Notes on the Discrete Fourier Transform

$$X_k \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i k n/N}, \quad k \in \mathbb{Z}$$

$$|X_k|/N = \sqrt{\mathrm{Re}(X_k)^2 + \mathrm{Im}(X_k)^2}/N$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i 2\pi k n/N}, \quad n \in \mathbb{Z}$$

Fourier Series (1)

• Any periodic function can be broken up into an infinite Series of sines and cosines … *using Fourier series*

"Harmonic Coefficients"

$$y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty}\left(A_n \cos n\omega_0 t + B_n \sin n\omega_0 t\right)$$ ***(Fourier Series)***

"Harmonic Order" ***"Fundamental frequency"***

$$\begin{bmatrix} A_n = \left[\frac{\omega}{\pi}\right]\int_0^{\frac{2\pi}{\omega}} y(\tau)\cos(n\omega_0\tau)d\tau \\ B_n = \left[\frac{\omega}{\pi}\right]\int_0^{\frac{2\pi}{\omega}} y(\tau)\sin(n\omega_0\tau)d\tau \end{bmatrix} = \begin{bmatrix} A_n = \left[\frac{2}{T}\right]\int_0^{T} y(\tau)\cos(\frac{2\pi}{T}n\tau)d\tau \\ B_n = \left[\frac{2}{T}\right]\int_0^{T} y(\tau)\sin(\frac{2\pi}{T}n\tau)d\tau \end{bmatrix}$$

• T --> period

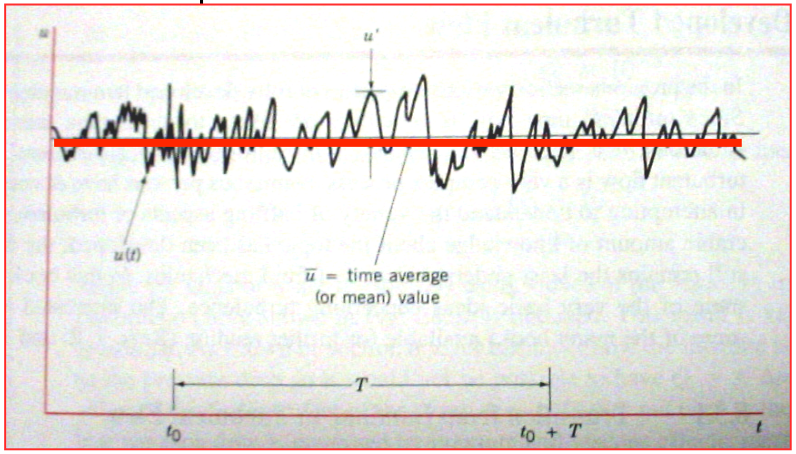
$\omega_0 = 2\pi f_0 = 2\pi / T$

• ***$A_0/2$ is the "mean" or DC level of the signal***

$$\frac{A_0}{2} = \frac{1}{2}\left[\frac{2}{T}\right]\int_0^{t} y(\tau)\cos(0\cdot\omega_0\tau)d\tau \rightarrow$$

$$\frac{A_0}{2} = \left[\frac{1}{T}\right]\int_0^{t} y(\tau)d\tau = y_{mean}$$

y(0)----->



Generalized Waveforms: Fourier Series (2)

- Any periodic wave form may be represented by its Fourier Series

"Harmonic Coefficients"

$$y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty}\left(A_n \cos n\omega_0 t + B_n \sin n\omega_0 t\right)$$ ***(Fourier Series)***

"Harmonic Order" *"Fundamental frequency"*

"Harmonic Coefficients"

$$\begin{bmatrix} A_n = \left[\dfrac{\omega_0}{\pi}\right] \displaystyle\int_0^{\frac{2\pi}{\omega_0}} y(\tau)\cos(n\omega_0\tau)d\tau \\ B_n = \left[\dfrac{\omega_0}{\pi}\right] \displaystyle\int_0^{\frac{2\pi}{\omega_0}} y(\tau)\sin(n\omega_0\tau)d\tau \end{bmatrix}$$

- Can be used to represent Some pretty interesting waveforms

$$f_0 = \text{"}\,fundamental\ frequency\,\text{"}$$

$$f_0 = \frac{1}{T} \to \omega_0 = 2\pi f_0 = \frac{2\pi}{T}$$

$$\boxed{\to \frac{2\pi}{\omega_0} = T} \to \boxed{\frac{\omega_0}{\pi} = \frac{T}{2}}$$

Fourier Series (3)

- Periodic Waveform, T = *Waveform Period*

$$f_0 = \text{"}\textit{fundamental frequency}\text{"}$$

$$f_0 = \frac{1}{T} \rightarrow \omega_0 = 2\pi f_0 = \frac{2\pi}{T}$$

$$\boxed{\rightarrow \frac{2\pi}{\omega_0} = T} \rightarrow \boxed{\frac{\omega_0}{\pi} = \frac{2}{T}}$$

"equivalent forms"

$$\left[\begin{array}{l} A_n = \left[\dfrac{\omega_0}{\pi}\right] \displaystyle\int_0^{\frac{2\pi}{\omega_0}} y(\tau)\cos(n\omega_0\tau)d\tau = \left(\dfrac{2}{T}\right)\int_0^T y(\tau)\cos\left(\dfrac{2\pi}{T}\cdot n\cdot\tau\right)d\tau \\ B_n = \left[\dfrac{\omega_0}{\pi}\right] \displaystyle\int_0^{\frac{2\pi}{\omega_0}} y(\tau)\sin(n\omega_0\tau)d\tau = \left(\dfrac{2}{T}\right)\int_0^T y(\tau)\sin\left(\dfrac{2\pi}{T}\cdot n\cdot\tau\right)d\tau \end{array}\right]$$

Applications of Fourier Analysis

Can be used for four cases:

1) The function y(t) is known analytically (Fourier series)
2) The function y(t) is an analog signal, electronic equipment can be purchased to perform this analysis *(old-school tech)*
3) **The waveform is digitally sampled and y(t) is stored for discrete points in time. Discrete equations are required for numerical computations …**

   "Discrete Fourier Transform"
4) Processing of the Spectrum Allows Data to be Filtered!

