

Fourier Transforms, DFTs, and FFTs

Author: John M. Cimbala, Penn State University
 Latest revision: 22 February 2010

Introduction

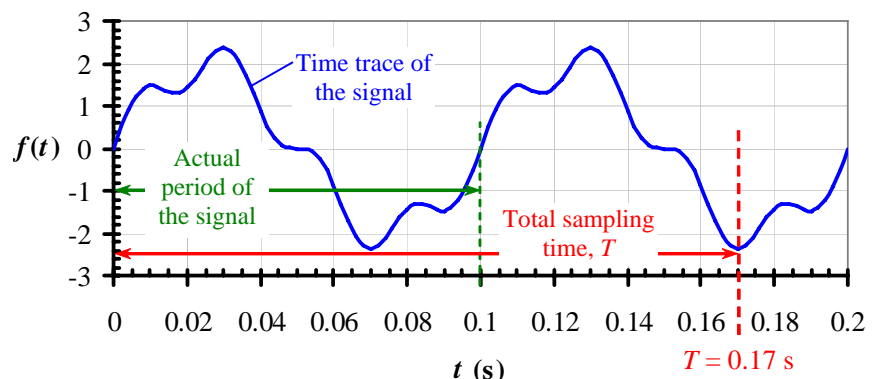
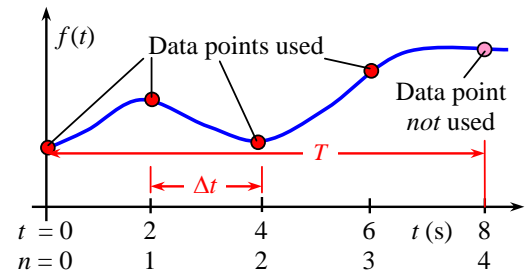
- In *spectral analysis*, our goal is to determine the *frequency content* of a signal.
- For analog signals, we use *Fourier series*, as discussed in a previous learning module.
- For digital signals, we use *discrete Fourier transforms*, as discussed in *this* learning module.

Fourier transform (FT)

- The *Fourier transform (FT)* is a *generalization of the Fourier series*.
- Instead of the *sines* and *cosines* in a Fourier series, the Fourier transform uses *exponentials* and *complex numbers*. Recall that $e^{i\theta} = \cos \theta + i \sin \theta$; thus, we are able to combine the sine and cosine components of the Fourier series into *combined* components of the Fourier transform.
- Instead of the *summations* used in a Fourier series, the Fourier transform uses *integrals*.
- For a signal or function $f(t)$, the *Fourier transform* is defined as $F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$, and the *inverse Fourier transform* is defined as $f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$, where $e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$, i is the *unity imaginary number*, defined as the square root of -1 , $i = \sqrt{-1}$, and ω is the range of angular frequencies associated with the signal – the *frequency content* of the signal.
- When working with time and the signal $f(t)$, we are working in the *time domain*, and the variables are *real*.
- When working with angular frequency and the Fourier transform $F(\omega)$, we are working in the *frequency domain*, and $F(\omega)$ is *complex*.
- The FT is an *analog tool* – it is used for analyzing the frequency content of *continuous* signals.

Discrete Fourier transform (DFT) [also sometimes called the digital Fourier transform]

- We define the *discrete Fourier transform (DFT)* – a Fourier transform for a *discrete* (digital) signal.
- The DFT is a *digital tool* – it is used for analyzing the frequency content of *discrete* signals.
- The discrete Fourier transform is defined as $F(k\Delta f) = \sum_{n=0}^{N-1} f(n\Delta t) e^{-i(2\pi k\Delta f)(n\Delta t)}$ for $k = 0, 1, 2, \dots, N - 1$.
- Note that *summation* has replaced *integration* since *discrete* rather than *continuous* data are being examined.
- In the *time domain*, the relevant variables are:
 - N = total number of discrete data points taken.
 - T = total sampling time.
 - Δt = time between data points, $\Delta t = T/N$.
 - f_s = sampling frequency, $f_s = 1/\Delta t = N/T$.
- Note that *integers n and k in the above definition of the DFT have values from 0 to $N - 1$, not from 1 to N .*
- For example, consider data sampled as in the plot to the right. Data are sampled discretely at a sampling frequency of 0.5 Hz. Starting at time $t = 0$, $N = 4$ data points are taken. Here, $T = 8$ s, $\Delta t = T/N = (8 \text{ s})/4 = 2$ s, and $f_s = 1/\Delta t = 0.5$ Hz. The discrete data used to calculate the DFT are those at $t = 0, 2, 4,$ and 6 seconds. The data point at $n = N$ (at $t = T = 8$ s) is *not used*.
- Rather, *the last data point is assumed to be identical to the first data point*.
- Consider a periodic signal that repeats itself every 0.1 s, as plotted.



