The Discrete Fourier Transform

Claudia Feregrino-Uribe, Alicia Morales-Reyes
Original material: Dr. René Cumplido

CCC-INAOE
Autumn 2015

The Discrete Fourier Transform

- Fourier analysis is a family of mathematical techniques, all based on decomposing signals into sinusoids.
- The discrete Fourier transform (DFT) is the family member used with *digitized* signals.

Contents

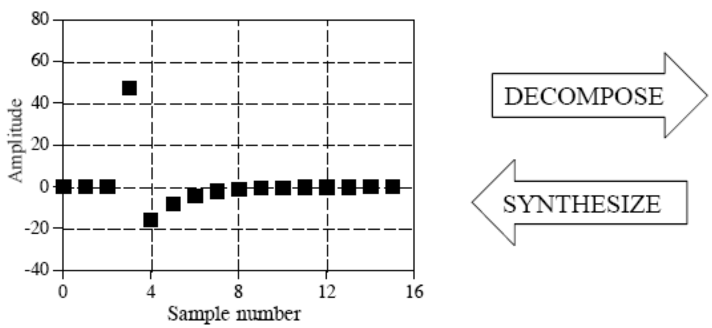
The Family of Fourier Transform

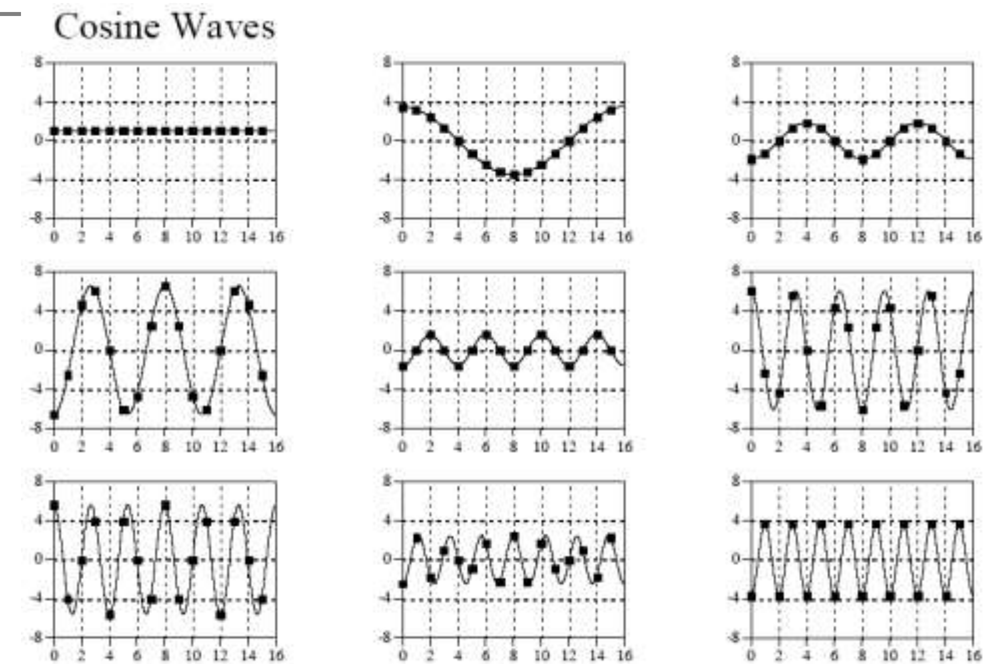
- Fourier analysis is named after Jean Baptiste Joseph Fourier (1768-1830)

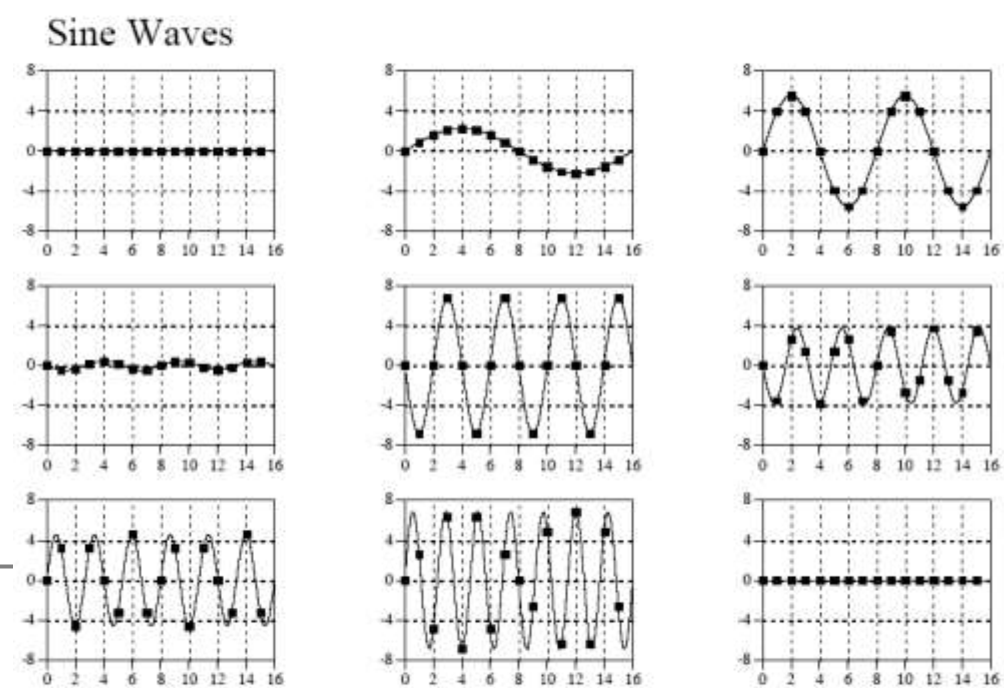
- Fourier was interested in heat propagation, and presented a paper in 1807 on the use of sinusoids to represent temperature distributions.

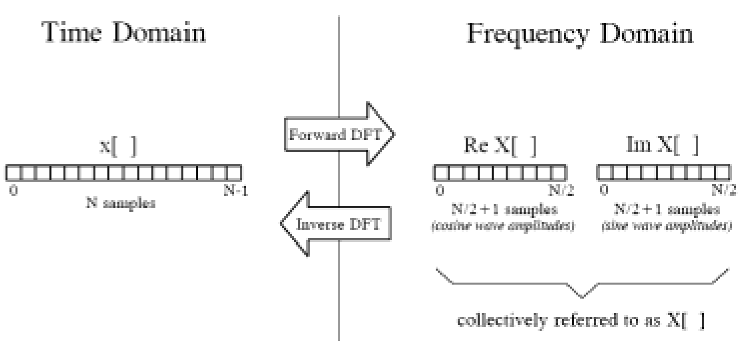
- The paper contained the controversial claim that *any continuous periodic signal could be represented as the sum of properly chosen sinusoidal waves*.

Any signal can be decomposed into sine and cosine waves.









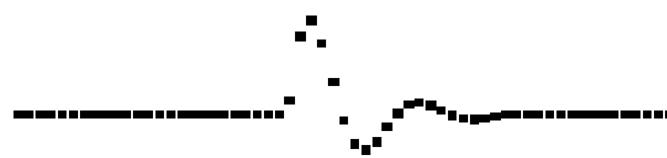
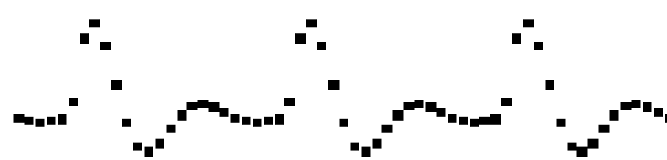
There are an infinite number of ways that a signal can be decomposed. Then, why are sinusoids used instead of, for instance, square or triangular waves?

