EE538 DSP I

Module 20

Outline:

- Discrete Fourier Transform (DFT)
  - Sect. 7.1
- Fast Fourier Transform (FFT)
  - Sect. 8.1.3

- recall DTFT:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

- Two problems relative to practical computation:

1. infinite limits
2. ω is continuous

- Thus, we require
  1. Truncation to finite limits
  2. discretize ω

- Consider truncating $x[n]$ to L nonzero points and shifting so that it starts at $n=0$
- Note:

$$x[n-n_0] \xleftrightarrow{\text{DTFT}} e^{-j\omega n_0} X(\omega)$$

- evaluate $X(\omega)$ at N equi-spaced frequencies $0 < \omega < 2\pi$

$$X_N(k) = X(\omega)\Big|_{\omega_k = \frac{2\pi k}{N}} \qquad k = 0, 1, \ldots, N-1$$

- yields N pt. DFT:

$$X_N(k) = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi k}{N} n}$$

$$k = 0, 1, \ldots, N-1$$

- if $N > L$, notation assumes $N-L$ zeroes are appended (zero padding)
- Why $0 < \omega < 2\pi$ rather than $-\pi < \omega < \pi$?

- $X_N(k)$ stored in an array on computer, DSP chip, μ-proc, etc
  $\Rightarrow$ memory locations are indexed by positive integers
- correspondence between each value of k and a DT frequency ω_k:
- $k \in [0, \lfloor \frac{N}{2} \rfloor)$ $\Rightarrow$ "positive" frequencies
  $\Rightarrow$ corresponding $\omega_k = 2\pi \frac{k}{N}$

- $k \in \left(\lfloor \frac{N}{2} \rfloor + 1, N-1\right)$:

corresponding DT frequency:

$$\omega_k = -2\pi \frac{(N-k)}{N}$$

- Fast Computation of the DFT
- Benchmark: Direct Computation

$$X_N(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N} n}$$

$$k = 0, 1, \ldots, N-1$$

- for each k; require:
  N mults. and $N-1$ adds
- for all N values of k:
  N^2 mults. and $N(N-1)$ adds
- even if $x[n]$ is real-valued, mults. and adds are complex-valued

- Decimation-in-Time Radix 2 FFT $\Rightarrow$ premised on choosing $N = 2^{\nu}$, where ν is an integer
- will show that the number of mults. can be reduced from N^2 to $\frac{N}{2} \log_2(N)$ $\left(= \frac{N\nu}{2}\right)$

$$X_N(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N} n}$$

$$= \sum_{n'=0}^{\frac{N}{2}-1} x[2n'] e^{-j 2\pi \frac{k}{N}(2n')}$$

$$+ \sum_{n'=0}^{\frac{N}{2}-1} x[2n'+1] e^{-j 2\pi \frac{k}{N}(2n'+1)}$$

$$X_N(k) = \sum_{n=0}^{\frac{N}{2}-1} f_1[n]\, e^{-j\frac{2\pi k}{N/2} n}$$

$$+\, e^{-j\frac{2\pi k}{N}} \sum_{n=0}^{\frac{N}{2}-1} f_2[n]\, e^{-j\frac{2\pi k}{N/2} n}$$

- where: $f_1[n] = x[2n]$

$f_2[n] = x[2n+1]$, $n = 0, 1, \ldots, \frac{N}{2}-1$

• consider:

$$X_N\left(k+\frac{N}{2}\right)=\sum_{n=0}^{N/2-1} f_1[n]\, e^{-j\frac{2\pi\left(k+\frac{N}{2}\right)}{N/2}n}$$

$$+\underbrace{e^{-j\frac{2\pi}{N}\left(k+\frac{N}{2}\right)}}_{e^{-j\frac{2\pi k}{N}}\cdot \underbrace{e^{-j\pi}}_{-1}}\sum_{n=0}^{N/2-1} f_2[n]\, e^{\underbrace{-j\frac{2\pi\left(k+\frac{N}{2}\right)n}{N/2}}_{e^{-j\frac{2\pi k n}{N/2}}\cdot \underbrace{e^{-j2\pi n}}_{1}}}$$

