{46\pi}{100}.$$

But there could be a problem: how do we know that the sinusoid will increase at the next time step? To put this another way, we know that the cosine function returns the same value for θ as it would for $2\pi - \theta$, so which value should we use for our phase?

To solve this problem, we borrow from calculus (in a very painless way!). We have a graphical representation for the sinusoid that we are after, and we can tell just by looking at the graph if it increases or decreases. In the case of the third graph on Figure 4.11, we see that it increases. If we add a little bit to the phase angle, simulating a very small time increment, the value returned by $a_2 \times \cos((a\ little\ bit) + \phi_2)$ should be a little more than $a_2 \times \cos(\phi_2)$. Let's use this to test our value for ϕ_2.

$$0.8cos\left(\frac{46\pi}{100}\right) = 0.1003$$
$$0.8cos\left(0.01 + \frac{46\pi}{100}\right) = 0.0923$$

But this shows that the sinusoid would *decrease*, contrary to our graph. Therefore, we know that the phase angle must be our previous estimate negated, or $-46\pi/100$. We can (and should) double-check our work by applying the same trick as above.

$$0.8cos\left(\frac{-46\pi}{100}\right) = 0.1003$$
$$0.8cos\left(0.01 - \frac{46\pi}{100}\right) = 0.1082$$

We see that the next values increase, just as we observe from the graph. We can continue in like fashion for the rest of the graphs.

The program below demonstrates the sum-of-sinusoids concept. It does not require any parameters, since it makes the "original" signal randomly. This program finds the frequency-domain representation X of the random signal x, then uses the frequency-domain information to get the original signal back again (as the array *my_sum_of_sins*). We see that, when run, this program calculates the error between the original signal and the reconstructed one, an error so small that it is due only to the precision used.

```
% sum_of_sins.m
% Show a random signal as a sum of sins
%

% Make our x signal
x = round(rand(1,8)*10);
Xsize = length(x);

% Get FFT of x
X = fft(x);
Xmag = abs(X);
Xphase = angle(X);

% Show the freq-domain info as sum of sinusoids
% Find the IFFT (the hard way) part 1
n=0:Xsize-1;  %m=0:Xsize-1;
% Do the sinusoids as discrete points only
s0 = Xmag(1)*cos(2*pi*0*n/Xsize + Xphase(1))/Xsize;
s1 = Xmag(2)*cos(2*pi*1*n/Xsize + Xphase(2))/Xsize;
s2 = Xmag(3)*cos(2*pi*2*n/Xsize + Xphase(3))/Xsize;
s3 = Xmag(4)*cos(2*pi*3*n/Xsize + Xphase(4))/Xsize;
s4 = Xmag(5)*cos(2*pi*4*n/Xsize + Xphase(5))/Xsize;
s5 = Xmag(6)*cos(2*pi*5*n/Xsize + Xphase(6))/Xsize;
s6 = Xmag(7)*cos(2*pi*6*n/Xsize + Xphase(7))/Xsize;
s7 = Xmag(8)*cos(2*pi*7*n/Xsize + Xphase(8))/Xsize;
% Redo the sinusoids as smooth curves
t = 0:0.05:Xsize-1;
smooth0 = Xmag(1)*cos(2*pi*0*t/Xsize + Xphase(1))/Xsize;
smooth1 = Xmag(2)*cos(2*pi*1*t/Xsize + Xphase(2))/Xsize;
smooth2 = Xmag(3)*cos(2*pi*2*t/Xsize + Xphase(3))/Xsize;
smooth3 = Xmag(4)*cos(2*pi*3*t/Xsize + Xphase(4))/Xsize;
```

```
smooth4 = Xmag(5)*cos(2*pi*4*t/Xsize + Xphase(5))/Xsize;
smooth5 = Xmag(6)*cos(2*pi*5*t/Xsize + Xphase(6))/Xsize;
smooth6 = Xmag(7)*cos(2*pi*6*t/Xsize + Xphase(7))/Xsize;
smooth7 = Xmag(8)*cos(2*pi*7*t/Xsize + Xphase(8))/Xsize;
% Find the IFFT (the hard way) part 2
my_sum_of_sins = (s0+s1+s2+s3+s4+s5+s6+s7);
smooth_sum = smooth0 + smooth1 + smooth2 + smooth3;
smooth_sum = smooth_sum + smooth4 + smooth5 + smooth6 + smooth7;


% Show both discrete points and smooth curves together
xaxis1 = (0:length(smooth0)-1)/20;  % for 8 points
xaxis2 = (0:length(s0)-1);
figure(1);
subplot(4,1,1);  plot(xaxis1, smooth0, 'g', xaxis2, s0, 'r*');
subplot(4,1,2);  plot(xaxis1, smooth1, 'g', xaxis2, s1, 'r*');
subplot(4,1,3);  plot(xaxis1, smooth2, 'g', xaxis2, s2, 'r*');
subplot(4,1,4);  plot(xaxis1, smooth3, 'g', xaxis2, s3, 'r*');
ylabel('amplitude');
xlabel('sample');
figure(2);
subplot(4,1,1);  plot(xaxis1, smooth4, 'g', xaxis2, s4, 'r*');
subplot(4,1,2);  plot(xaxis1, smooth5, 'g', xaxis2, s5, 'r*');
subplot(4,1,3);  plot(xaxis1, smooth6, 'g', xaxis2, s6, 'r*');
subplot(4,1,4);  plot(xaxis1, smooth7, 'g', xaxis2, s7, 'r*');
ylabel('amplitude');
xlabel('sample');

% Show the original versus the reconstruction from freq-domain
figure(3);

subplot(2,1,1); plot(0:Xsize-1, x, 'bd');
title('original');
min_y = floor(min(smooth_sum));
max_y = ceil(max(smooth_sum));
axis([0, Xsize-1, min_y, max_y]); % Make the axes look nice
xaxis1 = (0:length(smooth_sum)-1)/20;
xaxis2 = (0:length(s0)-1);
subplot(2,1,2);
```

```
plot(xaxis1, smooth_sum, 'b', xaxis2, my_sum_of_sins, 'rd');
title('sum of sinusoids');
Error = my_sum_of_sins - x
```