- In general, *we do not know the period of the signal ahead of time*, and the sampling may stop at a *different phase* in the signal than where sampling started; the last data point is then *not* identical to the first data point.
- In the above example, we start sampling at $t = 0$, and stop sampling at $T = 0.17$ s – the phase at $t = T$ differs from that at $t = 0$.
- This is because, unlike Fourier series analysis, **T in the discrete Fourier transform is not necessarily equal to the fundamental period of the original signal.**
- In most actual experimental measurements, the fundamental period of the signal is not even *known*; thus **T is, in general, totally unrelated to anything inherent in the signal itself.** In some experimental measurements, the signal to be examined by a Fourier transform is not even a purely periodic signal!
- ***This phase discrepancy leads to some inaccuracies when using discrete Fourier transforms***, as discussed later.
- In the *frequency domain*, the relevant variables are defined as follows:
 - Δf = the **frequency increment**, also called the **frequency resolution**, of the DFT output, $\Delta f = 1/T$.
 - $F(k\Delta f)$ = the discrete Fourier transform output, one complex value for each discrete frequency, that provides information about the relative contribution to the signal by each discrete frequency.
- **The frequency increment Δf of a DFT is analogous to the fundamental frequency of a Fourier series**, in that the DFT provides information about the *relative contribution* of the **harmonics** of Δf , just as the Fourier series coefficients provide information about the relative contribution of the harmonics of the fundamental frequency.
 - For $k = 1$, $F(1\Delta f)$ is the DFT at the *first* harmonic frequency Δf .
 - For $k = 2$, $F(2\Delta f)$ is the DFT at the *second* harmonic frequency $2\Delta f$.
 - For $k = 3$, $F(3\Delta f)$ is the DFT at the *third* harmonic frequency $3\Delta f$, and so on.
- Note, however, that unlike a Fourier series, **the frequency increment Δf , in general, has nothing to do with the frequency content or the fundamental frequency of the original signal!** Why? Because, Δf is simply $1/T$, where T is an arbitrary time period (the total sampling time), as discussed above. This leads to some inherent errors associated with discrete Fourier transforms that will become clearer when examples are shown later.
- The **Nyquist criterion** is also important in DFT analysis. When sampling at frequency f_s , **we obtain reliable frequency information only for frequencies less than $f_s/2$.** (Here, *reliable* means *without aliasing problems*.)
- In the DFT output, there are N output values, $F(k\Delta f)$ for $k = 0, 1, 2, \dots, N - 1$, at N discrete frequencies.
- Since $\Delta f = 1/T$ and $f_s = N/T$, we can easily calculate at what value of k the frequency $k\Delta f$ equals $f_s/2$. Namely,

$$k\Delta f = \frac{f_s}{2} \text{ when } k = \frac{f_s}{2\Delta f} = \frac{N/T}{2(1/T)} = \frac{N}{2}.$$
- Therefore, we conclude that the **maximum useful frequency f_{\max}** from a DFT output, also called the **folding frequency f_{folding}** , is

$$f_{\max} = f_{\text{folding}} = \frac{f_s}{2} = \frac{N}{2} \Delta f.$$
- In other words, **only half of the N available DFT output values are useful** – for k from 0 to $N/2$, which corresponds to a frequency from 0 to $f_s/2$. **This is a direct result of the Nyquist criterion.**

Example

- Suppose we sample a signal for $T = 4$ seconds, at a sampling rate of $f_s = 100$ Hz.
 - How many data points are taken? Answer: $N = Tf_s = (4 \text{ s})(100 \text{ Hz}) = N = 400$ data points.
 - How many *useful* DFT output values are obtained? Answer: $N/2 = 200$ useful output values. **Actually, the output contains amplitudes at both extremes: 0 Hz (DC) and f_{folding} , so the real answer is 201 values.**
 - What is Δf ? Answer: $\Delta f = 1/T = 1/(4 \text{ s}) = 0.25$ Hz. **Frequency resolution = 0.25 Hz.**
 - What is the maximum frequency for which the DFT output is useful and reliable? Answer: $f_{\max} = f_s/2 = (100 \text{ Hz})/2 = 50$ Hz. Or, $f_{\max} = (N/2)\Delta f = (400/2)(0.25 \text{ Hz}) = 50$ Hz. **$f_{\max} = f_{\text{folding}} = 50$ Hz.**
- The other half of the output values ($f > f_{\text{folding}}$) are thrown out or ignored.