The Complex Fourier Transform

Consider the A, B Fourier Series Coefficients

$$\left[\begin{array}{l} A_n = \left[\frac{2}{T}\right] \int_0^T y(\tau)\cos\left(\frac{2\pi}{T} n\tau\right) d\tau \\ B_n = \left[\frac{2}{T}\right] \int_0^T y(\tau)\sin\left(\frac{2\pi}{T} n\tau\right) d\tau \end{array}\right]$$

• T --> period

$\omega_0 = 2\pi f_0 = 2\pi/T$

$$let \rightarrow Y_n = \frac{A_n - j\cdot B_n}{2} \rightarrow where, \boxed{\begin{array}{c} j=\sqrt{-1} \\ j^2=-1 \\ \frac{1}{j} = -j \end{array}}$$

Further $\rightarrow$ define complex exponential as

$$e^{j\cdot\theta} = \cos\theta - j\cdot\sin\theta$$

$$\rightarrow equivalently... \boxed{\begin{array}{c} \cos\theta = \dfrac{e^{j\cdot\theta} + e^{-j\cdot\theta}}{2} \\ \sin\theta = \dfrac{e^{j\cdot\theta} - e^{-j\cdot\theta}}{2\cdot j} \end{array}}$$

$$Y_n = \frac{1}{2}\cdot\left[\frac{2}{T}\right]\left[\begin{array}{l} \int_0^T y(\tau)\cos\left(\frac{2\pi}{T} n\tau\right) d\tau \\ -j\cdot\left[\frac{2}{T}\right] \int_0^T y(\tau)\sin\left(\frac{2\pi}{T} n\tau\right) d\tau \end{array}\right] =$$

$\rightarrow$

$$\left[\frac{1}{T}\right]\int_0^T y(\tau)\left[\cos\left(\frac{2\pi}{T} n\tau\right) - j\cdot\sin\left(\frac{2\pi}{T} n\tau\right)\right] d\tau =$$

$$\left[\frac{1}{T}\right]\int_0^T y(\tau)\cdot e^{-j\cdot\frac{2\pi}{T} n\tau}\cdot d\tau$$

The Complex Fourier Transform (2)

Substitute complex exponential for sin and cosine terms in Fourier series

$$y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty}\left(A_n \cos n\omega_0 t + B_n \sin n\omega_0 t\right) =$$

$$\frac{A_0}{2} + \sum_{n=1}^{\infty}\left(A_n \cdot \frac{e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)} + e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}}{2} + B_n \cdot \frac{e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)} - e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}}{2\cdot j}\right) =$$

$$\frac{A_0}{2} + \sum_{n=1}^{\infty}\left(\frac{1}{2}\left(A_n - j\cdot B_n\right)e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) + \sum_{n=1}^{\infty}\left(\frac{1}{2}\left(A_n + j\cdot B_n\right)e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right)$$

Look at second term ….

$$\sum_{n=1}^{\infty}\left(\frac{1}{2}\left(A_n + j\cdot B_n\right)e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) = \sum_{n=-\infty}^{-1}\left(\frac{1}{2}\left(A_{-n} + j\cdot B_{-n}\right)e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right)$$

The Complex Fourier Transform (3)

From previous slide

$$\sum_{n=1}^{\infty}\left(\frac{1}{2}(A_n + j\cdot B_n)e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) = \sum_{\eta=-1}^{-\infty}\left(\frac{1}{2}(A_{-n} + j\cdot B_{-n})e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right)$$

From Fourier Coefficient definition

$$\left[\begin{array}{l} A_{-n} = \left[\frac{2}{T}\right]\int_0^T y(\tau)\cos(-\frac{2\pi}{T}n\cdot\tau)d\tau = \left[\frac{2}{T}\right]\int_0^T y(\tau)\cos(\frac{2\pi}{T}n\cdot\tau)d\tau = A_n \\ B_{-n} = \left[\frac{2}{T}\right]\int_0^T y(\tau)\sin(-\frac{2\pi}{T}n\cdot\tau)d\tau = -\left[\frac{2}{T}\right]\int_0^T y(\tau)\sin(\frac{2\pi}{T}n\cdot\tau)d\tau = -B_n \end{array}\right]$$

Thus from above

$$\sum_{n=1}^{\infty}\left(\frac{1}{2}(A_n + j\cdot B_n)e^{-j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) = \sum_{n=-\infty}^{-1}\left(\frac{1}{2}(A_n - j\cdot B_n)e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right)$$

The Complex Fourier Transform (4)

Further, since …. $\frac{A_0}{2} = \frac{1}{2}\cdot\left(A_0 - j\cdot B_0\right)\ldots because\ldots B_0 = \left[\frac{2}{T}\right]\int_0^T y(\tau)\sin\left(\frac{2\pi}{T}\cdot 0\cdot\tau\right)d\tau = 0$