We assume that the signal contains only real information. In general, this is not the case. However, since we started with a signal of all real values, we will end up with a signal of all real values, since we do not change the information, we only transform it from the time-domain to the frequency-domain, then transform it back. The Fourier transform results in a complex signal, though the original signal was real. To reconcile this, we simply treat the original real signal as if it were complex, but the imaginary part is zero. The signal returned from the inverse-transform must also have an imaginary part of zero for all values. The program performs the inverse-transform in a sense, but it does not take the complex part into consideration, as it would have to do if the original signal were complex. In other words, the program treats each sinusoid as if it were $X_{mag}[k]\cos(2\pi k f_0 t + X_{angle}[k])$, when in fact each sinusoid is (from Euler's formula) $X_{mag}[k]\cos(2\pi k f_0 t + X_{angle}[k]) + jX_{mag}[k]\sin(2\pi k f_0 t + X_{angle}[k])$. We are able to get away with this simplification since we know in advance that the complex parts, $j\sin(\circ)$, must cancel each other out.

4.8 Spectrum

A *spectrum* is the name given to the frequency plot of a signal. The spectrum is a graph of the frequencies that make up a signal, with their amplitudes shown in relation to each other. A separate graph shows the phases for each sinusoid.

Previously, in Figure 4.10, we saw that a digital signal can be represented by a sum of sinusoids. These sinusoids can then be graphed individually, as shown in Figures 4.11 and 4.12. Sometimes this sinusoid information provides insight into the signal. For example, if the original signal is music, then the sinusoids tell us about the instruments that are being played. If the original signal was of a person talking, with a high-pitched extraneous sound such as a train whistle, we might be able to remove the higher-frequency whistle.