- The goal of decomposition is to end up with something *easier* to deal with than the original signal.

- The component sine and cosine waves are simpler than the original signal because they have a property: *sinusoidal fidelity.*
  - A sinusoidal input to a system is guaranteed to produce a sinusoidal output.
  - Only the amplitude and phase of the signal can change
  - The frequency and wave shape remain the same.
  - Sinusoids are the only waveform that have this useful property.

Fourier transform

The general term: *Fourier transform*, can be broken into four categories:

Type of Transform	Example Signal
Fourier Transform *signals that are continious and aperiodic*	
Fourier Series *signals that are continious and periodic*	
Discrete Time Fourier Transform *signals that are discrete and aperiodic*	
Discrete Fourier Transform *signals that are discrete and periodic*	

Fourier transform

- There is not a version of the Fourier Transform that uses finite length signals.
- What then if you only have a finite number of samples, say a signal formed from 1024 points?
  - Sine and cosine waves are *defined* as extending from negative infinity to positive infinity.
  - You cannot use a group of infinitely long signals to synthesize something finite in length.
  - The way around is to make the finite data *look like* an infinite length signal.
  - This is done by imagining that the signal has an infinite number of samples on the left and right of the actual points.
  - If all these “imagined” samples have a value of zero, the signal looks *discrete* and *aperiodic*
    - Discrete Time Fourier Transform applies.
  - Or, if the imagined samples can be a duplication of the actual 1024 points. The signal then looks discrete and periodic, with a period of 1024 samples.
    - Discrete Fourier Transform is used.

DFT in DSP

To synthesize a signal that is *aperiodic,* an *infinite* number of sinusoids are required.

- This makes it impossible to calculate the Discrete Time Fourier Transform in a computer algorithm.
- Thus, by elimination the only type of Fourier transform that can be used in DSP is the DFT.
  - Digital computers can only work with information that is *discrete* and *finite* in length.

DFT

- Look back at the example DFT decomposition.
- A 16 point signal is decomposed into 18 sinusoids, each consisting of 16 points.
  - The 16 point signal must be viewed as a single period of an infinitely long periodic signal.
  - Likewise, each of the 18 sinusoids represents a 16 point segment from an infinitely long sinusoid.
  - The key point to understand is that this periodicity is invoked in order to use a *mathematical tool*, i.e., the DFT.

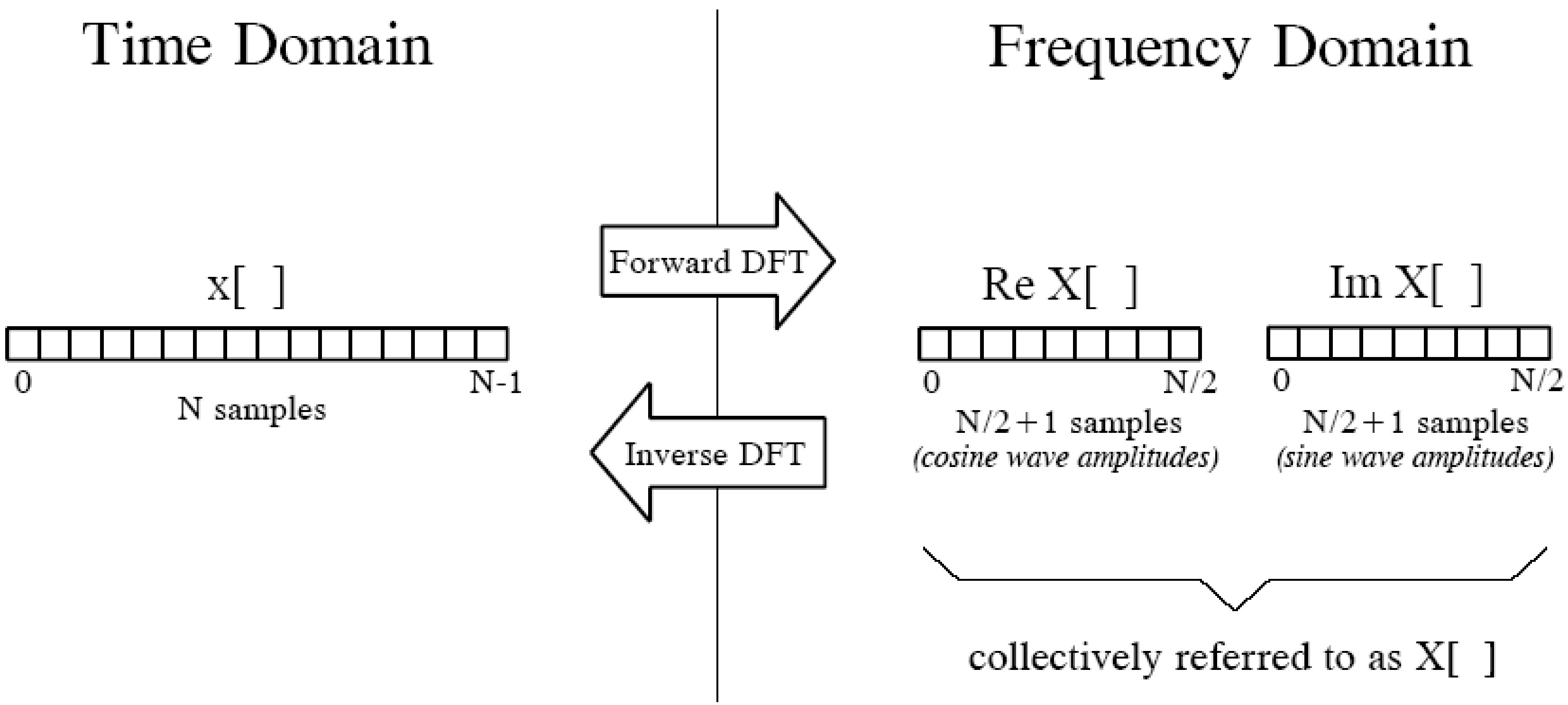
Transforms

- The mathematical term: **transform**, is extensively used in DSP
  - Fourier transform, Laplace transform, Z transform, Hilbert transform, Discrete Cosine transform,
- What is a transform?
  - In short, a transform is any fixed procedure that changes one chunk of data into another chunk of data.

Notation and Format of the Real DFT

- The discrete Fourier transform changes an *N* point input signal into two *N*/2+1 point output signals.
- The input signal contains the signal being decomposed
- The two output signals contain the *amplitudes* of the component sine and cosine waves.
- The input signal is said to be in the **time domain**.
  - The most common type of signal entering the DFT is composed of samples taken at regular intervals of *time*.
- The term **frequency domain** is used to describe the amplitudes of the sine and cosine waves

DFT terminology



DFT terminology

- The frequency domain contains exactly the same information as the time domain, just in a different form.
- If you know one domain, you can calculate the other.
- Given the time domain signal, the process of calculating the frequency domain is called
  - **Decomposition**
  - **Analysis**
  - the **forward DFT**, or
  - **the DFT**
- If you know the frequency domain, calculation of the time domain is called
  - **Synthesis**
  - **Inverse DFT**.

DFT terminology

- The number of samples in the time domain is usually represented by the **variable *N***.
  - A power of two is usually chosen, i.e., 128, 256, 512, 1024, etc.
  - There are two reasons for this
    - digital data storage uses binary addressing
    - The most efficient algorithm for calculating the DFT, the Fast Fourier Transform (FFT), usually operates with *N* that is a power of two.
- Typically, *N* is selected between 32 and 4096. In most cases, the samples run from 0 to *N*-1, rather than 1 to *N*.

