Table of Contents

How to use this series of documents

"Digital Signal Processing Foundations" provides a gentle introduction to the world of DSP. This series of documents deals with topics relevant to DSP and they provide links to online video content in an effort to make some perhaps tricky concepts easier to understand for the reader.

If you are completely new to DSP then I'd recommend you take a look at the foundations document first. After reading this document you may wish to review frequency response in "Digital Filters – A practical guide".

I believe that the approach of integrating text with video takes advantage of the unique visualisations that video material has to offer with the more in-depth detail and speed of review that text-based material does. Most of the video material relates to my own youtube channel, which at the time of writing (2023) had over 18,000 subscribers and 3.7 million views.

I also provide Octave/Matlab code examples throughout the document and I'd encourage anyone who wants to develop practical DSP skills to download Octave, which is available free of charge, and implement your ideas.

My intention is to continue this series so as to deal with the major elements of DSP, such as convolution, correlation, the Z-Transform and so on. I'll post updates to this series at http://www.pzdsp.com/docs if you want to check out any new additions.

If you find any of this work useful I'd be most grateful if you could cite the relevant resource when appropriate to provide recognition.

Regards,

David

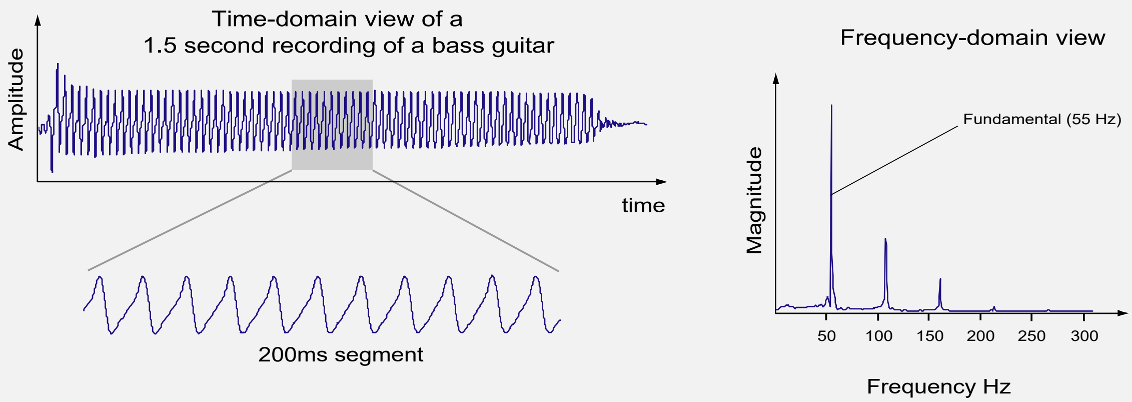
An introduction to the frequency-domain

When someone plays the guitar different sounds are created because the guitar strings vibrate or oscillate at different frequencies. A similar effect can be heard if you stretch an elastic band between your fingers and pluck it and you'd notice that changing either the length or the tension of the band would alter the frequency of the sound since this causes the band to vibrate at a different rate or frequency.

When something is oscillating a repeating pattern is being produced over time. This can be seen with a vibrating elastic band as it moves backwards and forwards through its initial position.

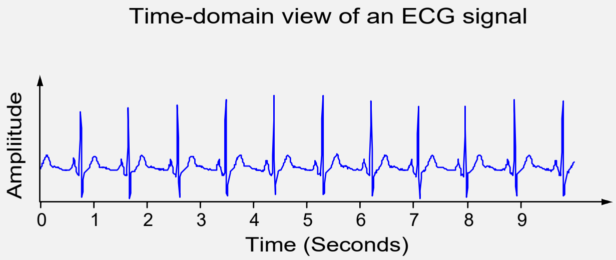
The repeating nature associated with the movement of a guitar string can also be seen in a plot of the audio signal it produces, as shown below, where the amplitude of a bass guitar audio signal is seen to move up and down over time as the strings vibrate. You should note that the rate of oscillation of the string is the same as the rate of oscillation of the audio signal since it is the string vibrations that cause pressure variations in the air which we perceive as sound (The audio recording of the bass guitar signal shown above can be downloaded from pzdsp.com/sig1). The change in air pressure can also be recorded by a microphone and stored on a computer as a discrete signal i.e. a sequence of numbers that were obtained by measuring the sound pressure at regular time intervals.

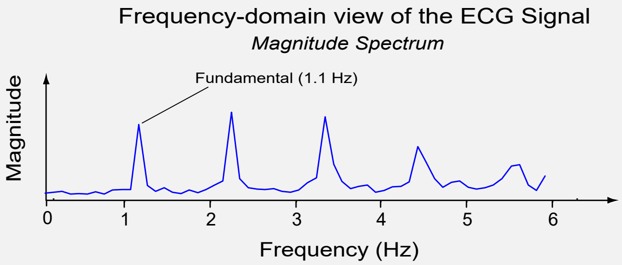
The frequency-domain representation of a signal is a convenient way of showing the oscillation rate associated with a signal, as explained in the following paragraph.



From the figure above, the sound pressure oscillates after the initial 'attack' or transient component at the start of the signal. This plot of pressure variation over time is referred to as a time-domain plot and by looking closely at this plot you can see that the time to complete one cycle of an oscillation is about 1.82 milliseconds (approx. 11 cycles over a 200ms segment). In other words, the cycle is repeating about 55 times every second. To the right of the time-domain plot is a plot of the *magnitude spectrum* which is a frequency-domain representation that can be used to quickly determine the rate of oscillations in time-domain signals. The three relatively large 'spikes' shown in the magnitude spectrum represent the fundamental frequency (55 Hz) and the first two harmonics (110 Hz and 165 Hz). You should notice that you can tell the rate of oscillation (55 Hz) quite easily when you look at the signal in the frequency-domain; much more quickly and easily than by analysing the period of the time-domain signal.

This type of repeating pattern doesn't just happen with audio signals and it can be observed in many signals, including those from our heart. Your heart will beat at particular rate, or frequency, depending on what you are doing and your heart rate will increase if you go for a run or cycle. Engineers and scientists (and musicians and doctors!) are often analysing the repeating nature of signals and the frequency-domain view of a signal shows the frequency of the repeating patterns in a convenient graph.





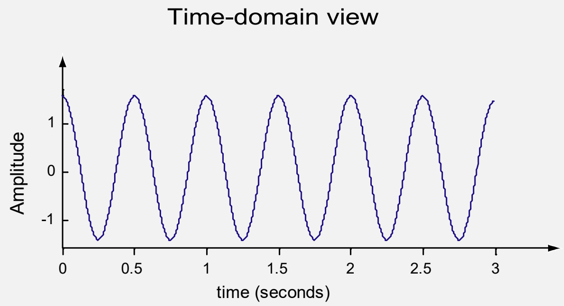
The frequency-domain view of a signal provides another way of analysing a signal which can provide valuable insight into a signals' behaviour. I find it useful to relate this to the way an architect has different drawings of a building depending on who she is dealing with: A client would find it easier to visualise what the building would look like by examining a 3-D view of the building; while a builder would require detailed plans in order to construct the building. Both sets of drawings are representations of the same building and both have their uses. It's the same with the time-domain and frequency-domain views of signals – both represent the same signal and both can be very useful when analysing signals. Here's a link to a video which demonstrates the benefit of both the time-domain view and frequency-domain view of a signal pzdsp.com/vid12.

Frequency-domain graphs of signals are very easy to create using software tools like Octave and Matlab and they make use of Fourier analysis techniques to extract frequency information from a time-domain signal (more on this later!). The basic principle behind all of the Fourier analysis techniques is that any signal can be broken down into a set of sinusoidal signals and this concept is explored further in the next couple of subsections.

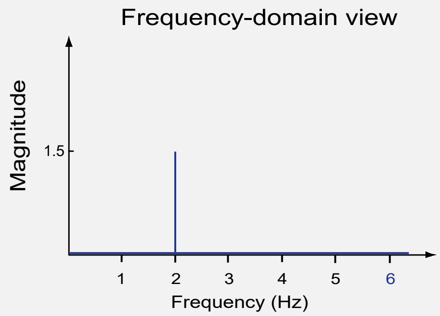
What are sinusoidal waveforms?