Thus …. $y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty}\left(A_n\cos n\omega_0 t + B_n \sin n\omega_0 t\right) =$

$$\frac{1}{2}\left(A_0 - j\cdot B_0\right) + \sum_{n=1}^{\infty}\left(\frac{1}{2}\left(A_n - j\cdot B_n\right)e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) + \sum_{n=-\infty}^{1}\left(\frac{1}{2}\left(A_n - j\cdot B_n\right)e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) =$$

$$\sum_{n=-\infty}^{\infty}\left(\frac{\left(A_n - j\cdot B_n\right)}{2}e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right) = \sum_{n=-\infty}^{\infty}\left(Y_n\cdot e^{j\cdot\left(\frac{2\pi}{T}\cdot n\cdot t\right)}\right)$$

Complex Fourier Series Equivalent …

$$\boxed{\begin{gathered} Y_n = \left[\frac{1}{T}\right]\int_0^T y(\tau)\cdot e^{-j\cdot\frac{2\pi}{T}\cdot n\cdot\tau}\cdot d\tau \\ y(t) = \sum_{n=-\infty}^{\infty}\left(Y_n\cdot e^{j\cdot\frac{2\pi}{T}\cdot n\cdot t}\right)\end{gathered}}$$

The Complex Fourier Transform (5)

Compare series representations

Real Series …..

$$y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty}\left[A_n \cos\left(\frac{2\pi}{T} n \cdot t\right) + B_n \sin\left(\frac{2\pi}{T} n \cdot t\right)\right]$$

$$\left[\begin{array}{l} A_n = \left[\frac{2}{T}\right] \int_0^T y(\tau)\cos\left(\frac{2\pi}{T} n\tau\right) d\tau \\ B_n = \left[\frac{2}{T}\right] \int_0^T y(\tau)\sin\left(\frac{2\pi}{T} n\tau\right) d\tau \end{array}\right]$$

$$A_n = 2 \cdot R_e\left(Y_n\right)$$

$$B_n = -2 \cdot \mathrm{Im}\left(Y_n\right)$$

Complex Series ….

$$Y_n = \frac{A_n - j \cdot B_n}{2}$$

$$Y_n = \left[\frac{1}{T}\right] \int_0^T y(\tau) \cdot e^{-j \cdot \frac{2\pi}{T} \cdot n \cdot \tau} \cdot d\tau$$

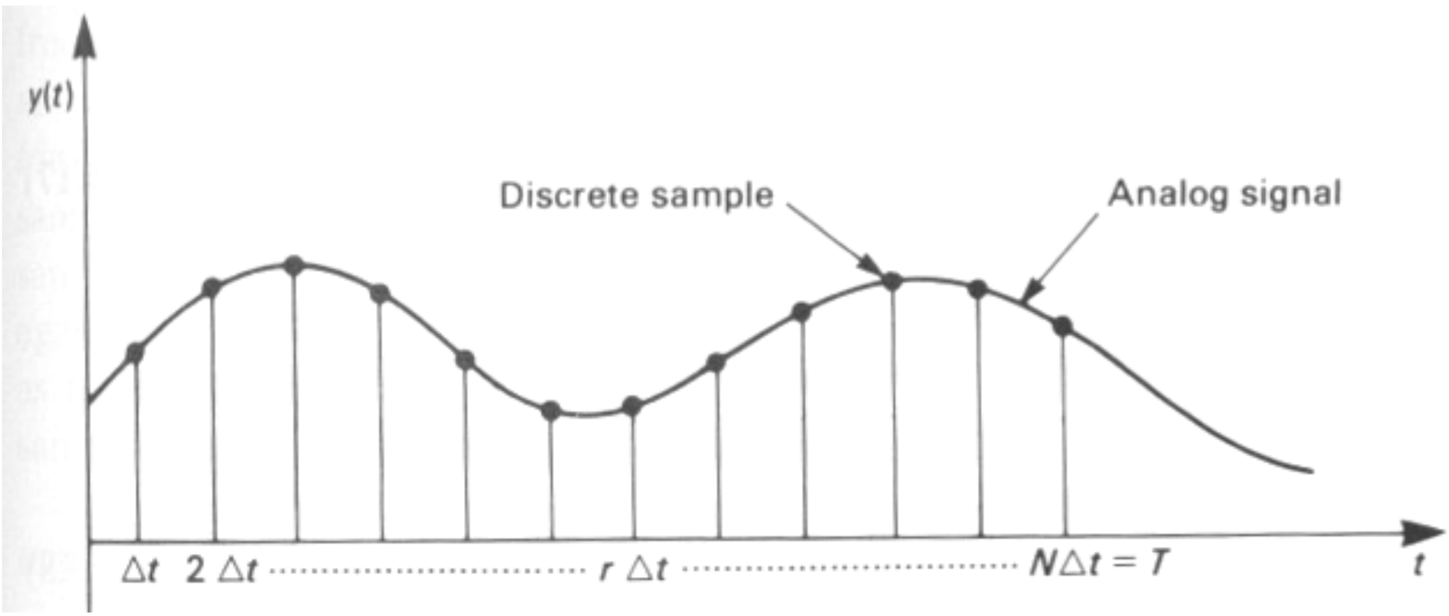
$$y(t) = \sum_{n=-\infty}^{\infty}\left(Y_n \cdot e^{j \cdot \frac{2\pi}{T} \cdot n \cdot t}\right)$$

"Mathematically Identical"