Plotting the information given to us by the sinusoids results in a spectrum. Continuing with the example from the last section, we see that one piece of information we are missing is the fundamental frequency. As stated before, we need to know how much time there is between samples before we can know the frequencies. But we will do our best without this. We see that the sinusoids repeat an integral amount in each graph, from 0 repetitions (0 frequency) up to 7 repetitions. That is, the sinusoids repeat with such a frequency that sample #8, if plotted, would be the same as sample

#0. As the number of repetitions gets larger, so does the corresponding frequency. Suppose, for the sake of argument, that the original signal was sampled at a rate of 125 milliseconds per sample, so we read each sample 125 milliseconds after the last. The second graph, on Figure 4.11, shows that the sinusoid repeats once every 1000 milliseconds, or once per second (it takes another 125 milliseconds before it gets to sample #8, which is identical to sample #0). Thus, it has a frequency of 1 Hz. Similarly, we see that the next graph has a frequency of 2 Hz, then 3 Hz for the next graph, etc., up to 7 Hz for the last graph of Figure 4.12. For the remainder of this chapter, we assume that we have this fundamental frequency of 1 Hz. Recalling our amplitudes {4.88, 0.86, 0.76, 0.81, 0.38, 0.81, 0.76, 0.86} from the previous section, and getting the corresponding phases in degrees $\{0, 22, -81, -56, 0, 56, 81, -22\}$, we can plot this information as follows, Figure 4.13. In the top part of the figure, we see a frequency magnitude plot, since it shows the positive amplitudes versus the frequencies. The bottom part of Figure 4.13 shows a frequency phase plot, since it shows the phase angles for each corresponding sinusoid frequency.

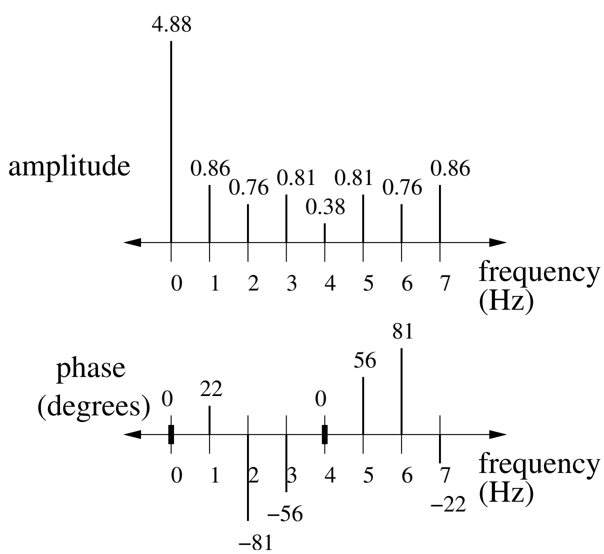

Figure 4.13: Frequency magnitude spectrum and phase angles.

One should observe the pattern of the spectrum: for the frequency magnitude plot, we see a mirror image centered around 4 Hz. On the phase plot, again the pattern centers around 4 Hz, but we have a flipped as well as mirrored image.

So far we have seen how to get the spectrum from the amplitudes and phases returned by the Fourier transform. To find the spectrum of an abstract signal, such as a mathematical model, use the inverse Euler's formula. Here we have only an introduction; this topic is covered extensively in Chapter 7, "The Number e." For example, to find and plot the magnitude spectrum of $x(t) = 2 + 2\cos(2\pi(200)t)$, we put it in terms of cosine functions.

$$\begin{aligned} x(t) &= 2 + 2\cos(2\pi 200t) \\ &= 2\cos(0) + 2\cos(2\pi 200t) \end{aligned}$$