A sinusoidal waveform that oscillates smoothly over time (see the plot below) and is associated with many signals that occur in nature. For example, when you whistle you create pressure variations in the air which have a sinusoidal shape or of you were to allow an object attached to the end of a spring bounce up and down then the motion of the object would also be sinusoidal (see pzdsp.com/vid13). Even more interestingly it turns out that sinusoidal waveforms are a fundamental building block of any signal so it's worth spending some time getting used to what they look like and how they can be represented mathematically. This fact was shown mathematically by a French mathematician called Jean Baptiste Joseph Fourier (1768-1830).

There are three features of sinusoidal waveforms that you'll need to be comfortable with to fully appreciate Fourier analysis: *frequency*, *amplitude* and *phase offset*.



A sinusoidal waveform of amplitude 1.5 and frequency 2 Hz



The figure above shows a time-domain plot of a cosine waveform to the left and its corresponding **magnitude spectrum** to the right. From the time-domain view notice that the sinusoids amplitude oscillates between 1.5 and -1.5 which means that the amplitude of the sinusoid is 1.5. You'll notice that the sinusoid is repeating every 0.5 seconds, in other words it has a period of 0.5 seconds, which means that it has a frequency of 2 Hz. I'd recommend you check out the interactive animation at pzdsp.com/sinusoids in order to get a clearer idea about these parameters.

The frequency-domain plot of the sinusoid above shows a single 'spike' at a frequency of 2 Hz. Anytime you have a time-domain plot of a single sinusoid you will observe a single 'spike' in the frequency-domain and the position of the 'spike' on the frequency axis corresponds to the frequency of the sinusoid. The magnitude (height) of the 'spike' is proportional to the amplitude of the sinusoid. You'll see examples of signals with more than one sinusoid present in the next section.

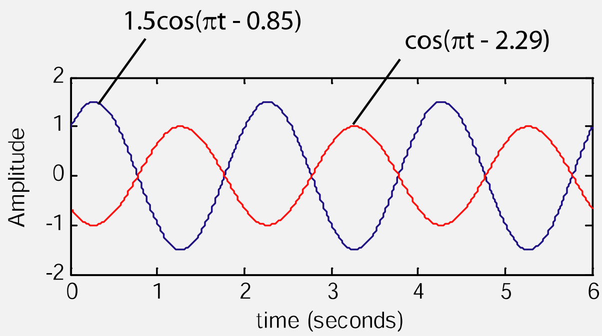
Octave code to create a plot of a sinusoid:

```
A = 1.5;
f = 2;
phi = 0;
duration = 1; %1 second
T = 1/f;
t=0:T/100:duration;
x = A*cos(2*pi*f*t + phi);
plot(t,x)
xlabel('time (seconds)')
ylabel('Amplitude')
```

Before we look at the phase associated with this sinusoid lets first take a look at a mathematical function often used to represent a sinusoid which is shown below:

$$x(t) = A\cos(2\pi f t + \varphi)$$

The A parameter specifies the amplitude of the sinusoid; f specifies the frequency and φ (Greek letter phi) parameter specifies the phase offset (also referred to as the initial phase or phase). The t variable represents time and the mathematical expression is evaluated for a range of values of t in order to create a time-domain signal. So, if you wanted to recreate the plot of the sinusoid shown above you'd substitute A with 1.5, f with 2 and φ with 0 to give $x(t) = 1.5\mathbf{cos}(4\pi t)$, and then you could evaluate this for a number of values of t before finally plotting your graph of $x(t)$ against time.



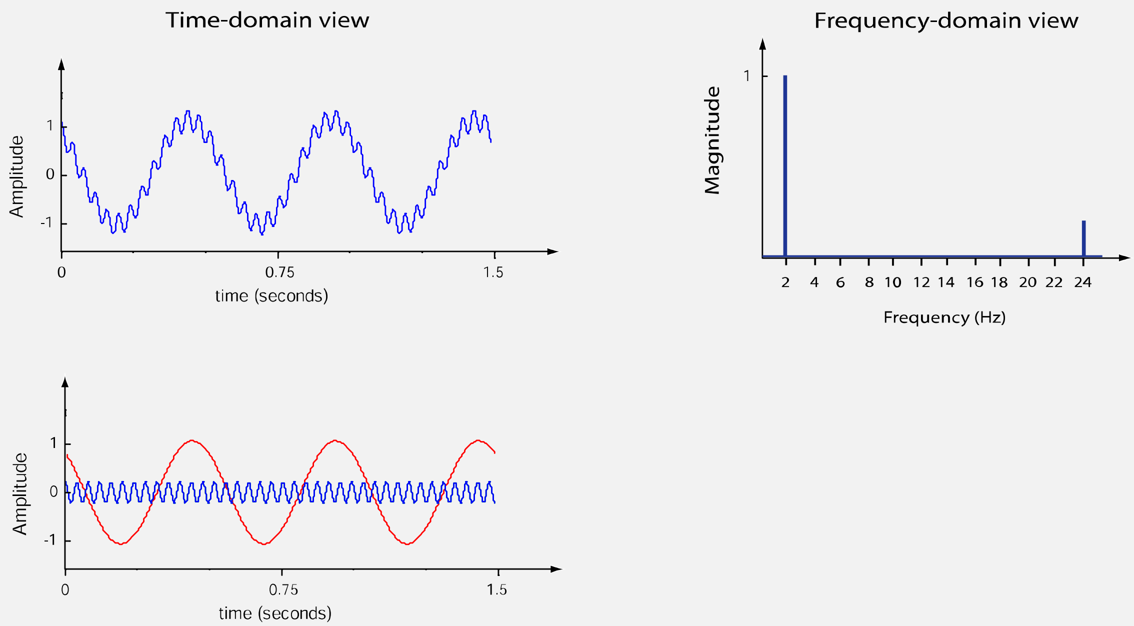
You should notice that when the phase value is zero that the waveform will be a maximum when t=0 and every period of the waveform after that. Changing the phase will change the times when the maximum of the sinusoid will occur. You should try this out for yourself using the code above and you should also observe that adding 2π to any phase offset value you try out will produce the exact same waveform. For example, the waveform produced when the phase offset is set to 1.4 will be the same as the waveform produced when the phase is set to 1.4+2π, or 1.4+4π, or even 1.4-2π for that matter. In fact, you will find that for any integer k the following relationship holds:

$$A\cos(2\pi f t + \varphi) = A\cos(2\pi f t + \varphi + k2\pi)$$

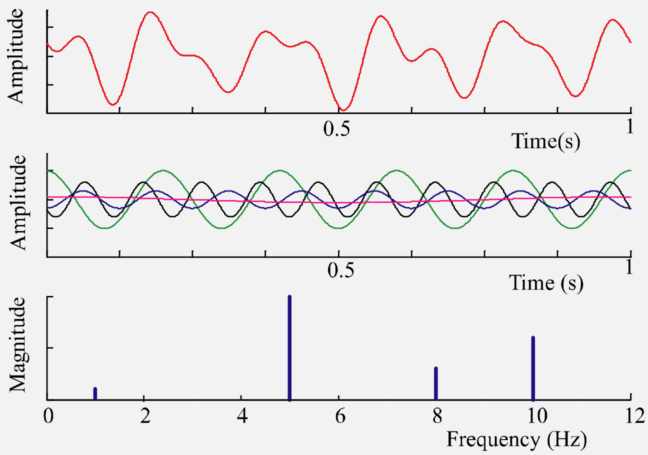
All signals can be decomposed into sinusoidal waveforms

The French mathematician Jean-Baptiste Joseph Fourier showed that any signal can be recreated by adding sinusoidal signals together. (See pzdsp.com/vid14 and pzdsp.com/vid15 for video tutorials/demonstrations on this concept).

The frequency-domain view of a signal provides a way to visualise the sinusoids that make up a signal i.e. the sinusoids that when added together reproduce the original signal. The magnitude spectrum shows the amplitudes of the various sinusoids which make up a signal, while the phase spectrum shows the phases of the sinusoids which make up a signal.



The figure above shows a waveform (top) which is a plot of the time-domain signal produced when the two sinusoids shown below it are added together. The frequency-domain view of this signal contains two spikes; the spike at 2 Hz is larger than the one at 24 Hz because the 2 Hz sinusoid is larger (5 times larger) than the 24 Hz sinusoid.