Complex series is far easier to implement on a computer

Discrete Fourier Transform

Look at Discrete Signal Sampled at time interval *ΔT*



$$y_k = \left\{ y_0 , y_{1\cdot\Delta t} , y_{2\cdot\Delta t} + \ldots y_{N\cdot\Delta t} \right\}$$

$$k \rightarrow \text{"}time\ index\text{"} \qquad \rightarrow N+1 = number\ of\ data\ samples$$

Discrete Fourier Transform (2)

"Record Length" of sampled signal is equivalent to the *Fundamental period*

$$y_k = \left\{ y_0, y_{1\cdot\Delta t}, y_{2\cdot\Delta t} + \ldots y_{N\cdot\Delta t} \right\} \quad \rightarrow N+1 = \textit{number of data samples}$$

$$T \equiv N \cdot \Delta t \rightarrow \omega_0 = \frac{2\pi}{T} = \frac{2\pi}{N \cdot \Delta t} \rightarrow t_k = k \cdot \Delta t$$

Approximate Complex Coefficient Integral by a Discrete Sum

$$Y_n = \left[\frac{1}{T}\right] \int_0^T y(\tau) \cdot e^{-j \cdot n \cdot \frac{2\pi}{T} \cdot \tau} \cdot d\tau \rightarrow$$

$$Y_n \approx \left[\frac{1}{N \cdot \Delta t}\right] \sum_{k=0}^{N} y_k \cdot e^{-j \cdot n \cdot \frac{2\pi}{N \cdot \Delta t}(k \cdot \Delta t)} \cdot \Delta t = \frac{1}{N} \cdot \sum_{k=0}^{N} y_k \cdot e^{-j \cdot \frac{2\pi}{N} \cdot n \cdot k}$$

Discrete Fourier Transform (3)

$$Y_n = \frac{1}{N}\sum_{k=0}^{N} y_k e^{-j\left[\frac{2\pi}{N}\cdot n\cdot k\right]} \rightarrow n = \{0,1,2,..\}$$

Time index

Frequency index

Computer Implementation of Fourier Series Analysis

Corresponding to … frequency …

$$f_n = n\cdot\Delta f = n\cdot\frac{1}{T} = \frac{n}{N\cdot\Delta T}$$

… Y_n are the discrete frequency components of the spectrum

$$f_n = \frac{n}{N\cdot\Delta T} \rightarrow N\cdot\Delta t = T \approx record\ length$$

$$t_k = k\cdot\Delta t \rightarrow sample\ time$$

Summation is Repeated for Each fundamental frequency harmonic

Discrete Fourier Transform (4)

$$Y_n = \frac{1}{N}\sum_{k=0}^{N} y_k e^{-j\left[\frac{2\pi}{N}\cdot n\cdot k\right]} \rightarrow n = \{0,1,2,..N/2\}$$

$$\boxed{\begin{array}{c} f_n = \dfrac{n}{N\cdot\Delta T} \rightarrow N\cdot\Delta t = T \approx record\ length \\ t_k = k\cdot\Delta t \rightarrow sample \end{array}}$$

$$\boxed{f_{\frac{N}{2}} = \frac{N/2}{N\cdot\Delta T} = \frac{1}{2\cdot\Delta T} = f_{NYQ}}$$

N/2 -→ harmonic correspondng To Nyquist frequency

How to reconstruct signal from discrete harmonics

$$y(k\cdot t) = \sum_{n=0}^{N/2}\left(Y_n\cdot \mathrm{e}^{-j\cdot\left(\frac{2\pi}{N}\cdot n\cdot k\right)}\right) + \sum_{n=\frac{N}{2}+1}^{N-1}\left(Y^{*}{}_{N-n}\cdot \mathrm{e}^{-j\cdot\left(\frac{2\pi}{N}\cdot n\cdot k\right)}\right)$$

Discrete Fourier Transform (5)

- Compare DTF to Fourier series …

Discrete coefficients

$$Y_n = \frac{1}{N}\sum_{k=0}^{N} y_k e^{-j\left[\frac{2\pi}{N}\cdot n\cdot k\right]}$$

$$\boxed{n = 0, 1, \ldots, N/2}$$

$$\boxed{T = N\cdot\Delta t}$$

$$\boxed{f_{\frac{N}{2}} = \frac{N/2}{N\cdot\Delta T} = \frac{1}{2\cdot\Delta T} = f_{NYQ}}$$

A, B coefficients

$$\begin{bmatrix} A_n = \left[\frac{2}{T}\right]\int_0^T y(\tau)\cos\left(\frac{2\pi}{T}n\tau\right)d\tau \\ B_n = \left[\frac{2}{T}\right]\int_0^T y(\tau)\sin\left(\frac{2\pi}{T}n\tau\right)d\tau \end{bmatrix}$$

Complex coefficients

$$\boxed{Y_n = \left[\frac{1}{T}\right]\int_0^T y(\tau)e^{-j\left[\frac{2\pi}{T}n\tau\right]}d\tau}$$

$$\begin{bmatrix} A_n = 2Re(Y_n) \\ B_n = -2Im(Y_n) \end{bmatrix}$$

- T --> period

$\omega_0 = 2\pi f_0 = 2\pi/T$

Discrete Fourier Transform (6)

- The D.F.T. can also be decomposed into real *sine* and *cosine* series *(this is how your text book – Beckwith -- does it)*

$$A_n = \frac{2}{N}\sum_{k=1}^{N} y(k\Delta t)\cos\left(\frac{2\pi\cdot k\cdot n}{N}\right) \qquad n = 0, 1,. . . , N/2$$

$$B_n = \frac{2}{N}\sum_{k=1}^{N} y(k\Delta t)\sin\left(\frac{2\pi\cdot k\cdot n}{N}\right) \qquad n = 1, 2, . . ., N/2 -1$$