DFT terminology

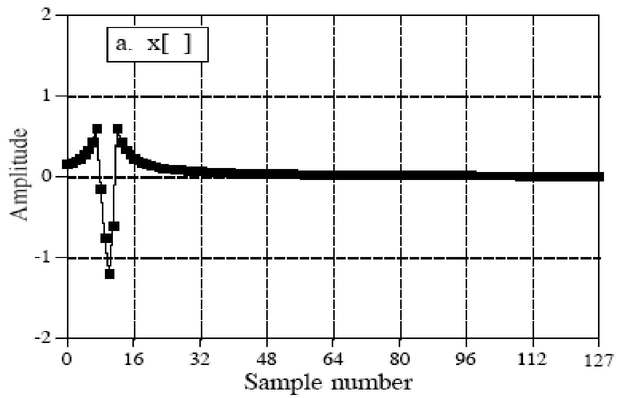
- Standard DSP notation uses **lower case letters** to represent time domain signals, such as $x[]$, $y[]$, $z[]$,
- The corresponding **upper case letters** are used to represent their frequency domains, $X[\ ]$ $Y[\ ]$ $Z[\ ]$
- The frequency domain of this signal consists of two parts, each an array of samples.
  - Real part of $X[\ ]$, written $\mathrm{Re}\ X[\ ]$
  - Imaginary part of $X[\ ]$, written $\mathrm{Im}\ X[\ ]$:
  - The values $Re\ X[\ ]$ are the amplitudes of the cosine waves
  - The values in $Im\ X[\ ]$ are the amplitudes of the sine waves

Complex DFT

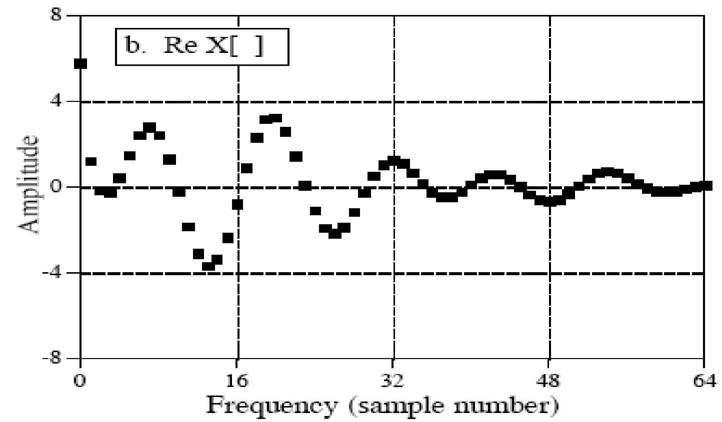
- Complex DFT changes an *N* point time domain signal into an *N* point frequency domain signal.
- Each "point" is a complex number consisting of real and imaginary parts.
- We focus on learning the real DFT

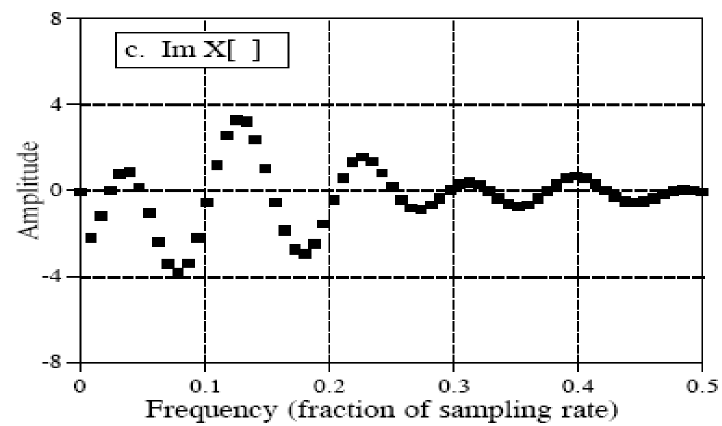
The Frequency Domain's Independent Variable

Time Domain



Frequency Domain





- The time domain signal is an the array: *x*[0] to *x*[127].
- The frequency domain signals are contained in the two arrays: *ReX*[0] to *ReX*[64] and *Im X*[0] to *Im X*[64]
- Notice that 128 points in the time domain corresponds to 65 points in each of the frequency domain signals,
- That is, *N* points in the time domain corresponds to points in the *N*/2+1 frequency domain

Horizontal axis

The horizontal axis of the frequency domain can be referred to in **four different ways**, all common in DSP.

- 1) Horizontal axis is labeled from 0 to 64, corresponding to the 0 to *N*/2 samples in the arrays.
  - Index for the frequency domain is an integer, from 0 to N/2
  - Programmers like this method
- 2) Horizontal axis is labeled as a *fraction of the sampling rate*.
  - The values along the horizonal axis always run between 0 and 0.5
  - discrete data can only contain frequencies between DC and one-half the sampling rate.
  - The index used with this notation is *f*, for frequency.
  - The real and imaginary parts are written: $ReX[f]$ and $ImX[f]$ where f takes on $N/2$ equally spaced values between 0 and 0.5.
- To convert from the first notation, k, to the second notation, f, divide the horizontal axis by *N*. That is, $f=k/N$.
- Remember: Discrete signals only contain frequencies between 0 and 0.5 of the sampling rate.

Horizontal axis

- 3) Horizontal axis is multiplied by 2π.
  - The index used with this labeling is ω
  - In this notation, the real and imaginary parts are written: *ReX* [ω] and *ImX* [ω]
    - where ω takes on $N/2+1$ equally spaced values between 0 and π.
  - The parameter, ω, is called the **natural frequency**, and has the units of **radians**.
    - This is based on the idea that there are 2π radians in a circle.
  - Mathematicians like this method because it makes the equations shorter:
    - $c[n] = \cos(2\pi kn/N)$
    - $c[n] = \cos(2\pi fn)$
    - $c[n] = \cos(\omega n)$

Horizontal axis

- 4) Horizontal axis in terms of the analog frequencies used in a *particular* application.
  - If the system being examined has a sampling rate of 10 kHz
    - graphs of the frequency domain would run from 0 to 5 kHz.
  - This method has the advantage of presenting the frequency data in terms of a *real world* meaning.
  - The disadvantage is that it is tied to a particular sampling rate, and is therefore not applicable to general DSP algorithm development.

DFT Basis Functions

- The sine and cosine waves used in the DFT are commonly called the DFT **basis functions**.
- In other words, the output of the DFT is a set of numbers that represent amplitudes.
- The basis functions are a set of sine and cosine waves with *unity* amplitude.
- By applying each amplitude in the frequency domain to the proper sine or cosine wave (the basis functions), the result is a set of *scaled* sine and cosine waves that can be added to form the time domain signal.

DFT Basis Functions

- The DFT basis functions are generated from the equations:

$$c_k[i] = \cos(2\pi k i / N)$$

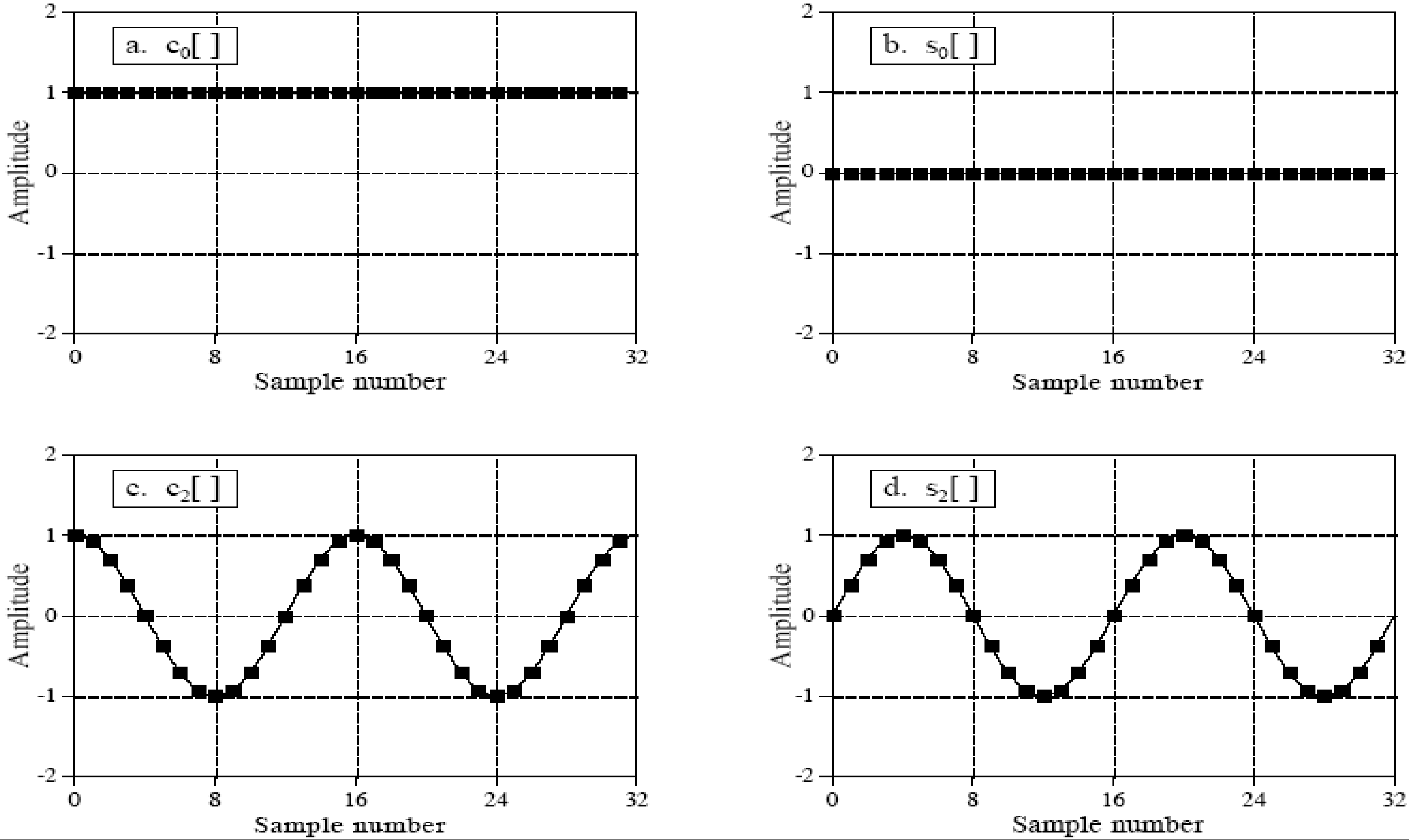
$$s_k[i] = \sin(2\pi k i / N)$$

where:

$c_k[\,]$ is the cosine wave for the amplitude held in $ReX[k]$,
$s_k[\,]$ is the sine wave for the amplitude held in $Im\,X[k]$.
each N points in length, running from 0 to N-1

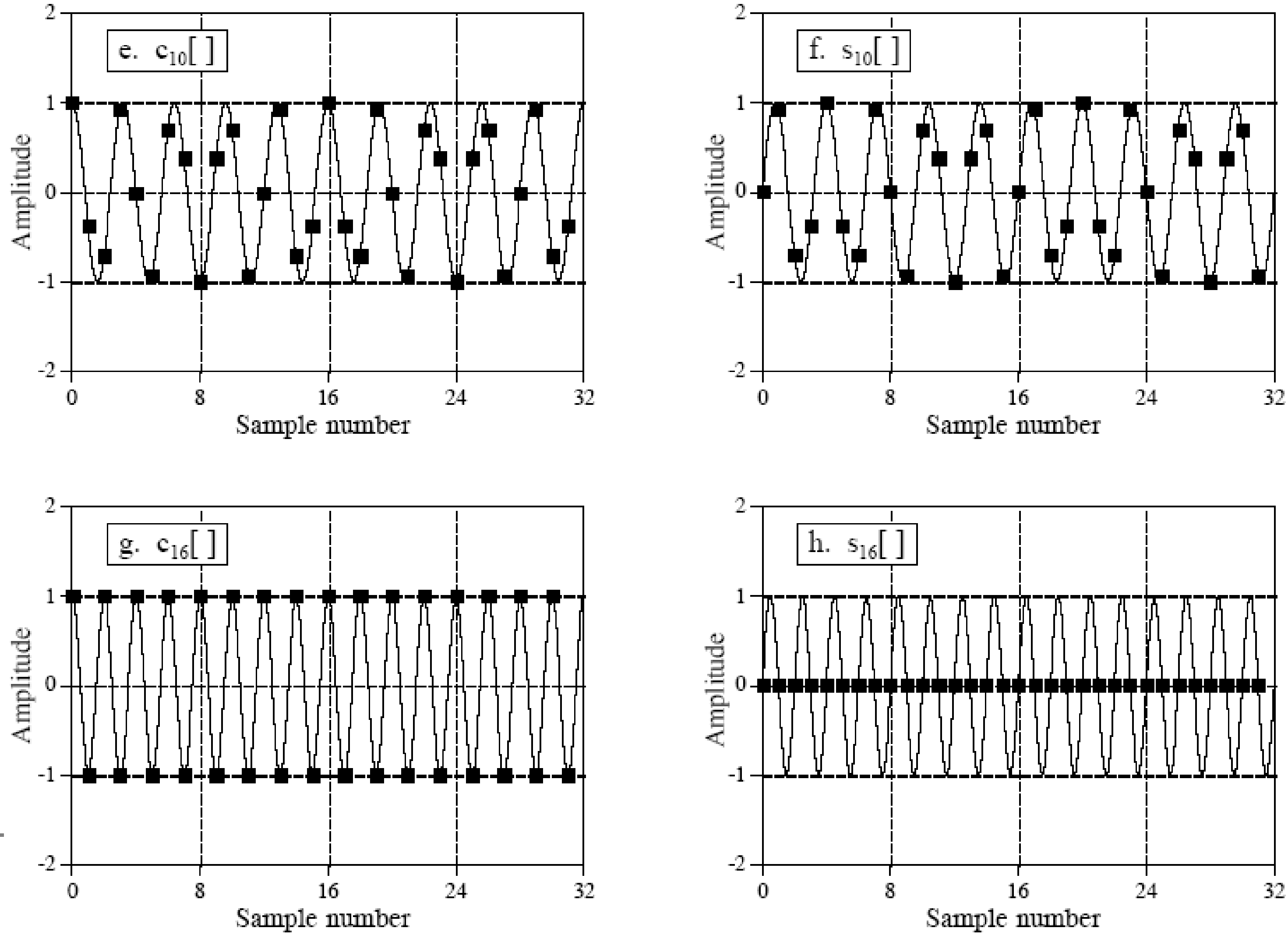
The parameter, *k*, determines the frequency of the wave.
k takes on values between 0 and N/2.

DFT Basis Functions



"The Scientist and Engineer's Guide to Digital Signal Processing, copyright ©1997-1998 by Steven W. Smith."

DFT Basis Functions



Synthesis, Calculating the Inverse DFT

Any N point signal, $x[i]$, can be created by adding cosine $N/2+1$ waves and $N/2+1$ sine waves.

The amplitudes of the cosine and sine waves are held in the arrays $Re\bar{X}[k]$ and $Im\bar{X}[k]$, respectively.

The synthesis equation multiplies these amplitudes by the basis functions to create a set of scaled sine and cosine waves.