The figure to the right shows the magnitude spectrum of a signal in the bottom plot; with the time-domain view of the same signal shown in the top plot. Each of the 'spikes' in the magnitude spectrum represents a sinusoid (there are 4 in total indicating the presence of 4 sinusoids in the signal; in other words the signal could be reproduced by adding four sinusoids together). Each of the four sinusoids, which when summed together produce the time-domain signal shown in the top plot, are shown in the middle plot. The green sinusoid has 5 cycles over the one second duration of the segment shown and therefore has a frequency of 5 Hz; it has the largest amplitude, as can also be seen in the corresponding magnitude spectrum plot where the 'spike' shown at 5 Hz is the largest. It can also be seen in the magnitude spectrum that the 'spike' at 8 Hz is less than half the height of the 5 Hz spike; this can also be seen in the middle plot whereby the sinusoid with 8 cycles in one second has an amplitude of less than a half the amplitude of the 5 Hz sinusoid.

The phase values for each of the sinusoids present in the signal are 0, 0, 3.14, 2.13 radians for the 1, 5, 8, and 10 Hz components. These phase values are phase shifts relative to cosine waveforms. A plot of the phase spectrum shows the phase values plotted against frequency in a similar way to the magnitude spectrum showing the magnitude values plotted against frequency.

If you would like to see a practical application of the frequency-domain then take a look at pzdsp.com/vid12.

A note on the duration of a sinusoidal waveform

From the mathematical description of a sinusoid a sinusoidal waveform exists for all instances of time. In this document I show plots of sinusoidal segments which have a finite duration and you'll notice that I still refer to these plots of sinusoidal segments as sinusoids, which is, strictly speaking, incorrect but makes the document a bit easier to read.

A note on Fourier analysis

Fourier transforms are mathematical techniques which determine the sinusoidal parameters (amplitude, frequency and phase) of the sinusoidal waveforms that are present in a mathematical function. When working with ‘real-world’ data (as opposed to mathematical functions) the discrete Fourier transform (DFT) is used determine the frequency-domain characteristics of this ‘real world’ data/signals. A detailed description of how the DFT works is provided in the document entitled “The Discrete Fourier Transform - A practical approach” available at https://pzdsp.com/docs.

This document doesn’t provide examples of Fourier transforms and the interested reader should explore other resources for further insight. From my experience DSP practitioners will gain a more valuable insight from understanding the DFT first as it can be used on practical signals. It can be easy to get so caught up in the mathematics of the Fourier transform that you can forget what the purpose of it is!

How to use Octave/Matlab’s fft function

The fft function can be used to determine the amplitude, frequencies and phases of the sinusoids that a signal is comprised of and is frequently used to obtain a frequency-domain plot of a signal. In this section I’ll explain how to use the fft function without getting into detail on its inner workings. You should note that the fft function is an implementation of the Discrete Fourier Transform algorithm which is described in detail in the document “The Discrete Fourier Transform - A practical approach” available at https://pzdsp.com/docs.

In this section I’ll first show how to create a frequency-domain plot of the bass guitar signal used in the Introduction, then I’ll provide another example which provides more insight on how to use the fft to analyse a signal which is based on the popular video on the subject that I created in 2012 pzdsp.com/vid16.

Using Octave/Matlab’s fft function to analyse a bass guitar signal

The following code can be used to load in an audio signal and plot its frequency content. The audio file in the example can be downloaded from pzdsp.com/sig1 and you should make sure the audio file is stored/saved in the ‘present working directory’ – this can be determined by typing pwd at the command line.

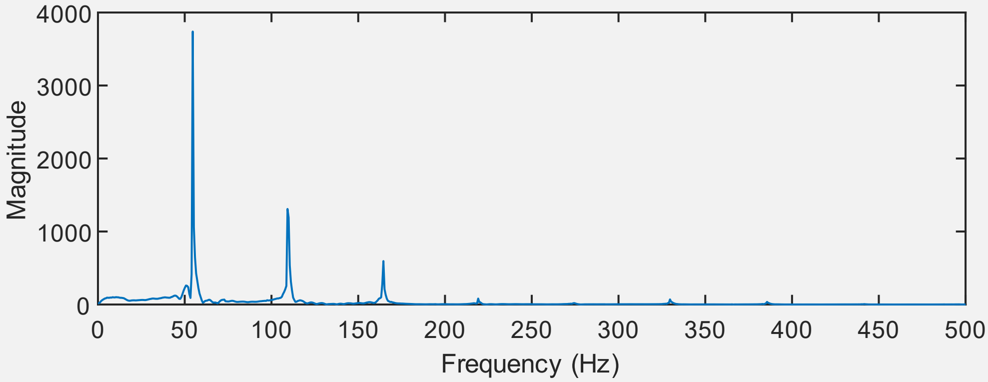
```
>> [b fs]= audioread('bass_note.wav');  % the variable b
contains the audio samples. The audioread function also
returns the sampling rate,fs, which is 44100 in this case
>> B = fft(b); % the fft returns 67822 complex numbers
which are stored in the array variable B. Note that, by
convention, capital letters are used to store frequency-
domain information while lowercase are used for time-
domain. There are 67822 samples in the time-domain signal
b. The fft function returns the same number of values as
are in the signal being analysed i.e. the time-domain
signal b in this case.
>> B_mags = abs(B); % the abs function determines the
magnitudes of the 67822 complex numbers.
```

The second line in the code above is the one that does all the hard work. The fft function analyses the time-domain signal b (as described in detail in the next section) to determine the magnitudes and phases of the sinusoids required to reproduce the time-domain signal b.

One of the most common ways to visually analyse the frequency content of signal is to plot the magnitudes of the values returned by the fft function, which provides a plot of the magnitude spectrum. You should note that there are numerous ways to plot the magnitude spectrum and the video available at pzdsp.com/vid17 provides a detailed explanation on how to do so. The following code shows one common method of plotting the magnitude spectrum against frequency using units of hertz, where the xlim([0 500]) command limits the range of frequencies being displayed to be from 0 to 500 Hz.

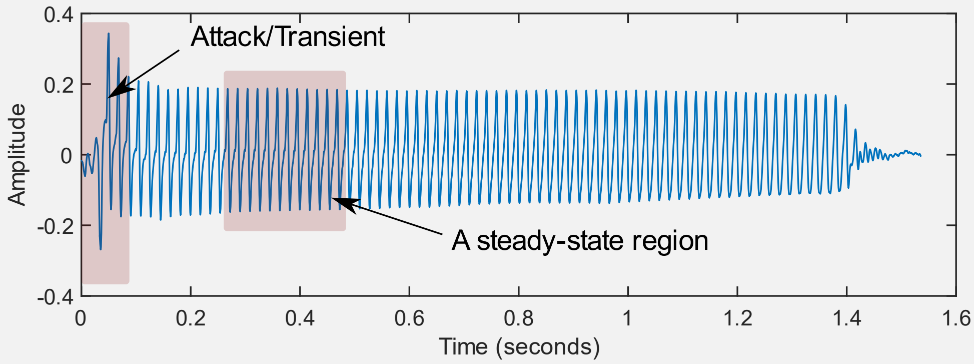
```
>> plot([0:length(b)-1]/length(b)*fs , B_mags)
>> xlim([0 500]); %limit the range of frequencies to be
from 0 to 500 Hz
>> xlabel('Frequency (Hz) '); ylabel('Magnitude');
```

A more detailed explanation of this code can be found from pzdsp.com/vid17. This code will produce the following plot in which the fundamental and first two harmonics of the guitar note can be clearly seen. There is a fundamental frequency component at about 55Hz, with strongly present harmonics at 110 Hz and 155 Hz, as indicated by the large spikes at these frequencies



The time-domain view of this signal can be plotted using the following code, and you can see that the signal contains a sharp attack/transient element from when the bass guitar string was plucked.