(N is even)

- *Inverse transform is*

$$y_k = \frac{A_0}{2} + \sum_{n=1}^{N/2-1}\left[A_n\cos\left(\frac{2\pi}{T}n\cdot k\cdot\Delta t\right) + B_n\sin\left(\frac{2\pi}{T}n\cdot k\cdot\Delta t\right)\right] + \frac{A_{N/2}}{2}\cos\left(\frac{2\pi}{T}N\cdot k\cdot t/2\right)$$

- ***Almost never used in practice …Because of Fast Fourier Transform … Is far easier to implement in complex number form***

Discrete Fourier Transform (7)

- DFT Notes
- Y_n components are generally *complex-valued* with real and imaginary components
- First sample Y_0 of the transformed series is the DC component, and is *real-valued*
- The DFT of a real series, ie: a physical series, results in a symmetric frequency spectrum series about the Nyquist frequency.
- Elements from $n=0 \ldots n = N/2$ .. Correspond to "real frequencies"
- $f_{N/2}$ is the nyquist frequency
- Elements from $n=N/2+1 \ldots n = N$.. Are mathematical artifacts

Correspond to non-physical "negative frequencies" … necessary for inverse transform to give a real-values result

Discrete Fourier Transform (8)

• DFT Notes

• The spectrum points corresponding to negative frequency are the complex conjugate of the positive frequency spectrum values.

$$Y_n = \left(Y_{N-n}\right)^* \rightarrow n \le \frac{N}{2}$$

• The highest positive (or negative) frequency sample is called the Nyquist frequency. Highest frequency component that should exist in the input series for the DFT to yield "uncorrupted" results.

• More specifically if there are no frequencies above Nyquist the original signal can be **exactly** reconstructed from the samples.

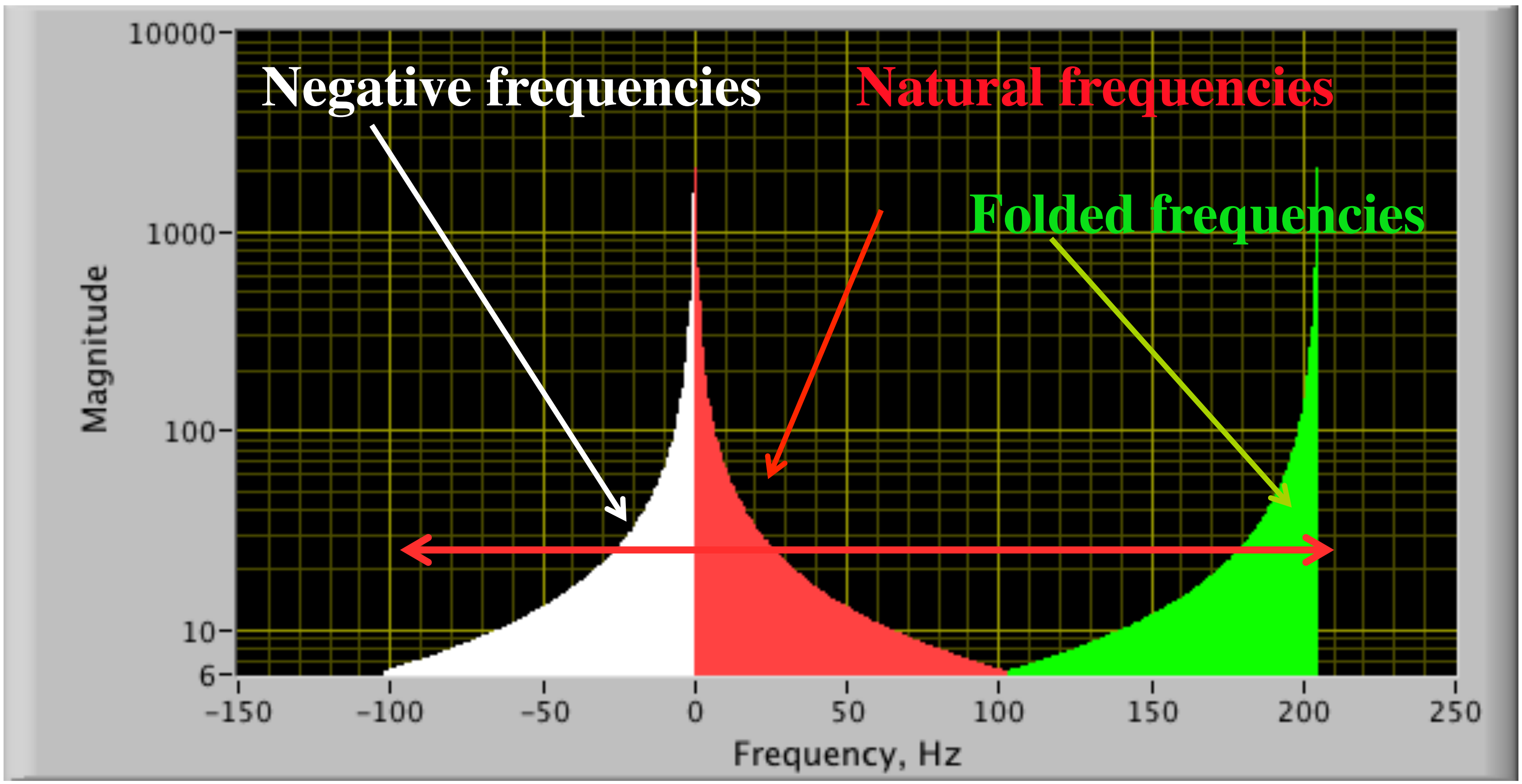
• ***The minimum frequency that can be resolved in a DFT spectrum is equal to One over the sample record length***