$$X_N(k) = F_1(k) + W_N^k F_2(k)$$

$$k = 0, 1, \ldots, \frac{N}{2} - 1$$

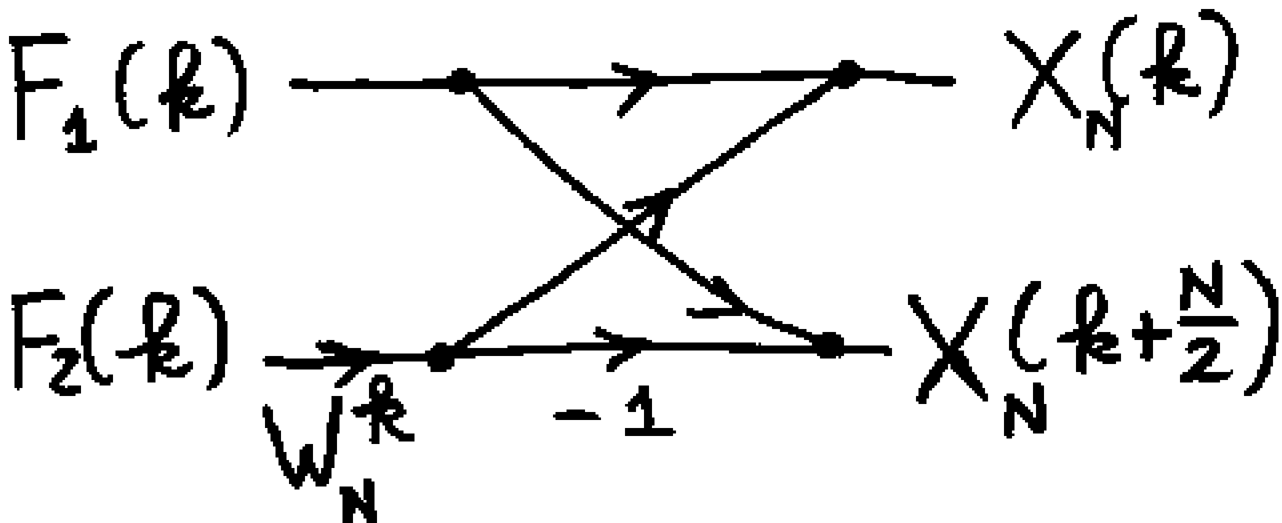
$$X_N\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k)$$

$$k = 0, 1, \ldots, \frac{N}{2} - 1$$

- where: $W_N = e^{-j\frac{2\pi}{N}}$

$f_1[n] = x[2n] \xleftrightarrow[\frac{N}{2}]{DFT} F_1(k)$ | $f_2[n] = x[2n+1] \xleftrightarrow[\frac{N}{2}]{DFT} F_2(k)$

- represent via butterfly as



- each butterfly yields two values of $X_N(k)$
- need $\frac{N}{2}$ butterflies total

- each butterfly requires 1 mult and 2 adds

$$f_1[n] \xleftrightarrow[N/2]{DFT} F_1(k) \quad \Big| \quad f_2[n] \xleftrightarrow[N/2]{DFT} F_2(k)$$

requires $\left(\frac{N}{2}\right)^2$ mults | requires: $\left(\frac{N}{2}\right)^2$ mults

$$+\left\{ \frac{N}{2} \text{ butterflies} \times 1 \frac{\text{mult.}}{\text{butterfly}} \right\}$$

$$\text{Total}: 2\left(\frac{N}{2}\right)^2 + \frac{N}{2} = \frac{N}{2}(N+1)$$

- apply decimation-in-time to

$f_1[n] = x(2n)$ $\quad \left(\frac{N}{4} \text{ is an integer}\right)$

to get $F_1(k)$ via two $\frac{N}{4}$ pt. DFT's :

$$2\left(\frac{N}{4}\right)^2 + \frac{N}{4} \text{ mults.}$$

- also the same for $F_2(k)$
- total after 2 stages: $2\left\{2\left(\frac{N}{4}\right)^2 + \frac{N}{4}\right\} + \frac{N}{2} = N\left(\frac{N}{4} + 1\right)$

- repeat process until ultimately have sequences of length 2
- note : 2 pt. DFT can be represented via a butterfly with $W_N^0 = 1$
- it takes $v = \log_2(N)$ stages to achieve this condition
- Example : $N = 8$
  See Figs. 8.1.4, 8.1.5, and 8.1.6

- Total Computation:
- $v = \log_2(N)$ stages
- each stage involves $\frac{N}{2}$ butterflies
- total no. of mults.:

$$\frac{1 \text{ mult}}{\text{butterfly}} \times \frac{N}{2} \frac{\text{butterflies}}{\text{stage}} \times \log_2 N \text{ stages}$$

$$= \frac{N}{2} \log_2(N) \text{ mults. vs. } N^2$$

- total no. of adds:

$$\underbrace{N \log_2(N)}_{\frac{2 \text{ adds}}{\text{butterfly}}} \quad \text{vs.} \quad \underbrace{N(N-1)}_{\text{Direct Computation}}$$

- See Table 8.1 on pg. 522 for Comparison Computation