Adding the scaled sine and cosine waves produces the time domain signal.

$$x[i] = \sum_{k=0}^{N/2} Re\bar{X}[k] \cos(2\pi ki/N) + \sum_{k=0}^{N/2} Im\bar{X}[k] \sin(2\pi ki/N)$$

Synthesis, Calculating the Inverse DFT

Conversion between the sinusoidal amplitudes and the frequency domain values.

$$Re\bar{X}[k] = \frac{ReX[k]}{N/2}$$

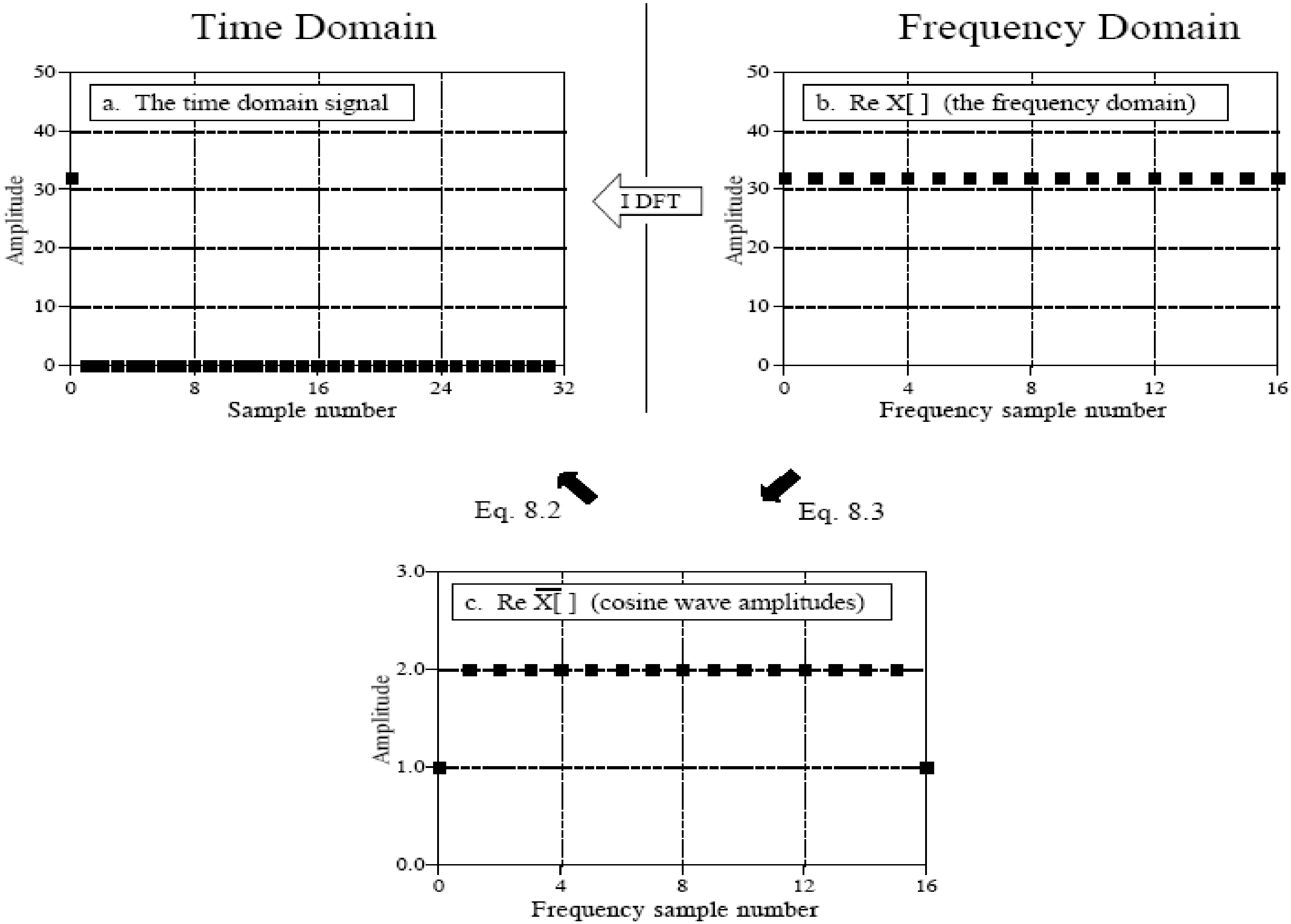
$$Im\bar{X}[k] = -\frac{ImX[k]}{N/2}$$

except for two special cases:

$$Re\bar{X}[0] = \frac{ReX[0]}{N}$$

$$Re\bar{X}[N/2] = \frac{ReX[N/2]}{N}$$

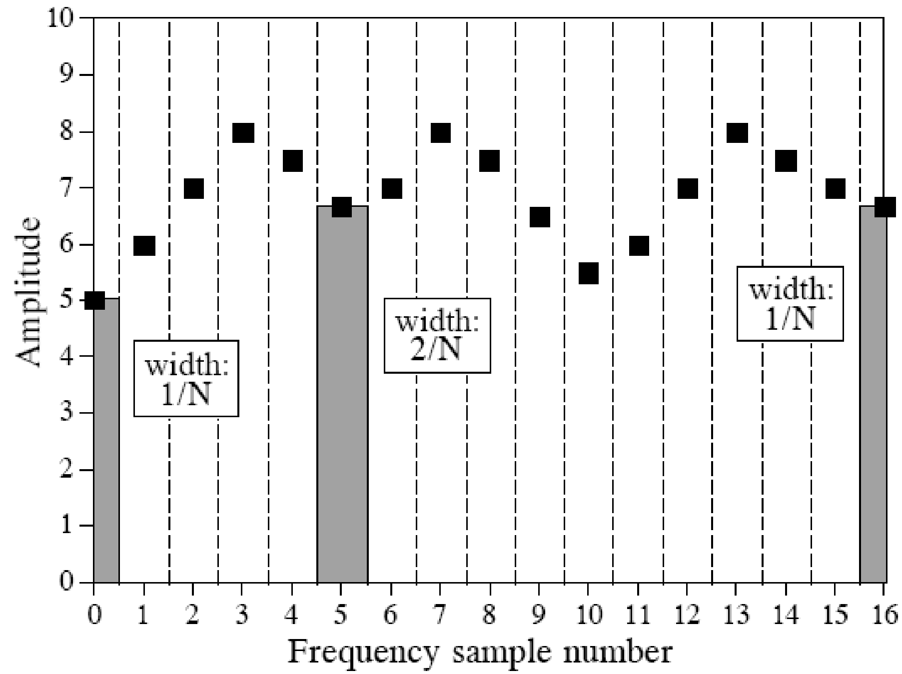
Inverse DFT



"The Scientist and Engineer's Guide to Digital Signal Processing, copyright ©1997-1998 by Steven W. Smith."

Bandwidth of frequency domain samples

- Each sample in the frequency domain is contained in a frequency band of width 2/*N*.
- The first and last samples, which have a bandwidth only one-half this wide, 1/*N*.



Analysis, Calculating the DFT

The DFT can be calculated in three completely different ways

- 1) *Simultaneous equations*. This method is useful for understanding the DFT, but it is too inefficient to be of practical use.
- 2) *Correlation.* This is based on detecting a known waveform in another signal.
- 3) Fast Fourier Transform (FFT), is an ingenious algorithm that decomposes a DFT with *N* points, into *N* DFTs each with a single point.
  - The FFT is typically hundreds of times faster than the other methods.
- The first two methods are discussed here, while the FFT will be discussed later.
- All three of these methods produce an identical output. Which should you use?
  - *Correlation* is the preferred technique if the DFT has less than about 32 points, otherwise the *FFT* is used.

DFT by Simultaneous Equations

Think about the DFT calculation in the following way.