Fast Fourier transform (FFT)

- **The fast Fourier transform is simply a DFT that is fast to calculate on a computer.**
- All the rules and details about DFTs described above apply to FFTs as well.
- For many FFTs (such as the one in Microsoft Excel), the computer algorithm restricts N to a power of 2, such as 64, 128, 256, and so on. However, some FFT algorithms do not have such a restriction. (For example, the FFTs used in LabVIEW and in MATLAB do not have this power of 2 restriction.)

- It is not the goal of this learning module to teach the details of how to efficiently calculate an FFT. Instead, **the goal here is to teach you how to use and interpret FFTs**, and how to set up the parameters so as to achieve adequate frequency resolution while minimizing problems such as *leakage*, which is discussed below.
- The output $F(k\Delta f)$ of an FFT subroutine is a series of complex numbers, one for each discretely sampled data point, representing each discrete frequency, only half of which are useful because of the Nyquist criterion.
- Although output $F(k\Delta f)$ is *complex*, **it is the magnitude or amplitude of the complex number that is actually used to compare the relative importance of the various frequencies**.
- Recall that for some complex number $z = x + iy$, where x is the real part and y is the imaginary part, the magnitude of z is given by $|z| = \sqrt{x^2 + y^2}$.
- Note that the magnitude of a complex number is called the **modulus**; in Microsoft Excel, the function to calculate the modulus of a complex number is **IMABS**. In other words, $\text{IMABS}(z) = |z| = \sqrt{x^2 + y^2}$.

Frequency spectrum

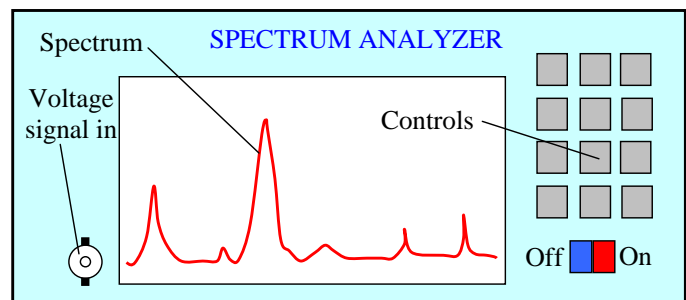
- A plot of the magnitude of the FFT output $|F|$ versus frequency f is called the **frequency spectrum**. Some authors call this the **amplitude spectrum** since the dimensions and units of the vertical axis are the same as those of the amplitude of the original signal, after some modifications as discussed below.
- In the typical case in which the input signal is a voltage, the units of the modified $|F|$ are also voltage.
- Over a given frequency range, the amplitude defined in this way indicates the **relative importance** of that frequency range to the signal. **A frequency spectrum plot formed from an FFT is analogous to the harmonic amplitude plot formed from a Fourier series**.

Usefulness of the FFT

- The fast Fourier transform (FFT) is extremely useful in analyzing **unsteady measurements**, because **the frequency spectrum from an FFT provides information about the frequency content of the signal**.
- Examples come from nearly every area of engineering, such as vibrations, where we need to know the frequency content of the vibration; fluid flow, where we need to know the frequency content of the turbulent fluctuations; and acoustics, where we need to know the frequency content of a sound signal, to mention just a few.

Spectrum analyzers

- There are laboratory instruments called **spectrum analyzers** that are designed to calculate and display the frequency spectrum of a voltage signal, as illustrated in the drawing to the right.
- Internally, an FFT is calculated.
- In the lab that accompanies this learning module, you will generate a **virtual spectrum analyzer** using LabVIEW and MATLAB.



FFT example – a pure sine wave

- Consider first the FFT of a **pure sine wave**. Suppose the signal is a 10 Hz sine wave with a peak-to-peak amplitude of -1 to 1 volt, $f(t) = \sin(2\pi(10 \text{ Hz})t)$.
- The **ideal Fourier transform** would have a spike of magnitude 1 Volt at a frequency of exactly 10 Hz, since **all of the energy in the signal is at 10 Hz, and no other frequency contains any energy**.
- Unfortunately, the actual frequency spectrum generated by an FFT of this signal will not be that clean, however, but depends on a number of factors, such as
 - sampling rate f_s
 - number of data points sampled N
 - total time of data collection T , where $T = N/f_s$
 - the windowing function used (to be explained in the next learning module)
- **Note:** The frequency axis of the FFT goes from 0 to $f_s/2$ (the folding frequency), and the output is at discrete frequencies with an interval determined by the frequency resolution $\Delta f = 1/T$.