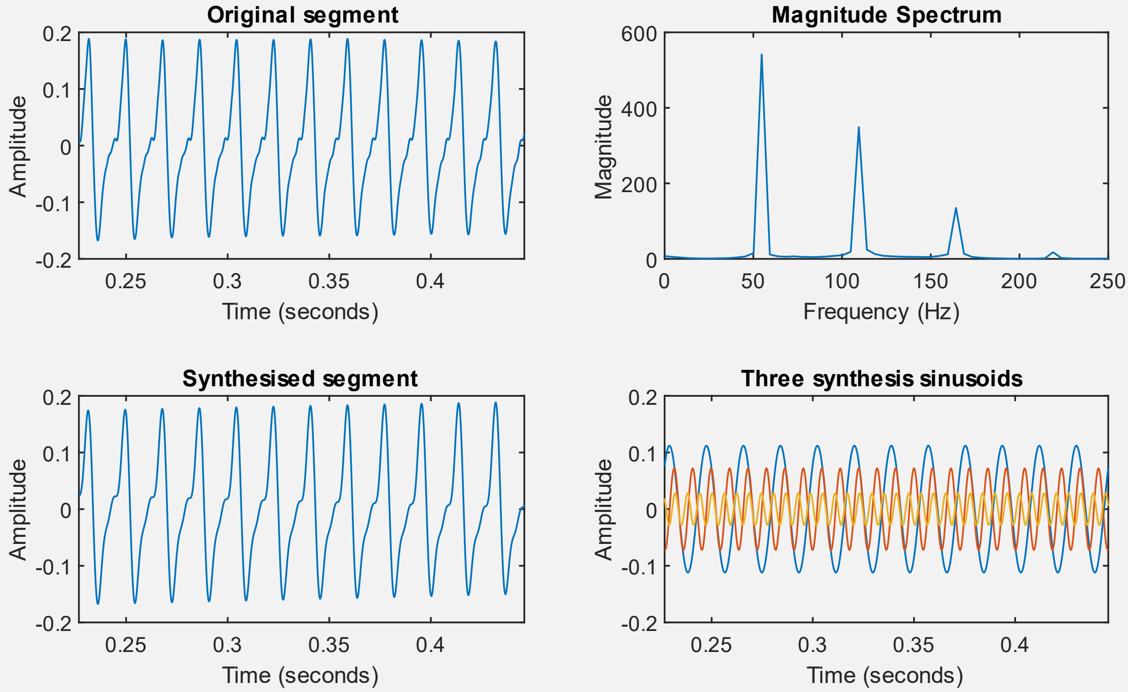
```
>> [b fs]= audioread('bass_note.wav');
>> t = [0:length(b)-1]*1/fs;
>> plot(t,b)
>> xlabel('Time (seconds)'); ylabel('Amplitude')
```



The figure above also highlights a steady-state region in which the signal is reasonably stationary. This steady-state/stationary segment can be reproduced reasonably well by adding just three sinusoidal components, as shown by the code and plots below. Note that the code doesn't show how the amplitudes, frequencies and phases of the three sinusoidal components are determined.

```
>> [ip fs]= audioread('bass_note.wav');
>> N = 9670;
```

```
>> stationary_seg = ip(10000:10000+N-1);
>> t = [0:N-1]/fs; t_offset = 10000/fs;
>> subplot(2,2,1); plot(t+t_offset, stationary_seg)
>> title('Original segment')
>> xlabel('Time (seconds)'); ylabel('Amplitude');
>> subplot(2,2,2);
>> plot([0:N-1]/N*fs , abs(fft(stationary_seg)))
>> xlim([0 250]); %limit frequencies from 0 to 250 Hz
>> xlabel('Frequency (Hz) '); ylabel('Magnitude');
>> title('Magnitude Spectrum')
>> fundamental = 0.112*cos(2*pi*54.7*t-0.82);
>> harmonic1 = 0.072*cos(2*pi*109.7*t+3);
>> harmonic2 = 0.028*cos(2*pi*164.6*t + 0.83);
>> synth_sig =  fundamental + harmonic1  + harmonic2 ;
>> subplot(2,2,3);plot(t+t_offset, synth_sig);
>> xlabel('Time (seconds)'); ylabel('Amplitude');
>> title('Synthesised segment')
>> subplot(2,2,4); plot(t+t_offset, fundamental);
>> hold on ; plot(t+t_offset, harmonic1);
>> plot(t+t_offset, harmonic2);
>> xlabel('Time (seconds)'); ylabel('Amplitude');
>> title('Three synthesis sinusoids')
```

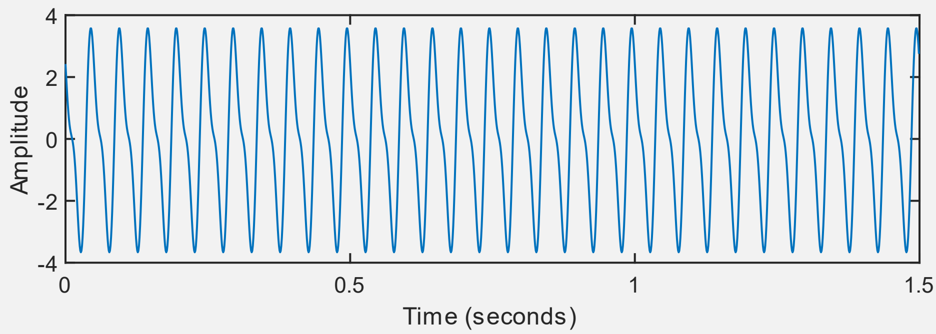


Interpreting the output of the fft function

The fft function returns a sequence of complex numbers, and these complex numbers describe the amplitude and phases of the sinusoidal waveforms that a time-domain signal is comprised of. In this section I'll attempt to explain how to interpret the complex numbers returned by the fft function using some synthesised example signals, as explained in pzdsp.com/vid16.

Let's start by synthesising a time-domain signal that contains three sinusoids. The fft function should be able to determine the amplitudes and phases of these three sinusoids so let's see how it does it.

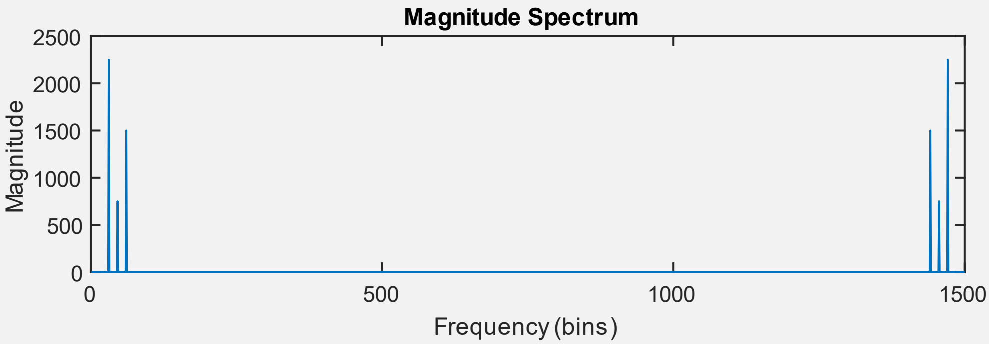
```
>> fs = 1000;
>> t = 0 : 1/fs :1.5-1/fs;
>> x = 3*cos(2*pi*20*t + 0.2) + 1*cos(2*pi*30*t -0.3) +
2*cos(2*pi*40*t + 2.4);
>> plot(t,x);
>> xlabel('Time (seconds) ');
>> ylabel('Amplitude');
```



We know that the time-domain signal shown above contains three sinusoids of frequencies 20 Hz, 30 Hz and 40 Hz with phase offsets of 0.2 radians, -0.3 radians and 2.4 radians, respectively, and amplitudes 3, 1, and 2, respectively. The time-domain signal contains 1500 samples (sampling rate is 1000 Hz) and when we apply the fft function to this signal 1500 complex numbers are returned. If we plot the magnitudes of these 1500 complex numbers, as shown below, we can see three 'spikes' on the left hand side of the plot with another three spikes 'mirrored' on the right hand side. The three pairs of 'spikes' represent the three sinusoidal components that the original signal is comprised of.

```
>> X = fft(x);
```

```
>> plot(abs(X)); xlabel('Frequency (bins)');
ylabel('Magnitude');
>> title('Magnitude Spectrum')
```



The amplitude of the 'spikes' correspond to the amplitude of the sinusoids. Referring to the three 'spikes' on the left-hand side; the spike furthest to the left corresponds to the 20 Hz sinusoid which has the largest amplitude; the middle 'spike' has the lowest amplitude and corresponds to the 30 Hz sinusoid; while the 'spike' to the right of the grouping is twice the amplitude of the middle 'spike' and corresponds to the sinusoid with a frequency of 40 Hz.