1. You are given *N* values from the time domain, and asked to calculate the *N* values of the frequency domain (ignoring the two frequency domain values that you know must be zero).
2. To solve for *N* unknowns, you must be able to write *N* linearly independent equations.
3. To do this, take the first sample from each sinusoid and add them together. The sum must be equal to the first sample in the time domain signal, thus providing the first equation.
4. An equation can be written for each of the remaining points, resulting in the required *N* equations.
5. The solution can then be found by using established methods for solving simultaneous equations, such as Gauss Elimination.

- Unfortunately, this method requires a tremendous number of calculations, and is virtually never used in DSP.
- However, it shows *why* it is possible to decompose a signal into sinusoids, how *many* sinusoids are needed, and that the basis functions must be linearly independent.

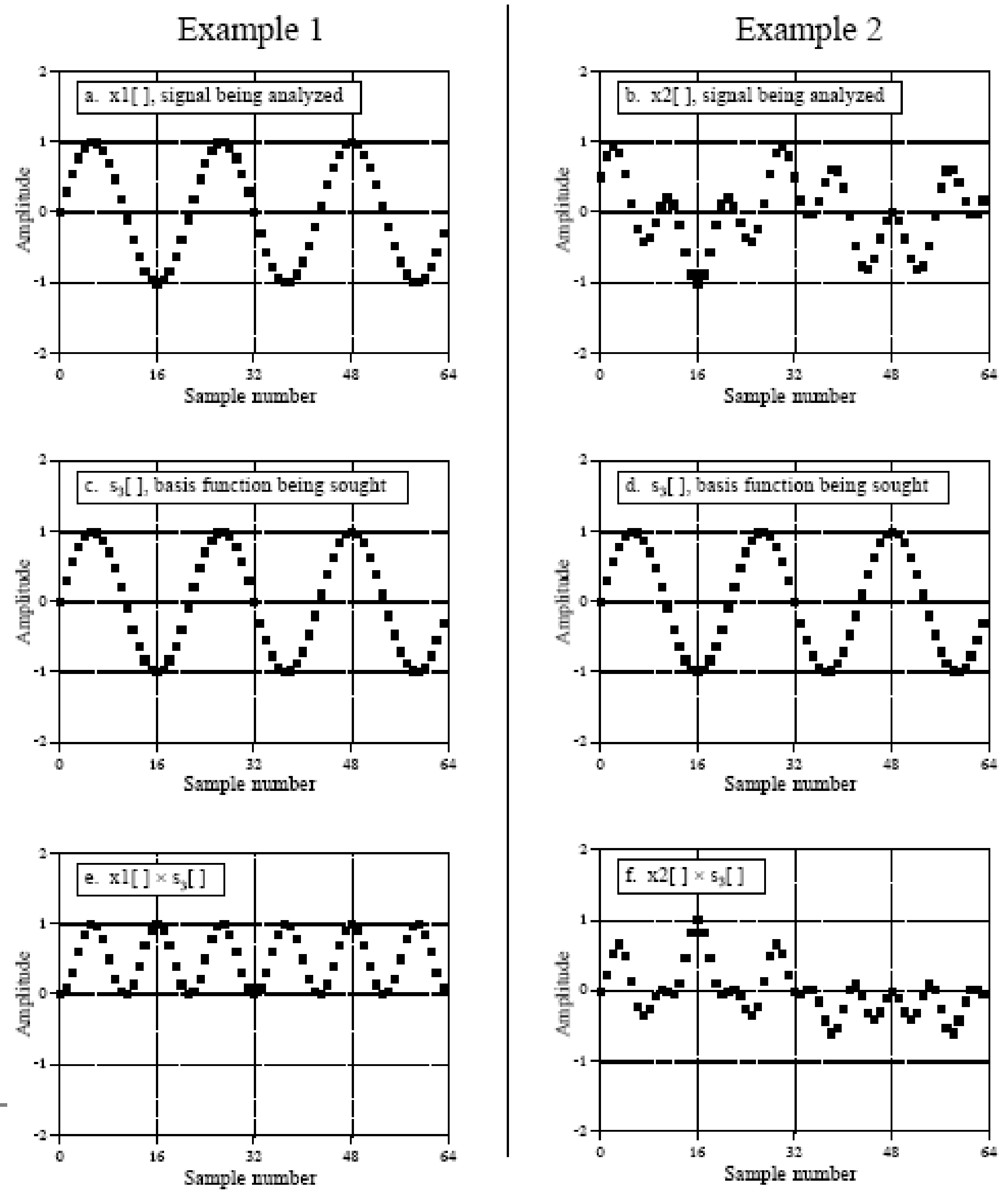
DFT by Correlation

- Suppose we are trying to calculate the DFT of a 64 point signal.
- This means we need to calculate the 33 points in the real part, and the 33 points in the imaginary part of the frequency domain.
- In this example we will calculate the sample, $Im\ X[3]$, i.e., the amplitude of the sine wave that makes three complete cycles between point 0 and point 63.

DFT by Correlation

Figure illustrates using correlation to calculate *Im X*[3].

- a) and b) show two example time domain signals
- *x1*[ ] is composed of a sine wave that makes three cycles between from 0 to63.
- *x2*[ ] is composed of several sine and cosine waves, *none* of which make three cycles from 0 to 63.
- These two signals illustrate what the algorithm for calculating must do.
- When fed *x1*[ ], the algorithm must produce a value of 32, the amplitude of the sine wave present in the signal.
- In comparison, when the algorithm is fed the other signal, *x2*[ ], a value of zero must be produced, indicating that this sine wave is not present in this signal.



DFT by Correlation

- Correlation detects a known waveform contained in another signal, multiply the two signals and add all the points in the resulting signal.
- The single number that results from this procedure is a measure of how similar the two signals are.
- Figures (c) and (d) both display the signal we are looking for, a sine wave that makes 3 cycles between samples 0 and 63.
- Figure (e) shows the result of multiplying (a) and (c) and (f) shows the result of multiplying (b) and (d).
- The sum of all the points in (e) is 32, while the sum of all the points in (f) is zero, showing we have found the desired algorithm for the DFT.

DFT by Correlation

This procedure is formalized in the *analysis equation*, the mathematical way to calculate the frequency domain from the time domain:

$$Re X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$Im X[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

In words, each sample in the frequency domain is found by multiplying the time domain signal by the sine or cosine wave being looked for, and adding the resulting points.

Orthogonal basis functions

- In order for this correlation algorithm to work, the basis functions must have an interesting property:
  - Each of them must be completely *uncorrelated* with all of the others.
- This means that if you multiply any two of the basis functions, the sum of the resulting points will be equal to zero.
- Basis functions that have this property are called **orthogonal**.
- Many other orthogonal basis functions exist, including:
  - square waves
  - triangle waves
  - Impulses
  - Signals can be decomposed into these other orthogonal basis functions using correlation, just as done here with sinusoids.
  - This is not to suggest that this is *useful*, only that it is *possible*.

Duality

- The synthesis and analysis equations are similar.
  - To move from one domain to the other, the known values are multiplied by the basis functions, and the resulting products added.
- The fact that the *DFT* and the *Inverse DFT* use this same mathematical approach is really quite remarkable, considering the totally different way we arrived at the two procedures.
- In fact, the only significant difference between the two equations is a result of the time domain being *one* signal of *N* points, while the frequency domain is *two* signals of *N*/2+1 points.
- This makes the two domains completely symmetrical, and the equations for moving between them virtually *identical*.
- This symmetry between the time and frequency domains is called **duality**
- **Duality** has some interesting properties.
  - A single point in the frequency domain corresponds to a sinusoid in the time domain.
    - By duality, the inverse is also true, a single point in the time domain corresponds to a sinusoid in the frequency domain.
  - Convolution in the time domain corresponds to multiplication in the frequency domain.
    - By duality, the reverse is also true: convolution in the frequency domain corresponds to multiplication in the time domain.