UtahState UNIVERSITY

Mechanical & Aerospace Engineering

Discrete Fourier Transform (9)

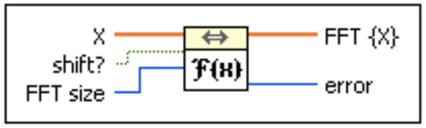
$$y(k \cdot t) = \sum_{n=0}^{N/2} \left(Y_n \cdot e^{-j \cdot \left(\frac{2\pi}{N} \cdot n \cdot k \right)} \right) + \sum_{n=\frac{N}{2}+1}^{N-1} \left(Y^*_{N-n} \cdot e^{-j \cdot \left(\frac{2\pi}{N} \cdot n \cdot k \right)} \right)$$



- Negative Frequencies get … folded to upper half of spectrum

Discrete Fourier Transform Labview VI

Real FFT



- *X is a real valued input vector function*
- *FFT(x) is a complex valued output vector*

DBL **X** is a real vector.

TF **shift?** specifies whether the DC component is at the center of **FFT {X}**. The default is FALSE.

I32 **FFT size** is the length of the FFT you want to perform. If **FFT size** is greater than the number of elements in **X**, this VI adds zeros to the end of **X** to match the size of **FFT size**. If **FFT size** is less than the number of elements in **X**, this VI uses only the first *n* elements in **X** to perform the FFT, where *n* is **FFT size**. If **FFT size** is less than or equal to 0, this VI uses the length of **X** as the **FFT size**.

CDB **FFT {X}** is the FFT of **X**.

I32 **error** returns any error or warning from the VI. You can wire **error** to the Error Cluster From Error Code VI to convert the error code or warning into an error cluster.

For 1D signals, the FFT VI computes the discrete Fourier transform (DFT) of the input sequence with a fast Fourier transform algorithm. The 1D DFT is defined

$$y_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N} \qquad \text{for } n = 0, 1, 2, \ldots, N-1$$

Labview VI does not divide by N

Proper Definition:

$$Y_n = \frac{1}{N}\sum_{k=0}^{N} y_k e^{-j\left[\frac{2\pi}{N}\cdot n\cdot k\right]}$$

where x is the input sequence, N is the number of elements of x, and Y is the transform result.

The frequency resolution, or the frequency spacing between the components of Y, is:

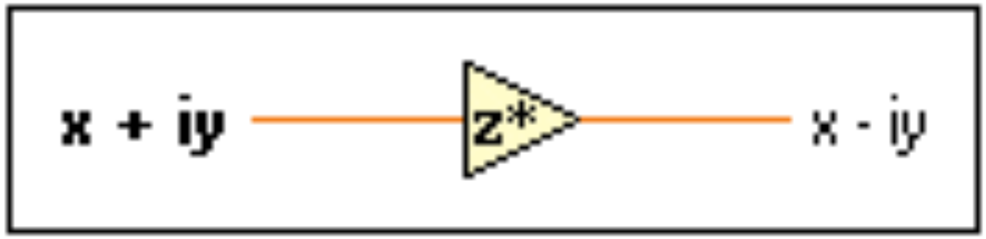
$$\Delta f = \frac{f_s}{N}$$

where f_s is the sampling frequency.

$$Y_{n_{labview}} = \frac{N}{2}\cdot\left(A_n - j\cdot B_n\right)$$

Discrete Fourier Transform Labview VI (2)

- **"Upper" half of Spectrum is complex Conjugate of "Lower" half … With reversed order!**

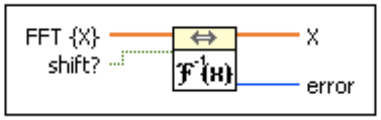


$$Y_{(f_i)} \rightarrow .....f_i = i \cdot \Delta f = \frac{i}{T} = \frac{i}{N\Delta t}, \left\{ i = \frac{N}{2}+1, \frac{N}{2}+2, \; .... N \right\} =$$

$$Y^*_{(f_i)} \rightarrow .....f_i = \frac{i}{N\Delta t}, \left\{ i = \frac{N}{2}-1, \; \frac{N}{2}-2, \; ..., \; 2, \; 1, \; 0 \right\}$$

Inverse Discrete Fourier Transform Labview VI

Inverse Real FFT



- *FFT(x) is a complex valued input vector*
- *X is a real valued output vector*

FFT {X} is the complex valued input sequence, which should be conjugated centrosymmetric except for the first element. This instance uses only the anterior half of **FFT {X}**.

shift? specifies whether the DC component is at the center of **FFT {X}**. The default is FALSE.

X is the inverse real FFT of **FFT{X}**.

error returns any error or warning from the VI. You can wire **error** to the Error Cluster From Error Code VI to convert the error code or warning into an error cluster.