Next, use inverse Euler's formula:

$$\cos(\phi) = (e^{j\phi} + e^{-j\phi})/2.$$

Notice how this results in two frequency components: one at the frequency given, and one at the negative frequency. For the spectrum, this implies that the range of frequencies will be different from what we saw with the previous example. Actually, the information stays the same, only our spectral plot has different, but equivalent, frequency values. This is explained in Chapter 6, "The Fourier Transform."

$$\begin{aligned} x(t) &= 2(e^{j0} + e^{-j0})/2 + 2(e^{j2\pi(200)t} + e^{-j2\pi(200)t})/2 \\ &= 2(1+1)/2 + (2/2)(e^{j2\pi(200)t} + e^{-j2\pi(200)t}) \\ &= 2 + e^{j2\pi(200)t} + e^{-j2\pi(200)t} \end{aligned}$$

This results in a magnitude of 2 at frequency 0 Hz, and a magnitude of 1 at frequencies 200 Hz and -200 Hz. This is shown in the magnitude plot, Figure 4.14.

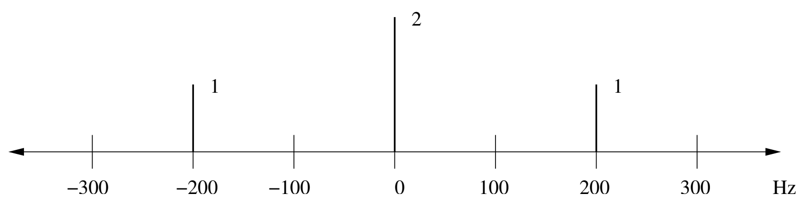

Figure 4.14: Spectrum plot: magnitude of $x(t) = 2 + 2\cos(2\pi(200)t)$.

Example:
For the signal $x(t)$ below,

$$x(t) = 4\ \cos(2\pi 100t + 3\pi/4) + 2\ \cos(2\pi 200t) + 5\ \cos(2\pi 300t - 3\pi/5)$$

draw the magnitude spectrum and the phase spectrum of this signal.

Answer:

$$
\begin{aligned}
x(t) &= \frac{4}{2}e^{j(2\pi 100t+3\pi/4)} + \frac{4}{2}e^{-j(2\pi 100t+3\pi/4)} \\
&+ \frac{2}{2}e^{j(2\pi 200t)} + \frac{2}{2}e^{-j(2\pi 200t)} \\
&+ \frac{5}{2}e^{j(2\pi 300t-3\pi/5)} + \frac{5}{2}e^{-j(2\pi 300t-3\pi/5)}
\end{aligned}
$$

$$
\begin{aligned}
x(t) &= 2\ e^{j2\pi 100t}e^{j3\pi/4} + 2\ e^{-j2\pi 100t}e^{-j3\pi/4} \\
&+ e^{j2\pi 200t} + e^{-j2\pi 200t} \\
&+ 2.5\ e^{j2\pi 300t}e^{-j3\pi/5} + 2.5\ e^{-j2\pi 300t}e^{j3\pi/5}
\end{aligned}
$$

Pulling the frequencies out $\{\ 100, -100, 200, -200, 300, -300\ \}$, and marking the corresponding magnitudes $\{2, 2, 1, 1, 2.5, 2.5\}$, gives us the graph of Figure 4.15. For each frequency, we also note the phases, $\{\ 3\pi/4, -3\pi/4, 0, 0, -3\pi/5, 3\pi/5\}$. The phase angles are shown in Figure 4.16.

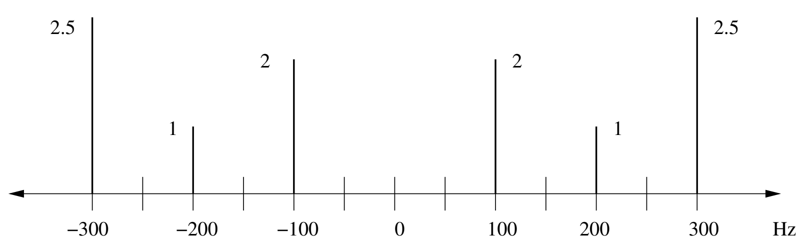

Figure 4.15: Spectrum plot: magnitudes.