If we took a closer look at the values of the variable `X` we'd see that they are complex numbers that contain a lot of zero values. While showing all 1500 values is impractical we can use the following matlab code to look at a few:

```
>> X(30:32) % X(30)    X(31)      X(32)
0+0j      2205.15+447j      0+0j
>> X(45:47)  % X(45)    X(46)      X(47)
0+0j      716.5-221.64j      0+0j
>> X(60:62)  % X(60)    X(61)      X(62)
0+0j      -1106.09+1013.19j      0+0j
```

We can see that there are three non-zero values at indices 31, 46 and 61. The magnitudes of these values are 2250, 750 and 1500, respectively, and these values can also be determined from the plot of the magnitude spectrum by examining the amplitude of each of the 'spikes'. This should make sense, since the plot of the magnitude spectrum is simply a visual representation of the magnitudes of the variable `X`. It's worth noting that if we divide these magnitude values by 750 (which is half the number of values in `X`) then we get a result of 3, 1 and 2 which exactly match the amplitudes of the three sinusoids that the synthesised signal is comprised of.

The phase angles of the complex numbers of X at indices 31, 46 and 61 are 0.2 radians, -0.3 radians and 2.4 radians, respectively. By referring to the code which synthesised the time-domain signal we can see that these phase angles directly correspond to the phases of the sinusoids that the synthesised signal is comprised of.

So, we can see that the fft function can determine the amplitudes and phase of the sinusoids that a signal is comprised. The remaining piece of information is the frequency of each of those sinusoids and to determine the frequency we have to examine the indices of the non-zero values of X i.e. 31, 46 and 61. Before continuing its important to note that matlab and octave index the first value of an array with the number 1, while mathematicians (and most other programming) languages will index the first element of an array with the number 0. The values of the array returned by the fft function (stored in the variable X) are referred to as bin values, with the first element of the array being referred to as bin number 0. The non-zero values of the variable X occurring at indices 31, 46 and 61 therefore correspond to bin numbers 30, 45 and 60. These bin numbers are related to frequency with the bin numbers associated with the left-hand side of the magnitude spectrum being converted to frequency in hertz using the following formula:

$$f = k.f_s/N$$

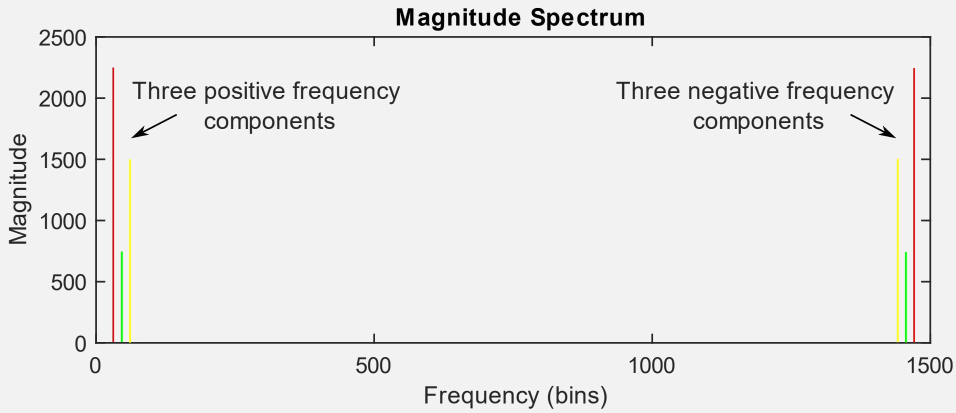
where f is the frequency associated with bin k, f_s is the sampling frequency and N is the total number of bins (which is equal to the number of values in the variable X).

Using this formula for bin values k set to 30, 45 and 60 gives frequencies 20 Hz, 30 Hz and 40 Hz, which correspond to the frequencies of the sinusoids used to synthesise the time-domain signal.

Negative Frequencies

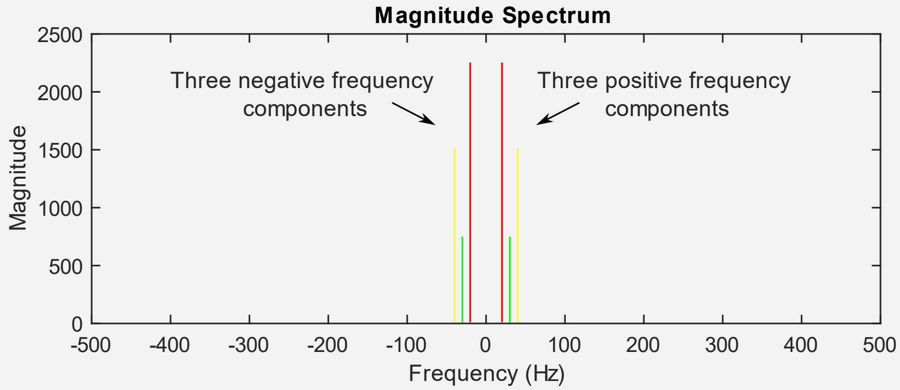
By this stage you should be comfortable with the idea that all signals can be decomposed into sinusoidal waveforms. However, a sinusoidal waveform can also be considered as being the sum of two other waveforms, known as complex exponentials (explained in detail later).

When you plot the magnitude spectrum of a signal you'll see that the spectrum contains 'mirrored' components/spikes at each end of the spectrum. For example, earlier on, when we examined the signal which comprised of three sinusoids, the following plot was produced (the plot has been modified so that frequency components are different colours). Notice how the left-hand side of the spectrum is a 'mirror image' of the right-hand side.



The three components/spikes to the left are associated with what are referred to as 'positive frequency complex exponentials' while the ones to the right are associated with 'negative frequency complex exponentials'. The plot shows 'mirrored pairs' of frequency components using three different colours (red, green yellow). Each 'mirrored pair' of complex exponentials actually represent a single sinusoidal waveform, as shown mathematically below. In most situations the 'mirrored' half of the spectrum is not required for engineers and scientists to analyse the magnitude spectrum, since they are effectively redundant, and so a single-sided spectrum is often displayed for simplicity.

Note that you will often find the magnitude spectrum plotted in units of Hertz with the negative frequency components shown to the left, as shown below. Octave/Matlab code to create these plots is explained in pzdsp.com/vid17.



Complex exponentials are described mathematically by the following function:

$$x(t) = Ae^{j(2\pi f t + \varphi)}$$

The A parameter specifies the amplitude of the sinusoid; f specifies the frequency and φ parameter specifies the phase offset. Notice the similarity of these parameters with sinusoidal parameters.

Using the well-known Euler's Formula $e^{j\theta} = \cos(\theta) + j\sin(\theta)$, and the fact that $\cos(-\theta) = \cos(\theta)$ and $\sin(-\theta) = -\sin(\theta)$, it can be shown that $e^{-j\theta} = \cos(\theta) - j\sin(\theta)$. It therefore can also be shown that

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

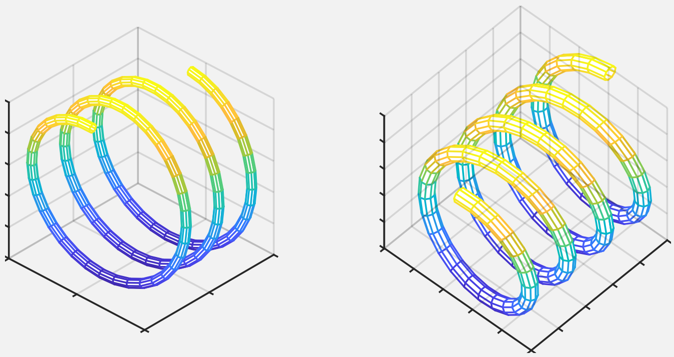
Using this result it can therefore be shown that any sinusoidal waveform is the sum of two complex exponential waveforms since

$$A\cos(2\pi f t + \varphi) = \frac{Ae^{j(2\pi f t + \varphi)} + Ae^{-j(2\pi f t + \varphi)}}{2}$$

The complex exponential $Ae^{j(2\pi f t + \varphi)}$ is the 'positive frequency component' while $Ae^{-j(2\pi f t + \varphi)}$ is the 'negative frequency component'.

