6.3000: Signal Processing

FFT

April 03, 2025

Inverse FFT

Here is the lecture code for the FFT algorithm.

```
from math import e,pi
def FFT(x):
    N = len(x)
    if N != 2*N//2:
        print('N must be a power of 2')
        exit(1)
    if N==1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k]+e**(-2j*pi*k/N)*Xo[k])/2)
    for k in range(N//2):
        X.append((Xe[k]-e**(-2j*pi*k/N)*Xo[k])/2)
    return X
```

How would you change the code to compute the inverse FFT?

Note: If `FFT(f)` returns `F`, then `iFFT(F)` should return `f`.

Inverse FFT

The FFT computes what we have been calling the DFT **analysis equation**.
The iFFT should compute the DFT **synthesis equation**.

$$F[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] e^{-j2\pi kn/N} \qquad \textbf{analysis equation}$$

$$f[n] = \sum_{k=0}^{N-1} F[k] e^{j2\pi kn/N} \qquad \textbf{synthesis equation}$$

The $\frac{1}{N}$ scale factor in the analysis equation is not in the synthesis equation.

The complex exponentials in the two equations are complex conjugates of each other.

Inverse FFT

What do you need to conjugate?

```
from math import e,pi
def FFT(x):
    N = len(x)
    if N != 2*N//2:
        print('N must be a power of 2')
        exit(1)
    if N==1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k]+e**(-2j*pi*k/N)*Xo[k])/2)
    for k in range(N//2):
        X.append((Xe[k]-e**(-2j*pi*k/N)*Xo[k])/2)
    return X
```

Inverse FFT

What do you need to conjugate?

```
from math import e,pi
def FFT(x):
    N = len(x)
    if N != 2*N//2:
        print('N must be a power of 2')
        exit(1)
    if N==1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k]+e**(+2j*pi*k/N)*Xo[k])/2
    for k in range(N//2):
        X.append((Xe[k]-e**(+2j*pi*k/N)*Xo[k])/2
    return X
```

The only complex numbers in the algorithm are in the complex exponential.

Inverse FFT

Where is the $\frac{1}{N}$ factor?

```
from math import e,pi
def FFT(x):
    N = len(x)
    if N != 2*N//2:
        print('N must be a power of 2')
        exit(1)
    if N==1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k]+e**(-2j*pi*k/N)*Xo[k])/2)
    for k in range(N//2):
        X.append((Xe[k]-e**(-2j*pi*k/N)*Xo[k])/2)
    return X
```

Inverse FFT

Where is the $\frac{1}{N}$ factor?

```
from math import e,pi
def FFT(x):
    N = len(x)
    if N != 2*N//2:
        print('N must be a power of 2')
        exit(1)
    if N==1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k]+e**(-2j*pi*k/N)*Xo[k])/2)
    for k in range(N//2):
        X.append((Xe[k]-e**(-2j*pi*k/N)*Xo[k])/2)
    return X
```

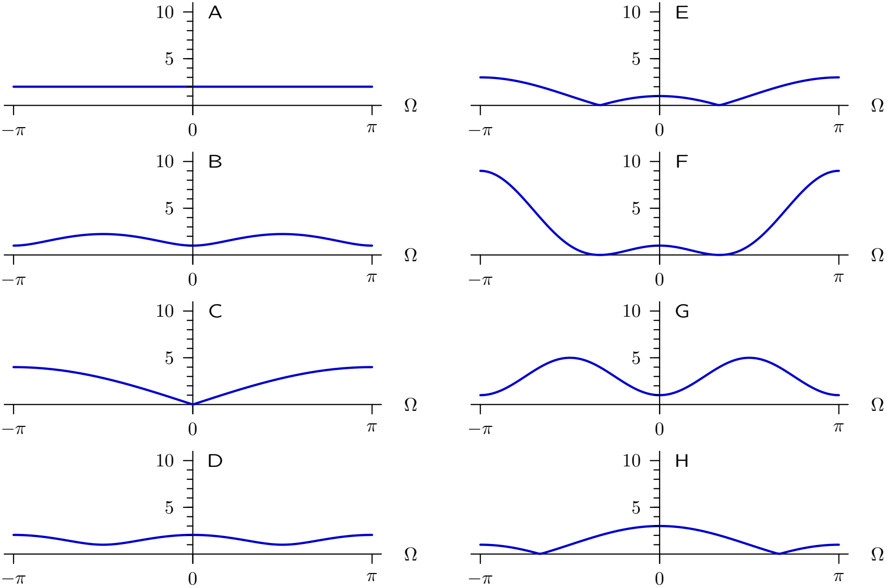
Q: How does a 2 result in N?

A: There is a 2 in each of the $\log_2(N)$ decimations.

Multiply 2 times itself $\log_2(N)$ times: $2^{\log_2(N)} = N$

Quiz Review: Composite Systems

The following plots should the magnitudes of the frequency responses of eight discrete-time, linear, time-invariant systems.



Composite Systems

Which plot (if any) shows the magnitude of the frequency response of a system with each of the following unit-sample responses?

$$h_1[n] = g_1[n] = \delta[n] - \delta[n-1] + \delta[n-2]$$

$$h_2[n] = g_2[n] = \delta[n] + \delta[n-1] - \delta[n-2]$$

$$h_3[n] = g_1[n] + g_2[n]$$

$$h_4[n] = g_1[n] - g_2[n]$$

$$h_5[n] = g_1[n] \times g_1[n]$$

$$h_6[n] = g_1[n] \times g_2[n]$$

$$h_7[n] = (g1 * g1)[n]$$

$$h_8[n] = (g2 * g2)[n]$$

Composite Systems

Start by determining expressions for $h_1[n]$ through $h_8[n]$.

Then determine the magnitudes of the frequency responses at $\Omega=0$ and π.

	sample at time n					$\lvert H(0)\rvert$	$\lvert H(\pi)\rvert$	match
	0	1	2	3	4			
$h_1[n]$	1	-1	1	0	0	1	3	E
$h_2[n]$	1	1	-1	0	0	1	1	B, G
$h_3[n]$	2	0	0	0	0	2	2	A
$h_4[n]$	0	-2	2	0	0	0	4	C
$h_5[n]$	1	1	1	0	0	3	1	H
$h_6[n]$	1	-1	-1	0	0	1	1	B, G
$h_7[n]$	1	-2	3	-2	1	1	9	F
$h_8[n]$	1	2	-1	-2	1	1	1	B, G

To distinguish B and G, evaluate $H(\pi/2)$.

$H_2(\pi/2) = 2 - j$ so $|H_2(\pi/2)| = \sqrt{5}$.

$H_6(\pi/2) = 2 + j$ so $|H_6(\pi/2)| = \sqrt{5}$.

$H_8(\pi/2) = 3 - 4j$ so $|H_8(\pi/2)| = 5$.

It follows that $H_2(\Omega)$ and $H_6(\Omega)$ correspond to B but $H_8(\Omega)$ to G.