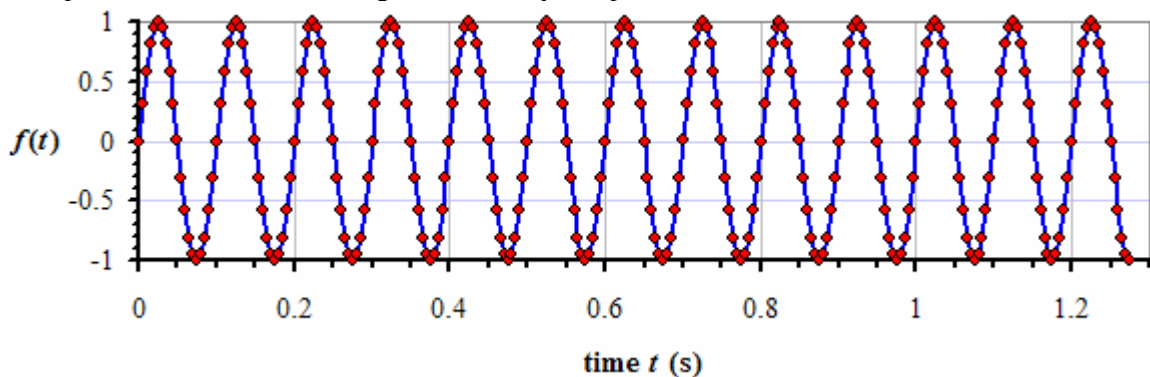
Calculating and plotting FFTs in Microsoft Excel

- Most spreadsheets have a built in FFT function. In fact, all the frequency spectra shown here were generated and plotted using Microsoft Excel.
- In Excel, an FFT is calculated and plotted using the following procedure:

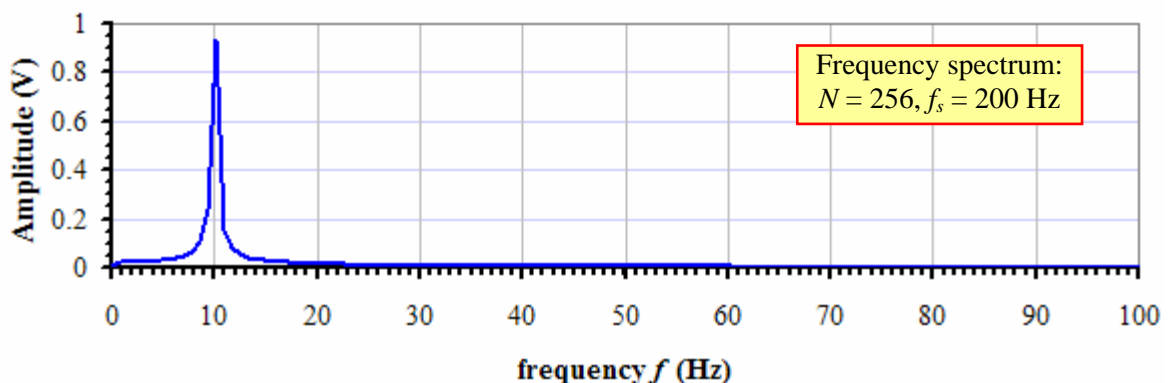
- Create and fill a column representing t (time for each discrete data point), from $t = 0$ to T , with $\Delta t = 1/f_s$.
- Fill a column representing the discrete signal $f(t)$ itself (typically obtained experimentally by sampling data digitally) – one data point for each row (each time from $t = 0$ to T in the time column).
- Click on Tools (Excel 2003) or the *Data group* (Excel 2007). Then Data Analysis-Fourier Analysis, and OK. *Note*: If the Data Analysis tool is not available in Excel, it must be added in by Tools-Add-Ins (Excel 2003) or the Office Button-Excel Options-Add Ins (Excel 2007), then install Analysis ToolPak.
- For the *Input Range*, select the signal to be analyzed – the column of discrete signal data $f(t)$.
- For the *Output Range*, select the top cell of an empty column as the top of the output column. OK. The FFT output – a column of complex numbers – is written to the spreadsheet.
- Since the output of the FFT is a column of complex numbers, convert each value to an *amplitude* by using Excel's IMABS function. Calculate your amplitudes in another separate column.
- Modify (correct) the amplitudes as follows (do this in yet another column):
 - Divide each amplitude by $(N/2)$ so that the corrected amplitude corresponds directly to the amplitude of the signal, as will be illustrated later by example.
 - In addition, divide the *first* amplitude (at $f = 0$ Hz) by 2 so that this amplitude corresponds properly to the DC offset (average value) of the signal.
- Note that only *half* of these amplitudes are useful for the frequency spectrum because of the Nyquist criterion. The second half of the column (everything beyond $k = N/2$) can be deleted or ignored.
- Create and fill a column representing the discrete frequencies, starting at $f = 0$ (the DC offset) and incremented by Δf up to the maximum useful frequency, $f_{\text{folding}} = f_s/2$.
- Finally, plot the values in the modified amplitude column as a function of discrete frequency. The result is the frequency spectrum. The horizontal axis should range from 0 to f_{folding} .
- One caution when using Excel – when the input signal changes, **the FFT output in Excel is not automatically updated when the time trace is updated. You must re-run the FFT macro every time the time trace changes.**