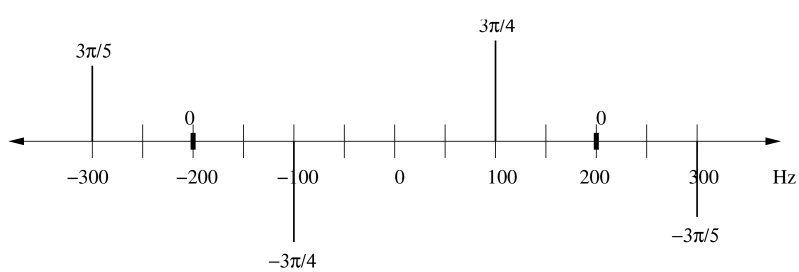

Figure 4.16: Spectrum plot: phase angles.

4.9 Summary

Real-world signals have sinusoid components, unless they are truly random, in which case they can still be approximated with a sum of sinusoids. Since all analog signals can be represented (or at least approximated) with a sum of sinusoids, we can view a signal in terms of the frequencies that comprise it. Given a sum-of-sinusoids equation for a signal, we use the inverse Euler's formula to convert to a sum of complex exponentials, such as $(a_1/2)e^{j2\pi f_1 t}e^{j\phi_1} + (a_1/2)e^{j2\pi(-f_1)t}e^{j(-\phi_1)}$. From this form, we can make a plot of the amplitudes ($a_1/2$, $a_1/2$) versus the frequencies (f_1, $-f_1$), as well as a plot of the phase angles versus the frequencies. Together, we call this information the spectrum, a view of the signal in the frequency-domain.

We will see how to get the frequency-domain information from a signal of only time-domain points using the Fourier transform, in Chapter 6.

4.10 Review Questions

1. What are the minimum and maximum values for $x(t)$?

$$x(t) = 3\cos(2\pi 100t + \pi/6)$$

2. Given $\sin(\theta) = \cos(\theta - \pi/2)$, if $x(t) = 3\sin(2\pi 100t + \pi/6)$, what is this function in terms of cos?

3. The analog signal, $x(t)$, is:

$$x(t) = 3\cos(2\pi 2000t + \pi/4) + 2\cos(2\pi 5000t) + \cos(2\pi 11000t - \pi/7)$$

   a. plot each frequency component (in the time-domain) separately
   b. graph this function (in the time-domain)
   c. represent $x(t)$ in terms of a fundamental frequency, amplitudes and phases.

4. The analog signal, $x(t)$, is:

$$x(t) = 3\cos(2\pi 3000t) + \cos(2\pi 1000t + \pi/3) + 4\cos(2\pi 5000t - \pi/3)$$

   Represent $x(t)$ in terms of a fundamental frequency, amplitudes and phases.

5. What is the difference between amplitude and magnitude? What does $|x(t)|$ mean?

6. DC component specifies the "shift of the signal" __________. ($f = 0$)

Phase-shift specifies shift of signal ____________. (ϕ)

7. What is another name for a plot of the frequency information of a signal?
8. Which signal, a square wave or a triangular wave, is easier to approximate with a series of harmonically related sinusoids? Why?
9. Why are amplitudes always positive?
10. Since $x(t)$ below is in the form $a_0 + \sum_{k=1}^{N} a_k \cos(2\pi(k)f_0 t + \phi_k)$, what are the values for N, a_k, ϕ_k, and f_0 ? What special name do we give this signal $x(t)$?

$$x(t) = 4 + 3\cos(2\pi(1)20t + \pi/2) + 7\cos(2\pi(2)20t - 2\pi/3)$$