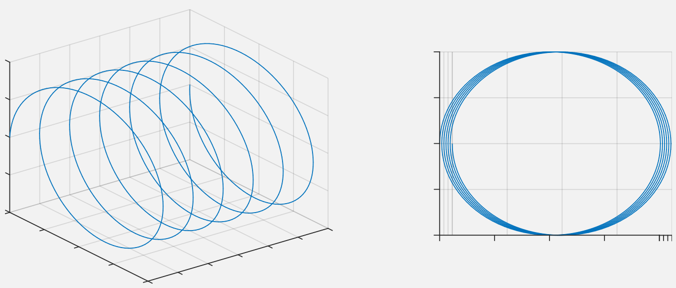
Visualising complex exponentials and negative frequency

Complex exponentials have a helix shape (like a spring or corkscrew) and they can be tricky to visualise if you're not sure what you're looking at. To make this process easier I'd like you to show helix examples that are reasonably easy to interpret. I'd also like you to appreciate that the two examples below spiral or rotate in opposite directions. The helix to the left rotates in an anti-clockwise direction while the one to the right rotates in a clockwise direction. This direction of rotation is extremely relevant for understanding negative frequency.

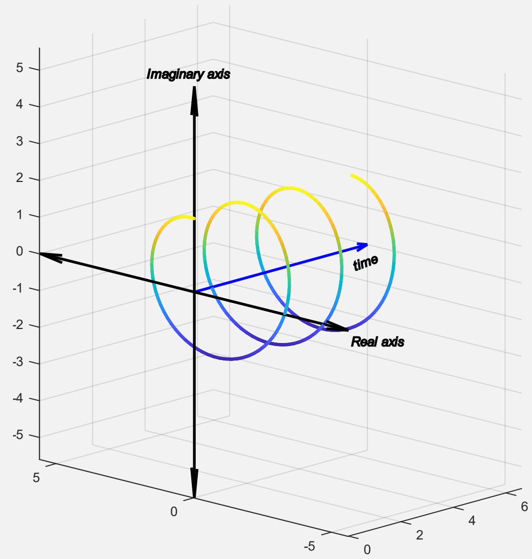


The direction of rotation of the above example helixes are relatively easy to interpret because of their 3-dimensional structure and colouring. The helix shapes you'll see later may be more difficult as they will not be rendered in such a 'solid' format. However, once you appreciate that you are viewing a helix this should not be an issue for most readers.

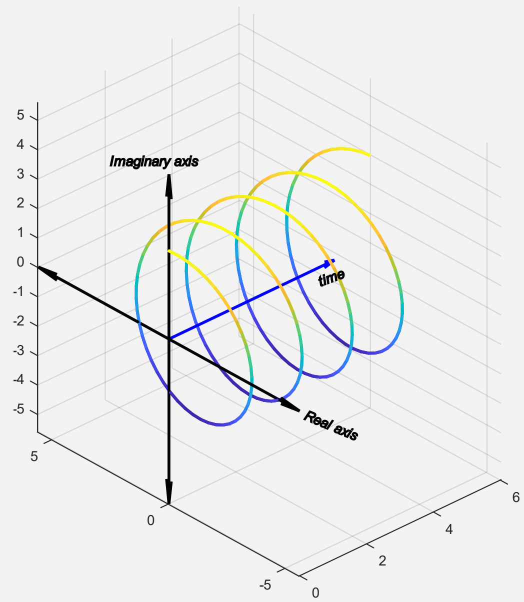
The plot below shows another helix which rotates in a clockwise direction. The helix to the right is the same helix but viewed 'face on' to highlight the fact that the helix has a circular shape when viewed from that perspective.



Now let's take a look at a visualisation of a complex exponential waveform (the colours on the helix are there just to make it easier to see the 3-D shape!). I'll explain where this visualisation comes from later, but it can be useful to have a mental picture before getting into the deeper explanation. Don't be worried if you don't fully understand the illustration at this stage! You'll notice that there are three axes, a time axis, a real axis and an imaginary axis. The complex exponential waveform shown in the plot is a helix (spring-like shape) that rotates in an anti-clockwise direction. It is the direction of rotation that is key to understanding negative frequencies!



The complex exponential waveform above is associated with a positive frequency because it rotates in an anti-clockwise direction. I'll explain why this is case later on, but for the moment just accept it, and also accept that negative frequencies are associated with complex exponential waveforms that rotate in a clockwise direction, like the one illustrated below.



As is the case with sinusoidal waveforms, complex exponential waveforms have an amplitude, frequency and phase associated with them, however the direction of rotation introduces another feature i.e. a positive frequency (for complex exponentials that rotate anti-clockwise) and negative frequency (for complex exponentials that rotate clockwise). Please note that a persistent feature of complex exponentials is that they have a helix shape. The negative frequency waveform shown has an amplitude of 3 while the positive frequency waveform has an amplitude of 2. Also, notice that the negative frequency complex exponential has four rotations over 6 second i.e. a frequency of 0.66 Hz. The positive frequency complex exponential has three complete rotations over 6 seconds i.e. a frequency of 0.5 Hz.

At this stage you have been exposed to the most important aspects of complex exponentials and negative frequencies. If you are able to appreciate the idea that positive and negative frequency relates to the direction of rotation of the helix shapes associated with complex exponentials then you are in a very good position!

Why complex exponential waveforms have a helix shape

Euler's Formula states that $e^{j\theta} = \cos(\theta) + j\sin(\theta)$. Notice the presence of the imaginary unit $j = \sqrt{-1}$ in the formula!

Complex exponentials are therefore the sum of a cosine waveform and a sine waveform, where the sine waveform is multiplied by the imaginary unit $j = \sqrt{-1}$, i.e.

$$Ae^{j(2\pi ft+\varphi)} = A\cos(2\pi ft + \varphi) + jA\sin(2\pi ft + \varphi)$$

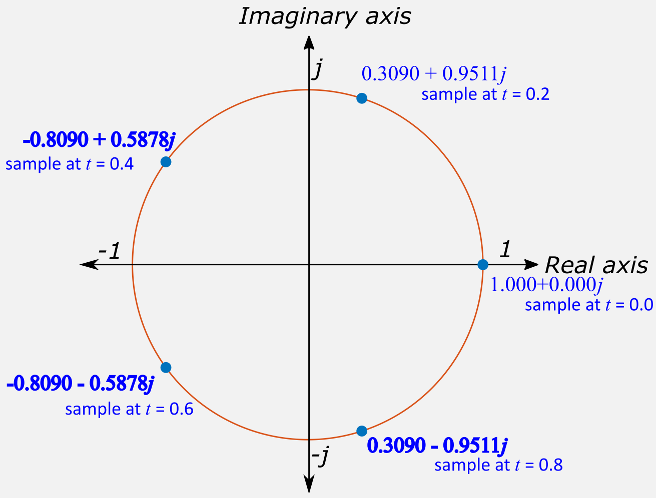
To create a plot of a sinusoidal waveform you could evaluate the mathematical expression $A\cos(2\pi ft + \varphi)$ for a range of values of the time variable t and plot the resulting sequence of 'real' amplitude values against time. If you were to evaluate complex exponential waveforms ($Ae^{j(2\pi ft+\varphi)}$) for a range of values of the time variable t, then you would obtain a sequence of 'complex' values i.e. values with 'real' and 'imaginary' terms.

Since complex exponentials have 'complex terms' (i.e. real and imaginary terms) then we can use an argand diagram to visualise these terms. The following video may help some readers at this stage pzdsp.com/vid27.