Leakage and Example FFTs in Microsoft Excel

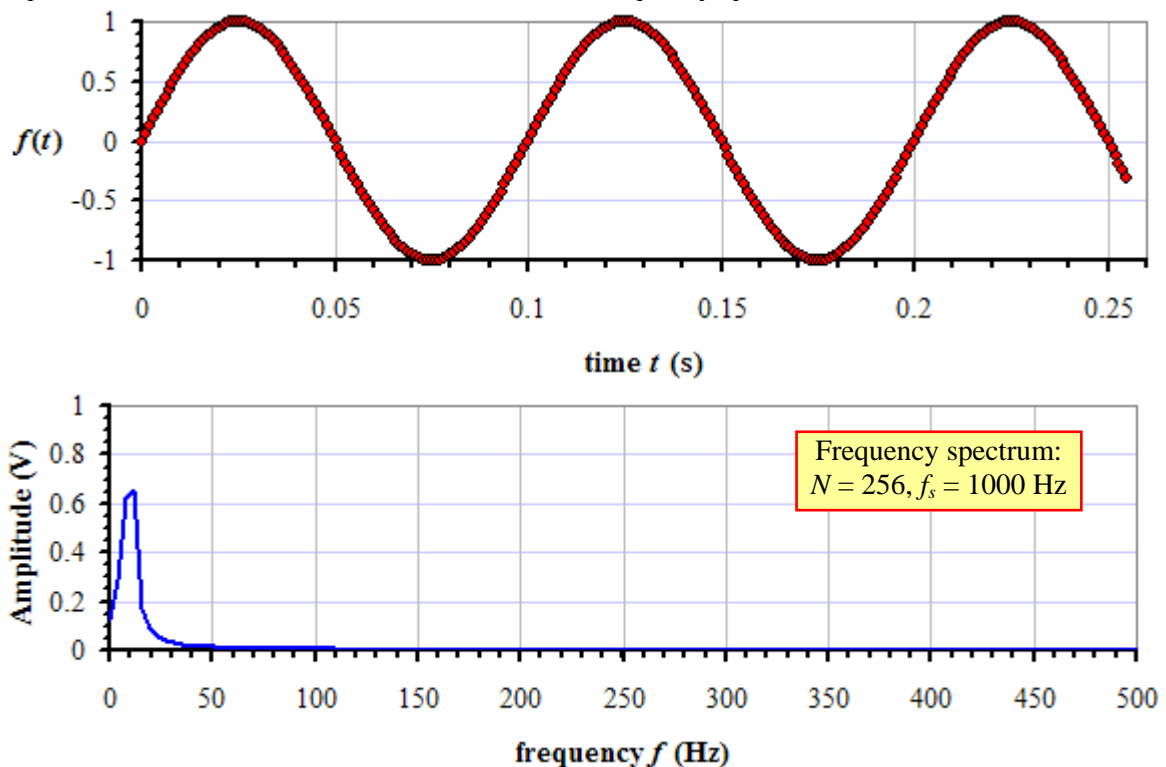
- To illustrate FFTs in Excel, consider a simple 10 Hz sine wave with amplitude = 1 V, as previously.
- As a starting point, f_s is chosen to be 200 data points per second, and N is chosen to be 256. In other words, the 10 Hz sine wave is sampled at 200 Hz for 1.28 seconds ($T = N/f_s = 256/(200 \text{ Hz}) = 1.28 \text{ s}$).
- Both the *time trace* (the signal in the time domain) and the *frequency spectrum* (the modified magnitude of the FFT output in the frequency domain) are shown below.
- First we plot the time trace, showing the discretely sampled data:



- Next, we plot the frequency spectrum for $N = 256$ and $f_s = 200$. The frequency resolution for the spectrum is $\Delta f = 1/T = 1/(1.28 \text{ s}) = 0.781 \text{ Hz}$ (to three significant digits):