For a 1D, *N*-sample, frequency domain sequence *Y*, the IDFT is defined as:

$$X_n = \frac{1}{N}\sum_{k=0}^{N-1} Y_k e^{j2\pi kn/N}$$

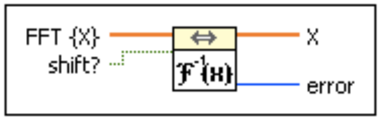
for $n = 0, 1, 2, \ldots, N-1$.

$$Y_k = \frac{N}{2}\cdot\left(A_k - j\cdot B_k\right)$$

$$x_n = x(n\cdot\Delta t)$$

Inverse Discrete Fourier Transform Labview VI (2)

Inverse Real FFT



- *FFT(x) is a complex valued input vector*
- *X is a real valued output vector*

FFT {X} is the complex valued input sequence, which should be conjugated centrosymmetric except for the first element. This instance uses only the anterior half of **FFT {X}**.

shift? specifies whether the DC component is at the center of **FFT {X}**. The default is FALSE.

X is the inverse real FFT of **FFT{X}**.

error returns any error or warning from the VI. You can wire **error** to the Error Cluster From Error Code VI to convert the error code or warning into an error cluster.

When **FFT {X}** is the Fourier transform of a 1D real time-domain signal with length N, the posterior half part of **FFT {X}** can be constructed by the anterior half part. The centrosymmetric relationship between the anterior and posterior half part of **FFT {X}** can be written as

$$f_{N-i} = f_i^*,\ i = 1, 2, \ldots, \left\lfloor \frac{N}{2} \right\rfloor,$$

where f_i is the element in **FFT {X}**.

The Inverse Real FFT instance VI uses only the anterior half part, from f_0 to $f_{\left\lfloor \frac{N}{2} \right\rfloor}$ to perform the inverse real FFT, where $\lfloor \bullet \rfloor$ means the floor operation.

Fast Fourier Transform (6)

• The Discrete *Fourier Transform* (DFT) is used to produce frequency analysis of discrete non-periodic signals.

• The FFT is another method of achieving the same result, BUT it is incredibly more efficient, often reducing the computation time by *hundreds* .. same improvement as flying in a jet aircraft versus walking!

• If the FFT were not available, many of the techniques described in this class would not be practical.

An FFT computation takes approximately N*log2(N) operations, whereas a DFT computation takes approximately N*N operations, so the FFT is significantly faster.

Speed ratio = N/log(N) …> 4096 pts --> 492 times faster

Fast Fourier Transform (2)

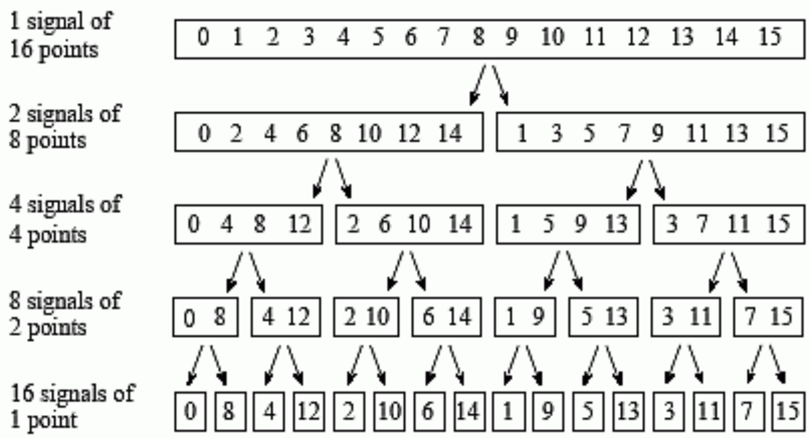
• While the FFT only requires a few dozen lines of code, it is one of the most complicated algorithms in DIGITAL SIGNAL PROCESSING. (DSP)

• FFT operates by decomposing an *N* point time domain signal into *N* time domain signals each composed of a single point.

• SECOND step is to calculate the *N* frequency spectra corresponding to these *N* time domain signals.

• Finally, the *N* spectra are synthesized into a single frequency spectrum.

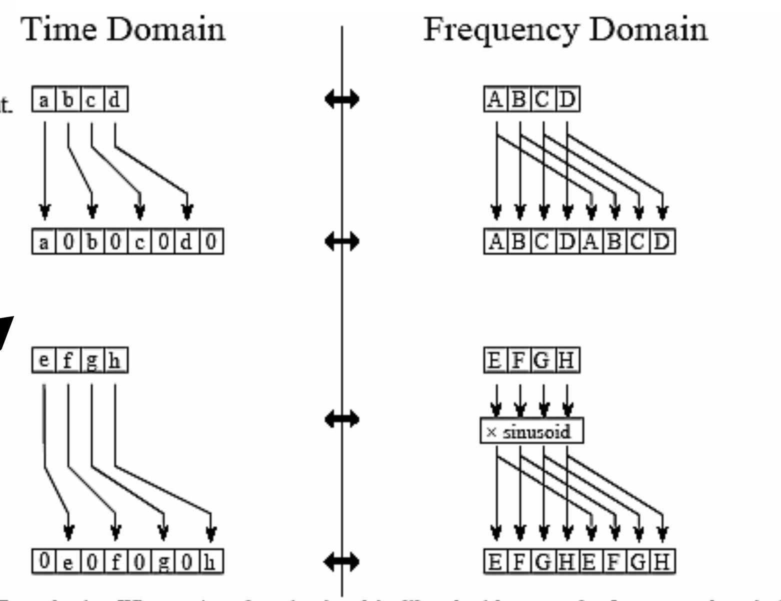
Fast Fourier Transform (3)



The FFT decomposition. An *N* point signal is decomposed into *N* signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.