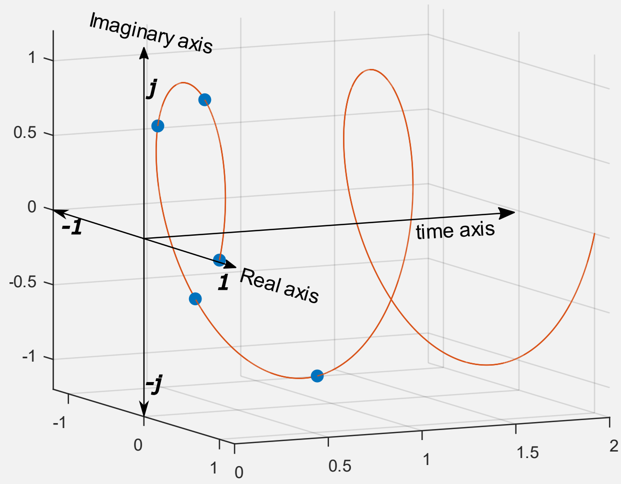
Let's evaluate a complex exponential waveform $e^{j2\pi t}$ which has unit amplitude, zero phase offset and frequency 1 Hz, just to see the shape that will be produced (and yes, it will be a helix!). The table below shows the result of evaluating $e^{j2\pi t} = \cos(2\pi t) + j\sin(2\pi t)$, for a range of increasing values of t.

time (t)	$\cos(2\pi t)$	$\sin(2\pi t)$	$e^{j2\pi t} = \cos(2\pi t) + j\sin(2\pi t)$
0	1.0000	0.000	1.0000 + 0.0000j
0.2	0.3090	0.9511	0.3090 + 0.9511j
0.4	-0.8090	0.5878	-0.8090 + 0.5878j
0.6	-0.8090	-0.5878	-0.8090 - 0.5878j
0.8	0.3090	-0.9511	0.3090 - 0.9511j

Plotting these complex numbers on an argand diagram shows that they all lie on a unit circle. Notice how the complex numbers evolve in anti-clockwise direction as time increases.

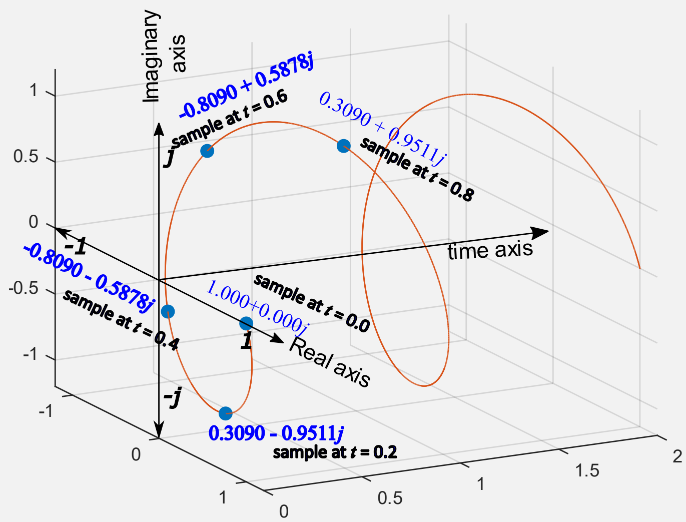


If we add a time axis to the argand plot the helix shape will emerge and you should notice that the helix rotates in an anti-clockwise direction. The red line showing the shape of the helix is the waveform $e^{j2\pi t}$ evaluated over the range 0 to 2 seconds, at intervals of 0.001 seconds. These 2000 values were, of course, determined using a computer!



Now let's evaluate the negative frequency complex exponential waveform $e^{-j2\pi t}$, over increasing values of t. These complex numbers evolve as helix shape rotating in a clockwise direction as time increases, for this case.

time (t)	$\cos(2\pi t)$	$\sin(2\pi t)$	$e^{-j2\pi t} = \cos(2\pi t) - j\sin(2\pi t)$
0	1.0000	0.000	1.0000 + 0.0000j
0.2	0.3090	0.9511	0.3090 - 0.9511*j*
0.4	-0.8090	0.5878	-0.8090 - 0.5878*j*
0.6	-0.8090	-0.5878	-0.8090 + 0.5878*j*
0.8	0.3090	-0.9511	0.3090 + 0.9511*j*

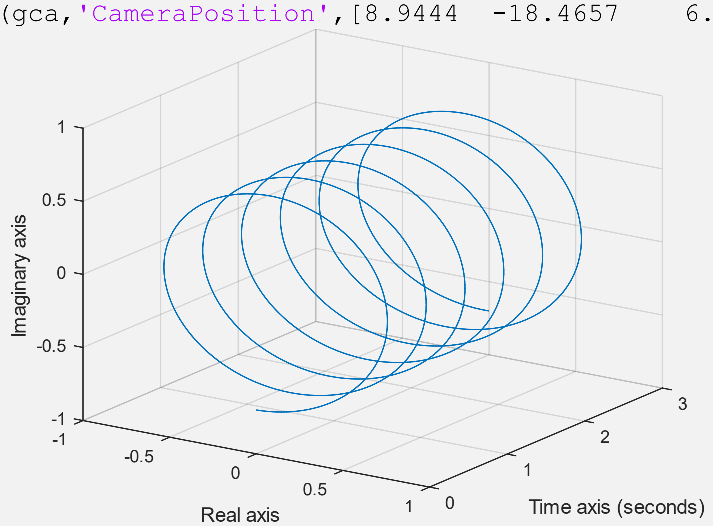


From above you can see that the positive frequency complex exponential $e^{j2\pi t}$rotates in an anti-clockwise direction while the negative frequency complex exponential $e^{-j2\pi t}$ rotates in a clockwise direction.

You should appreciate that in general, a complex exponential waveform will always have a helix shape. The waveform given by $Ae^{j(2\pi ft+\varphi)}$ will rotate at f rotations per second, and that the radius of the helix will be A. The phase offset φ simply moves the 'starting point' of the helix at time t = 0. The same applies to a waveform given by $Ae^{-j(2\pi ft+\varphi)}$, the only difference is the direction of rotation of the helix shape.

The following Matlab/Octave code can be used to create plots of helix's that the interested reader may find beneficial. As always skip the code provided if it's not helpful.

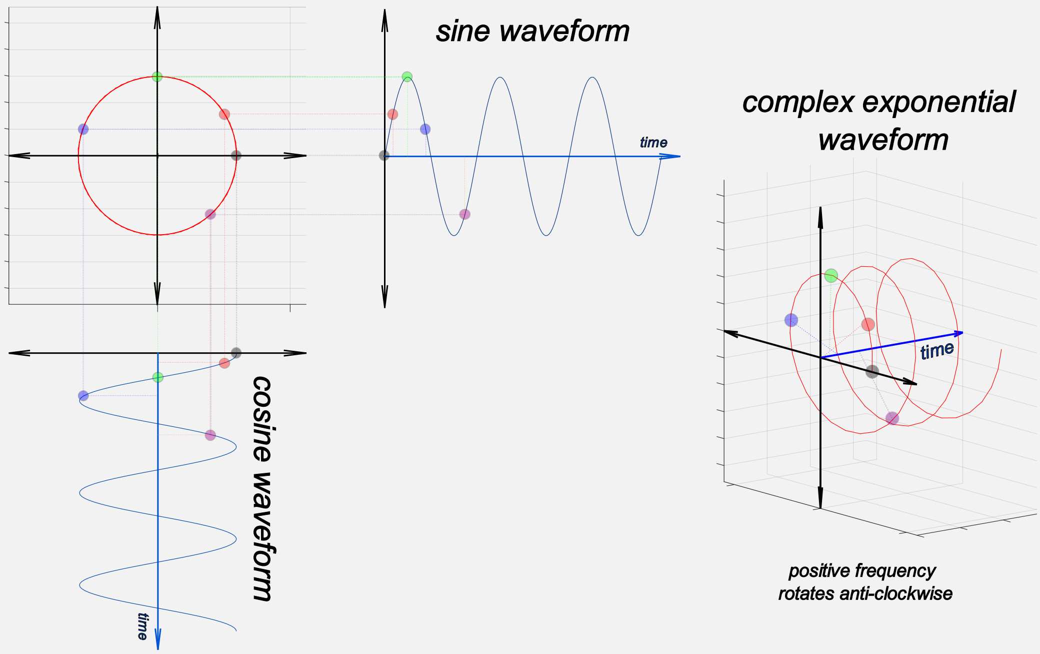
```
>> T = 0.001; % make this variable smaller to produce smoother plots
>> t = 0: T : 3; % used to evaluate the waveform from t=0 to t=3 (in steps of T)

>> A = 0.7; f = 2; w=2*pi*f; phi = -pi/2;
>> freq_direction = 1; %set to 1 for positive frequency
>> x = A*exp(freq_direction*j*(w*t + phi)); %evaluate the complex exponential waveform for the times specified earlier

>> plot3(real(x), t,imag(x))
>> ylabel('Time axis (seconds)')
>> xlabel('Real axis')
>> zlabel('Imaginary axis')
>> grid on
>> set(gca,'CameraPosition',[8.9444  -18.4657    6.5444])
```



Another perspective on complex exponentials

If you feel you have a good understanding of complex exponentials and negative frequency then feel free to skip this section. I just wanted to provide a slightly different perspective into visualising complex exponentials that might help some readers.