Polar notation

- The frequency domain is a group of amplitudes of cosine and sine waves (with slight scaling modifications).
  - This is called **rectangular** notation.
- Alternatively, the frequency domain can be expressed in **polar** form.
  - In this notation, there are also two arrays, now called
    - **Magnitude of *X*[ ]**, written in equations as: ***Mag X*[ ]**, and
    - **Phase of *X*[ ]**, written as: ***Phase X*[ ]**
- The magnitude and phase are a pair-for-pair replacement for the real and imaginary parts.
  - For example, $Mag\ X[0]$ and $Phase\ X[0]$ and are calculated using only $Re\ X[0]$ and $Im\ X[0]$

Polar notation

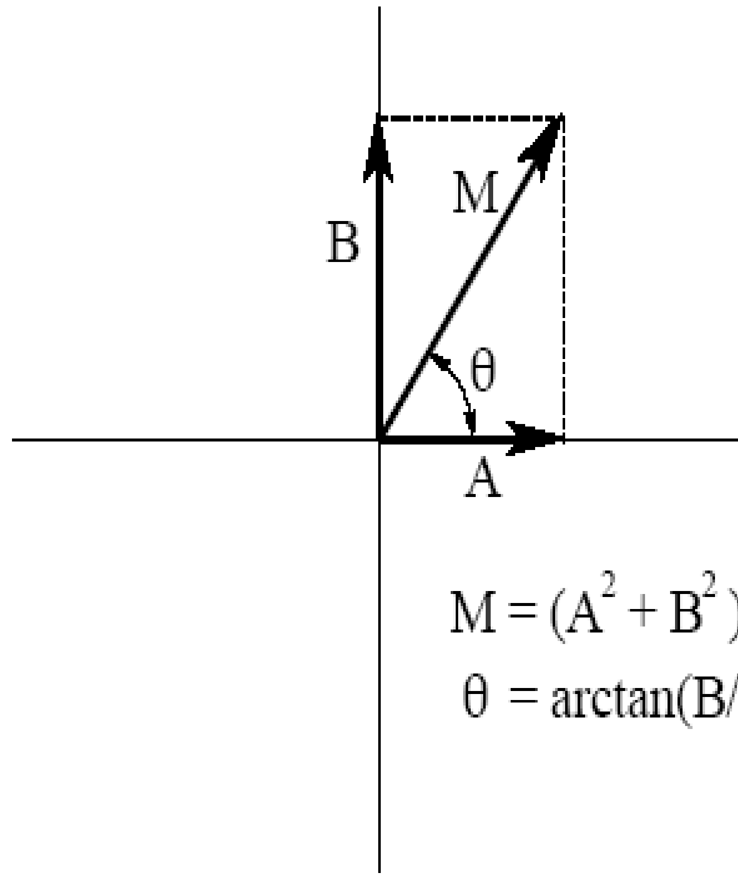
- To understand the conversion, consider what happens when you add a cosine wave and a sine wave of the same frequency.
  - The result is a cosine wave of the same frequency, but with a new amplitude and a new phase shift.
  - In equation form, the two representations are related:

$$A\cos(x) + B\sin(x) = M\cos(x + \theta)$$

No information is lost in this process; given one representation you can calculate the other. In other words, the information contained in the amplitudes A and B, is also contained in the variables M and θ.

Polar notation

- The equation follows the same conversion equations as do simple vectors.
- Figure shows analogous vector representation of how the two variables, A and B, can be viewed in a rectangular coordinate system, while M and θ are parameters in polar coordinates.



$$M = (A^2 + B^2)^{1/2}$$
$$\theta = \arctan(B/A)$$

Polar notation

- In polar notation, *Mag X*[ ] holds the amplitude of the cosine wave, while *Phase X*[ ] holds the phase angle of the cosine wave
- The following equations convert the frequency domain from rectangular to polar notation, and vice versa:

$$MagX[k] = \left(ReX[k]^2 + ImX[k]^2 \right)^{1/2}$$

$$PhaseX[k] = \arctan\left(\frac{ImX[k]}{ReX[k]} \right)$$

$$ReX[k] = MagX[k] \cos(PhaseX[k])$$

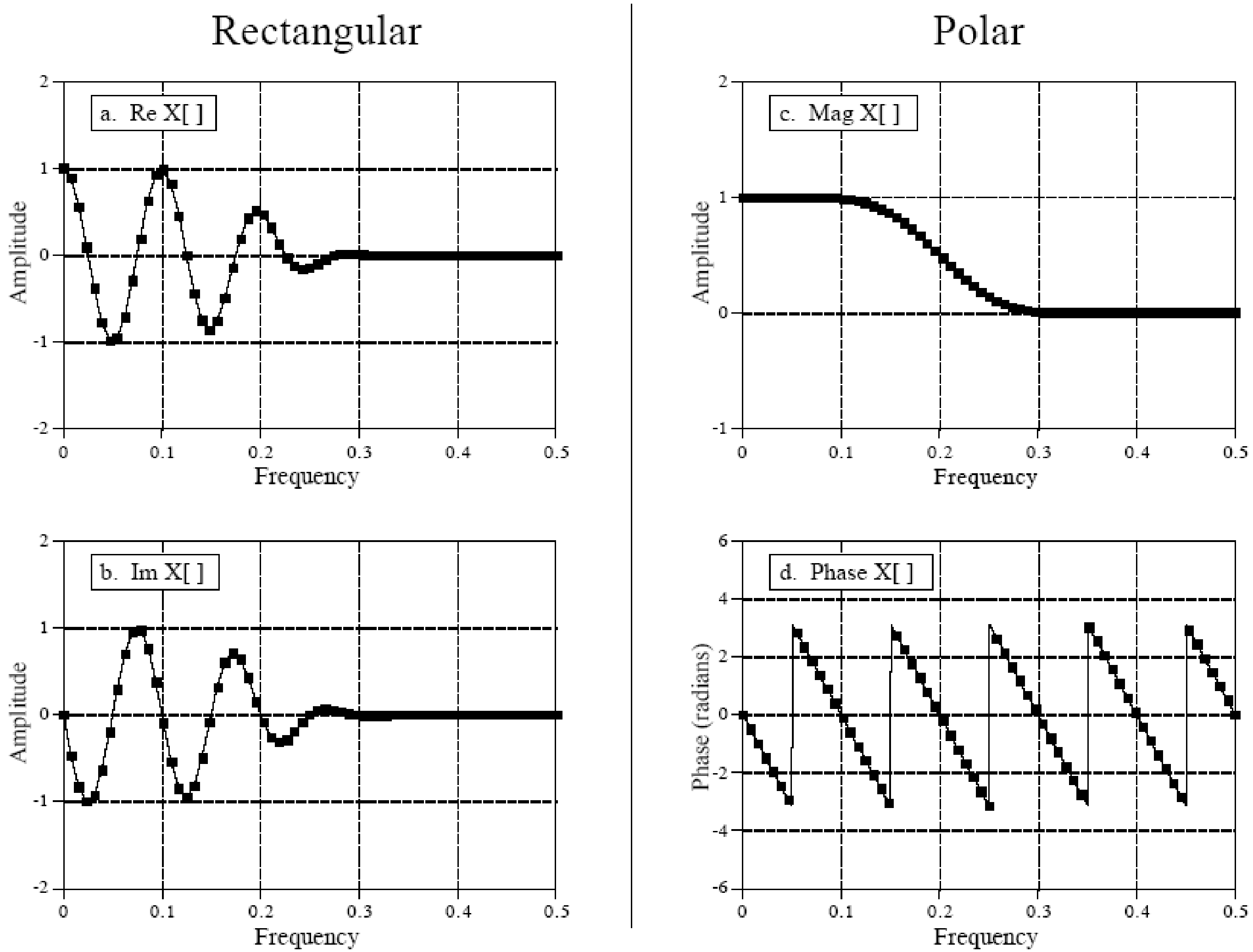
$$ImX[k] = MagX[k] \sin(PhaseX[k])$$

Polar notation

- Rectangular and polar notation allow you to think of the DFT in two different ways.
  - 1) With rectangular notation, the DFT decomposes an *N* point signal into $N/2+1$ cosine waves and $N/2+1$ sine waves, each with a specified *amplitude*.
  - 2) In polar notation, the DFT decomposes an *N* point signal into $N/2+1$ cosine waves, each with a specified *amplitude* (*magnitude*) and *phase shift*.
- Why does polar notation use cosine waves instead of sine waves?
  - Sine waves cannot represent the DC component of a signal, since a sine wave of zero frequency is composed of all zeros

Polar notation

E.g. Frequency response of a low-pass filter.