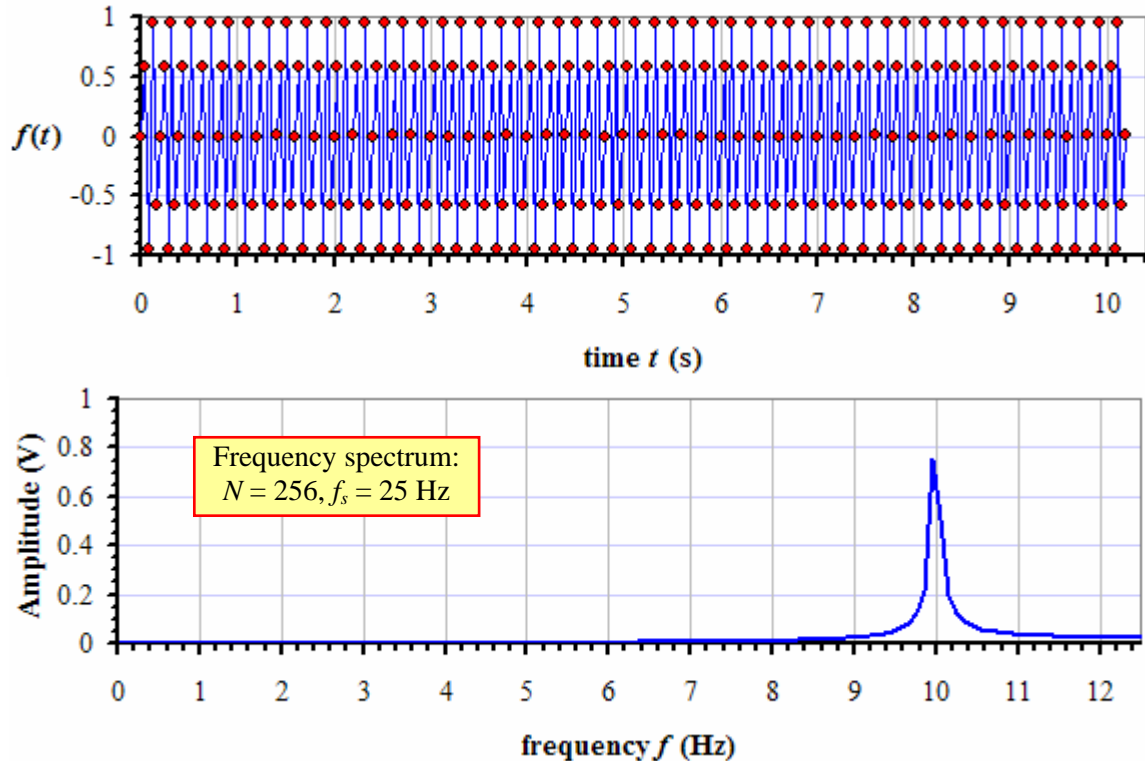


- The following points about the above plot are noted:
 - Although the frequency spectrum correctly shows a spike around 10 Hz, the spike is not infinitesimally narrow. In fact, it appears from the frequency spectrum that there is a significant component of the signal at frequencies *near* 10 Hz (specifically within about 5 Hz to either side of 10 Hz).
 - This unphysical error in the FFT is called **leakage**. **Leakage appears when the discrete data acquisition does not stop at exactly the same phase in the sine wave as it started.** In principle, if an infinite number of discrete data points are taken, leakage would not be a problem. However, any *real* data acquisition system performing FFTs uses a finite (rather than infinite) number of discrete data points, and there will always be some leakage.
 - The maximum amplitude of the FFT is not *exactly* 1 V (in fact it is less than 1 V), even though the amplitude of the original signal is exactly 1 V. This is another consequence of leakage – **some of the energy at the signal frequency (10 Hz) is erroneously distributed among frequencies near 10 Hz, thereby reducing the calculated amplitude at 10 Hz.**
 - The frequency range is from 0 to 100 Hz, half of the sampling frequency, as discussed above. This is a consequence of the Nyquist criterion. We ignore the other half of the data (100 to 200 Hz).
- How can leakage be reduced? Will a higher sampling rate help? To find out, we repeat this example, but with a higher sampling rate.
- The same number of data points (256) of this sine wave are now taken at 1000 Hz instead of 200 Hz. All other parameters remain the same. The time trace and frequency spectrum are shown below:

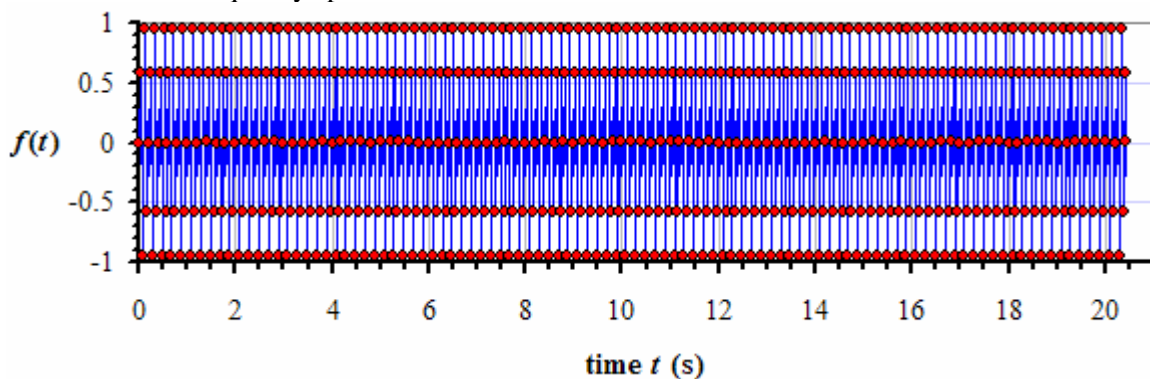


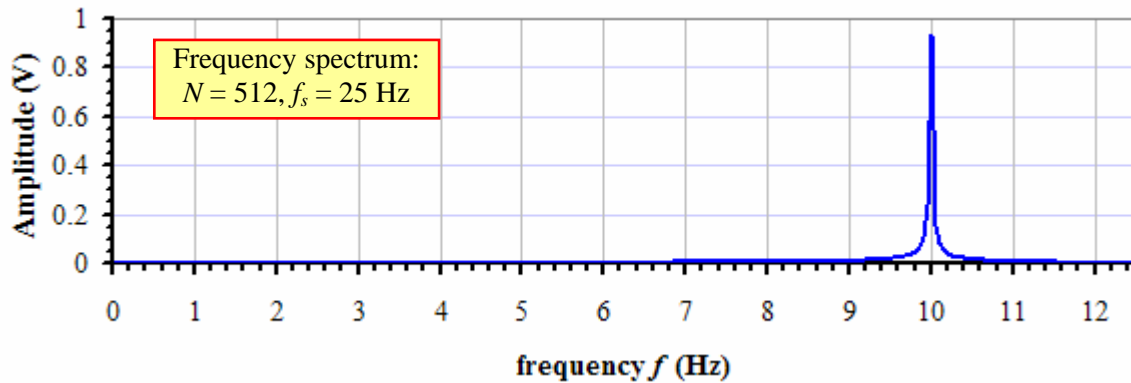
- Some observations about the above time trace and frequency spectrum:
 - Higher sampling frequency has vastly improved the resolution of the sine wave; there are now 100 data points per wavelength instead of just 20 data points per wavelength in the previous case.
 - However, increasing the sampling frequency did *not* improve the frequency spectrum!
 - In fact, the output frequency spectrum has much *poorer frequency resolution*. This can be clearly predicted by the above equations. Namely, the frequency resolution for this case is $\Delta f = 1/T = 1/(0.256 \text{ s}) = 3.91$ Hz (to three significant digits), whereas that of the previous case ($f_s = 200$ Hz) is approximately 0.781 Hz for the same number of data points, $N = 256$.
 - Furthermore, the peak amplitude of the frequency spectrum is now about 0.66, significantly smaller than the known signal amplitude of 1. **This is due to leakage, which is worse in this case because of the poorer frequency resolution.**
 - Even worse, the peak amplitude of the frequency spectrum occurs at a frequency of 11.72 Hz instead of the known 10 Hz. This is also a result of the poor frequency resolution.

- We conclude: **The only thing gained by increasing the sampling frequency is the corresponding increase in the maximum frequency or folding frequency of the frequency spectrum.**
- For this example – a 10 Hz sine wave – a sampling rate of 1000 Hz is *overkill*. In fact, the original 200 Hz sampling rate is also higher than necessary. By the Nyquist criterion, since the maximum frequency in the signal is 10 Hz, the *minimum required sampling rate to avoid aliasing* is 20 Hz.
- The same time signal is again sampled ($N = 256$ data points as before), but this time at a sampling frequency of only 25 Hz. The time trace and frequency spectrum are shown below:



- Some comments about the above time trace and frequency spectrum:
 - Lower sampling frequency has vastly degraded the resolution of the sine wave; there are now only 2.5 data points per wavelength instead of 100 in the previous example.
 - This time, however, the frequency resolution is *much* better (0.0977 Hz). **We have improved the frequency resolution at the expense of poorer time resolution.**
 - While there is still some leakage, the leakage is significant within only about 1 Hz to the right or left of the peak near 10 Hz.
 - The maximum amplitude is about 0.75 V, and occurs at 9.96 Hz – much closer to the true frequency of 10 Hz, due to the improved frequency resolution.
 - This illustrates a very significant, yet often unappreciated fact: **Higher sampling rate is not always better!**
- Can we improve the FFT results any further? The answer is *yes*, but at the cost of more computing time – more data points. The previous case is repeated, except the number of data points is *doubled* to $N = 512$. Everything else remains the same – the sampling frequency is 25 Hz, and the same sine wave is sampled. The time trace and frequency spectrum are shown below:

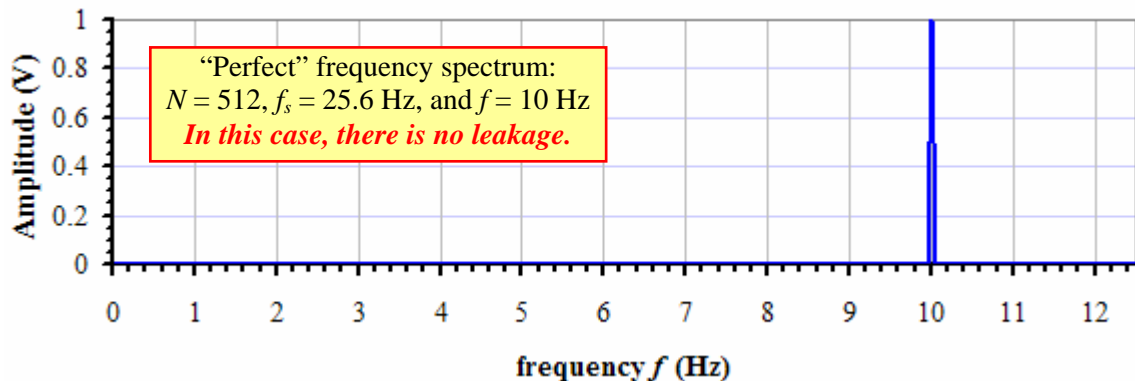




- Some comments:
 - Compared to the previous case, the time resolution has not changed – we still sample 2.5 data points per wavelength or period of the time signal, but we sample for $T = N/f_s = 512/(25 \text{ Hz}) = 20.48 \text{ s}$, twice as long as the previous case, in which $T = 10.24 \text{ s}$.
 - Compared to the previous case, the frequency resolution has improved by a factor of two because the number of discrete data points has increased by a factor of two (for the same sampling frequency).
 - The peak in the spectrum at 10 Hz is much narrower, with greatly reduced leakage.
 - The peak amplitude is now 0.934 V at a frequency of 10.01 Hz. This is *much* closer to the known exact amplitude of 1 V at 10 Hz. **The improvement is due to better frequency resolution and reduced leakage.**

Perfect FFT

- A final comment needs to be made about FFTs and frequency spectra. Namely, it is theoretically possible to obtain a **perfect FFT** (one without any leakage) from sampled data.
- However, **a perfect FFT is possible only if the sampled data begin and end at the same phase of the signal.**
- To illustrate, the above case is repeated one more time, but with a sampling frequency of exactly 25.6 Hz.
- This sampling frequency is chosen such that there is an *integral number of wavelengths in the sampled data set*, so that *the sampled signal starts and ends at exactly the same phase*. In this particular case, $f_s = 25.6$, $N = 512$, $T = N/f_s = \text{exactly } 20 \text{ s}$. This T corresponds to exactly 200 complete cycles for our 10 Hz sine wave.
- **The sampled signal starts and ends at the same (zero) phase.**
- The resulting frequency spectrum is shown below:



- Some comments:
 - The peak amplitude is *exactly* 1 V, and occurs at *exactly* 10 Hz.
 - The FFT is perfectly correct because there is *no leakage* in this case – the FFT output is *exactly* zero at all frequency values except 10 Hz, where its value is $-256i$. The magnitude at this point is 256, and when we divide by $N/2 = 512/2 = 256$, we get a modified amplitude of exactly 1 V – the FFT is perfect.
 - You must keep in mind that this “perfect FFT” is possible only because we know the frequency of the original signal, which is a perfect sine wave, and we select the sampling frequency appropriately.
 - In real-life laboratory situations, the frequency is *not* known in advance (otherwise, an FFT would not even be necessary!), and the signal is *not* a perfect sine wave.
 - In conclusion, although a perfect FFT is possible, it will never be encountered in actual laboratory situations.
- This leads us into the next topic, **windowing**, which is discussed in a separate learning module.