Sample numbers in normal order: *Decimal*	*Binary*		Sample numbers after bit reversal: *Decimal*	*Binary*
0	0000		0	0000
1	0001		8	1000
2	0010		4	0100
3	0011		12	1100
4	0100		2	0010
5	0101		10	1010
6	0110	⇨	6	0100
7	0111		14	1110
8	1000		1	0001
9	1001		9	1001
10	1010		5	0101
11	1011		13	1101
12	1100		3	0011
13	1101		11	1011
14	1110		7	0111
15	1111		15	1111

The FFT bit reversal sorting. The FFT time domain decomposition can be implemented by sorting the samples according to bit reversed order.



The FFT synthesis. When a time domain signal is diluted with zeros, the frequency domain is duplicated. If the time domain signal is also shifted by one sample during the dilution, the spectrum will additionally be multiplied by a sinusoid.

- ref: http://www.dspguide.com/ch12/2.htm

Fast Fourier Transform (4)

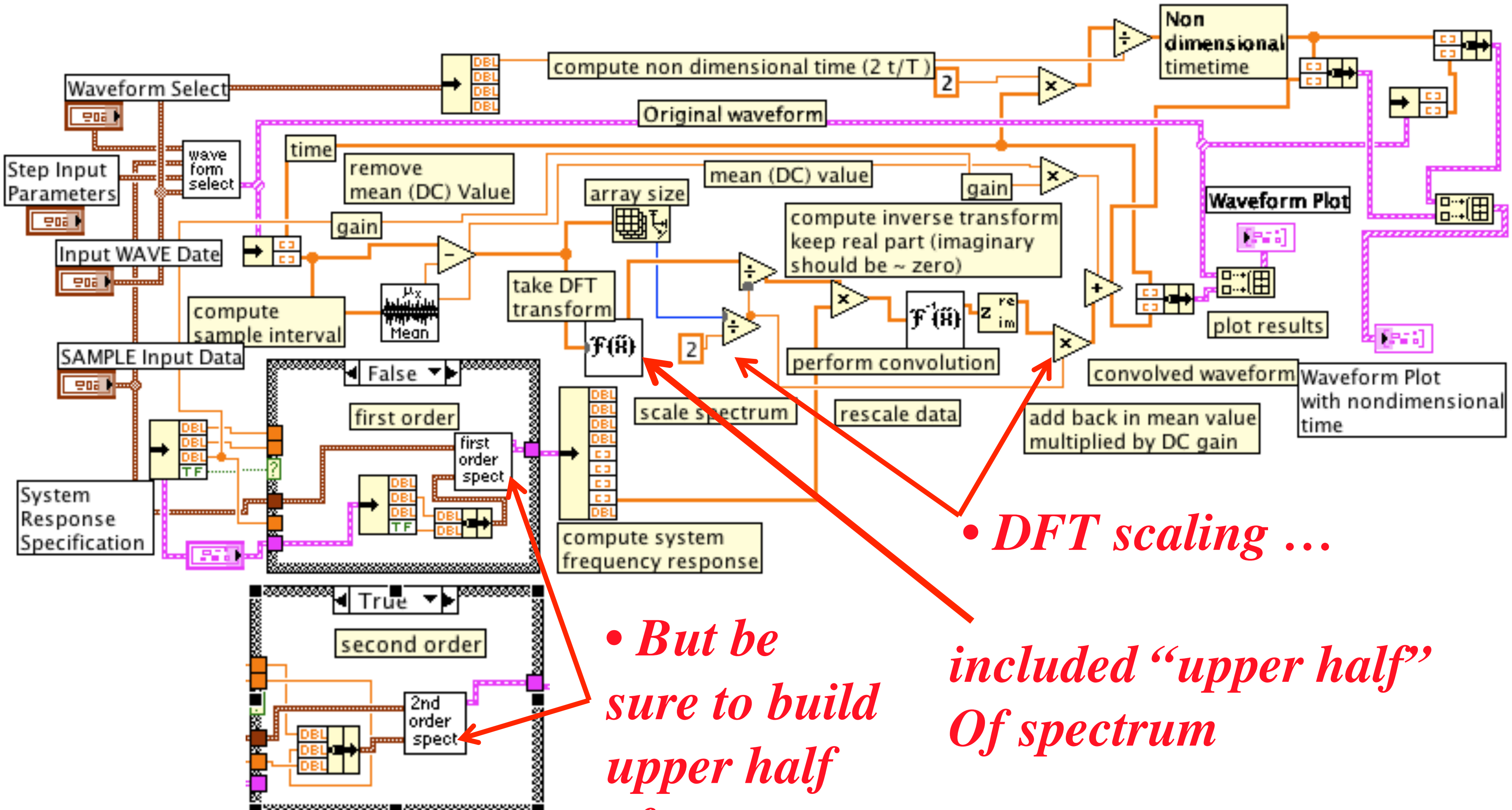
• Most common FFT algorithm required that sample length Be a power number of two data points

$N=2^{i}$, ……. i=2,3,4, ….

• You will NOT be required to develop an FFT algorithm in the class … Labview has a whole suite of FFT-Based signal processing ***codes***

Signal Processing Using DFT

- **We can "mechanize" the response of a system to an arbitrary Input using the Forward and Inverse Discrete Fourier transform**



• DFT scaling …

• But be sure to build upper half of spectrum

included "upper half" Of spectrum

DFT Applications … Noise Canceling headsets

- Essentially, this involves using a microphone, placed near the ear, and electronic circuitry which generates an "antinoise" sound wave with the opposite polarity of the sound wave arriving at the microphone. This results in destructive interference, which cancels out the noise within the enclosed volume of the headphone

UtahState University

Mechanical & Aerospace Engineering

"Simple" Noise Canceling Logic Written in Labview