Polar Nuisances

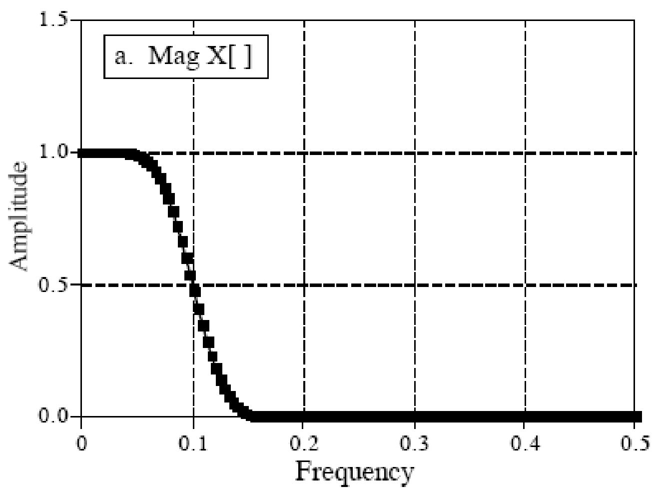
- 1) Radians vs. Degrees
- 2) Divide by zero error
  - The real part is zero and the imaginary part is some nonzero value

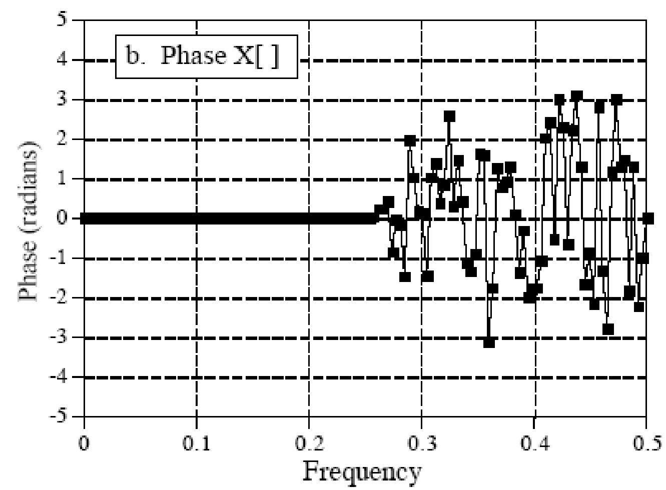
$$Phase\,X[k] = \arctan(Im\,X[k] \,/\, Re\,X[k])$$

- 3) Incorrect arctan.
  - This error occurs whenever the real part is negative.

Polar Nuisances

- 4) Phase of very small magnitudes

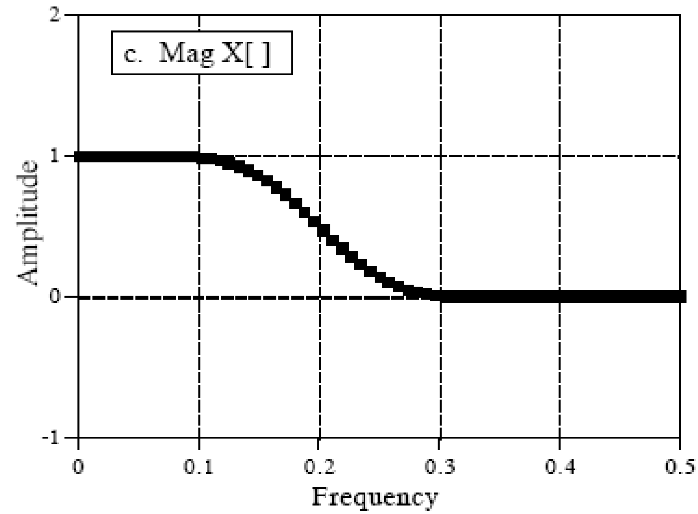


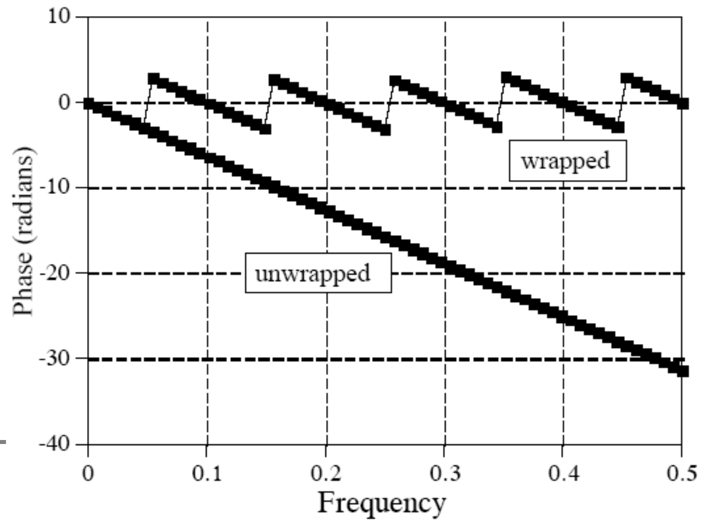


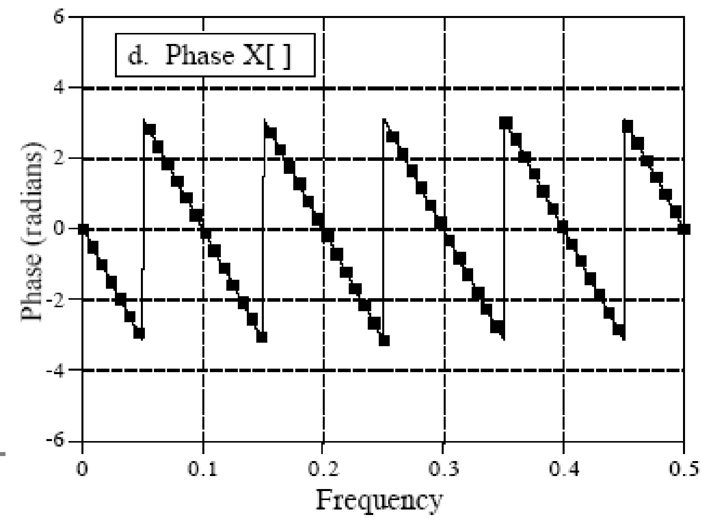
Polar Nuisances

- 5) 2π ambiguity of the phase

Polar







Polar Nuisances

- 6) The magnitude is always positive
  - Since the magnitude must always be positive (by definition), the magnitude and phase may contain abrupt discontinuities and sharp corners.
- 7) Spikes between π and -π
  - Random noise can cause the phase to rapidly oscillate between π or -π.