Complex exponentials are the combination of a cosine waveform and sine waveform. Consider the plot below which shows a sine waveform and a cosine waveform. In addition, there is a plot showing 5 points (coloured black, greed, red, blue, and purple) which all appear on the circumference of a circle.



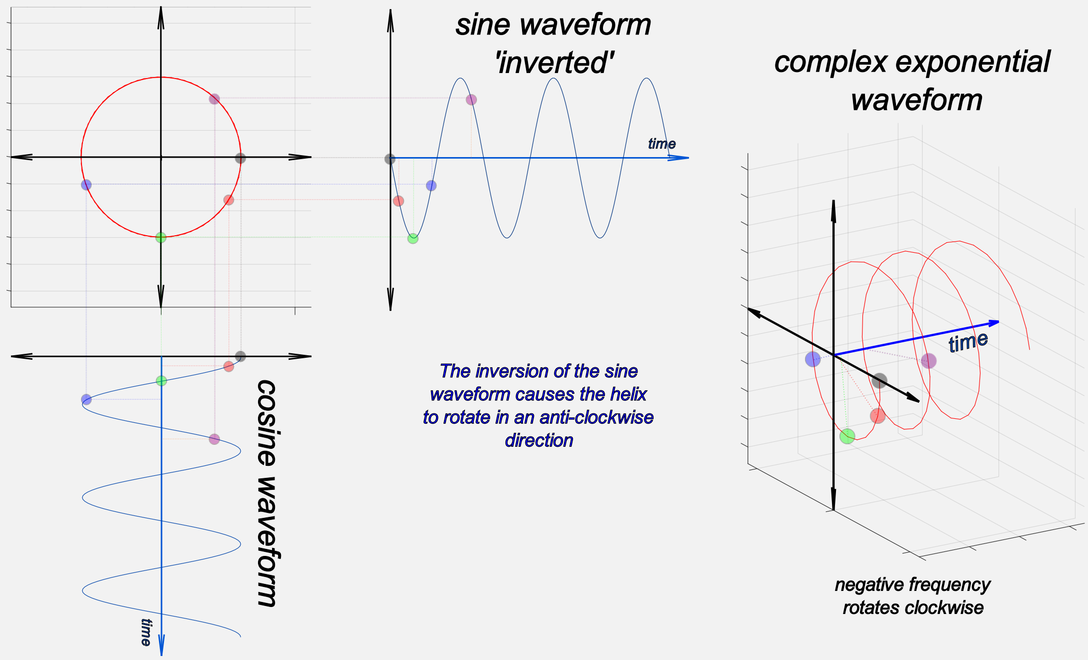
There are also 5 points shown on the cosine and sine waveforms and they appear in order of black, red, green, blue, and purple along the time axis, at the same time locations.

Notice how the horizontal position of the black point on the circle corresponds to the amplitude of the black point shown on the cosine waveform, and that its vertical position corresponds to the amplitude of black point on the sine waveform.

Also notice how the horizontal position of the green point on the circle corresponds to the amplitude of the green point shown on the cosine waveform, and that its vertical position corresponds to the amplitude of green point on the sine waveform.

The position of all the different coloured points on the circle are determined in a similar way. The helix plot to the right above shows that these points on the circle form a helix shape as they evolve over time.

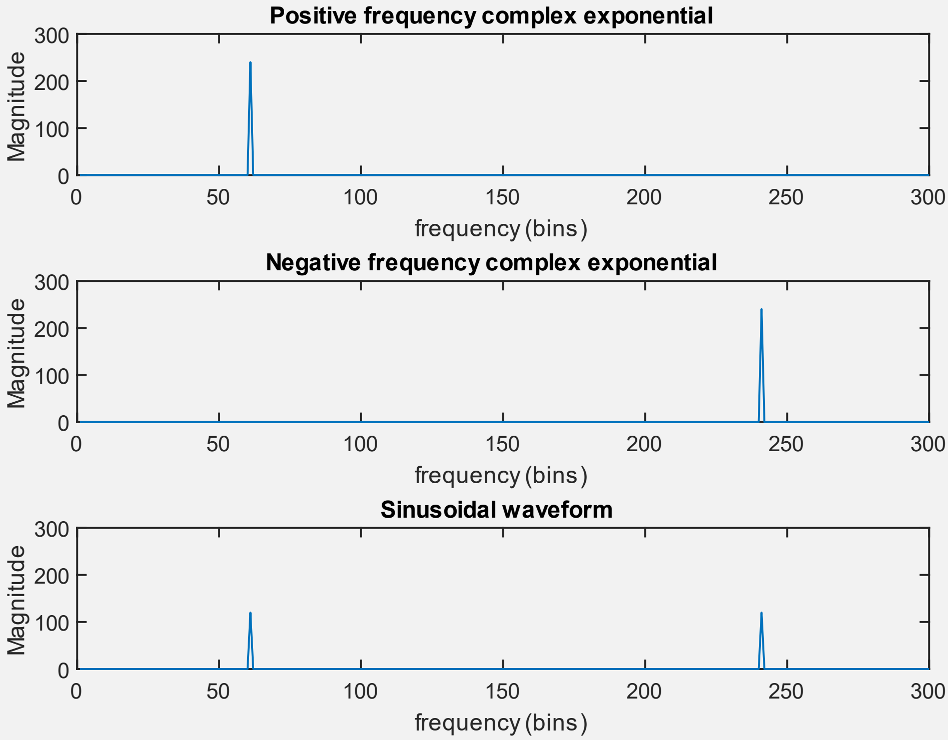
The plot below is very similar to the previous plot except that the sine waveform has been inverted. Inverting the sine waveform has the affect of causing the helix to rotate in the opposite direction.



The magnitude spectrum of a complex exponential

As discussed earlier, a Fourier analysis of a sinusoidal waveform will contain two components in its magnitude spectrum i.e. a positive frequency complex exponential and a negative frequency complex exponential.

As you might expect, a Fourier analysis of a complex exponential will contain a single frequency component/'spike' in the magnitude spectrum. A negative frequency complex exponential will appear on the opposite side of the spectrum to a positive frequency complex exponential (see Octave/Matlab code below for further insight).



The figure above shows the magnitude spectrum of a positive and negative complex exponential in addition to the spectrum of a sinusoidal waveform. All waveforms have the same amplitude, frequency and phase parameters (see code below). Notice that the two frequency components associated with the sinusoidal waveform are half the magnitude of the complex exponential waveforms. This makes sense since a sinusoidal waveform is comprised of two complex exponentials which are half the amplitude of the sinusoid i.e.

$$A\cos(2\pi ft+\varphi)=\frac{Ae^{j(2\pi ft+\varphi)}+Ae^{-j(2\pi ft+\varphi)}}{2}$$

```
>> T = 0.01; % make this variable smaller to produce time-domain
smoother plots
>> t = 0: T : 3-T; % used to evaluate the waveform from t=0 to t=3 (in
steps of T)

>> A = 0.8; f = 20; w=2*pi*f; phi = -pi/2;
>> pos_comp_exp = A*exp(j*(w*t + phi));
>> neg_comp_exp = A*exp(-j*(w*t + phi));
>> sinusoid = A*cos((w*t + phi));

>> subplot(3,1,1)
>> plot(abs(fft(pos_comp_exp)))
>> xlabel('frequency (bins)');ylabel('Magnitude');ylim([0 300]);
>> title('Positive frequency complex exponential')
>> subplot(3,1,2)
>> plot(abs(fft(neg_comp_exp)))
>> xlabel('frequency (bins)');ylabel('Magnitude');ylim([0 300]);
>> title('Negative frequency complex exponential')
>> subplot(3,1,3)
>> plot(abs(fft(sinusoid)))
>> xlabel('frequency (bins)');ylabel('Magnitude');ylim([0 300]);
>> title('Sinusoidal waveform')
```