ELEN E4810: Digital Signal Processing

Topic 10:
The Fast Fourier Transform

1. Calculation of the DFT
2. The Fast Fourier Transform algorithm
3. Short-Time Fourier Transform
1. Calculation of the DFT

- Filter design so far has been oriented to time-domain processing - cheaper!
- But: frequency-domain processing makes some problems very simple:
  - use all of $x[n]$, or use short-time windows
  - Need an **efficient** way to calculate DFT
The DFT

- Recall the DFT:

\[
X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}
\]

- discrete transform of discrete sequence

- Matrix form:

\[
\begin{bmatrix}
X[0] \\
X[1] \\
X[2] \\
\vdots \\
X[N-1]
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\
1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2}
\end{bmatrix} \begin{bmatrix}
x[0] \\
x[1] \\
x[2] \\
\vdots \\
x[N-1]
\end{bmatrix}
\]

\[
W_N = e^{-j\frac{2\pi}{N}}
\]

\(W_N\) @ \(2\pi/N\)

\(\Rightarrow W_N^r\) has only \(N\) distinct values

Structure \(\Rightarrow\) opportunities for efficiency
Computational Complexity

\[ X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \]

- \( N \) complex multiplies
  + \( N-1 \) complex adds per point \((k)\)
  \times \( N \) points \( (k = 0..N-1) \)
  - cpx mult: \((a+jb)(c+jd) = ac - bd + j(ad + bc)\)
  = 4 real mults + 2 real adds
  - cpx add = 2 real adds
- \( N \) points: 4\( N^2 \) real mults, 4\( N^2-2N \) real adds
Goertzel’s Algorithm

Now: $X[k] = \sum_{\ell=0}^{N-1} x[\ell]W_N^{k\ell}$

$= W_N^{kN} \sum_{\ell} x[\ell]W_N^{-k(N-\ell)}$

i.e. $X[k] = y_k[N]$ where $y_k[n] = x_e[n] \ast h_k[n]$

$x_e[n] = \begin{cases} x[n] & 0 \leq n < N \\ 0 & n = N \end{cases}$

$h_k[n] = \begin{cases} W_N^{-kn} & n \geq 0 \\ 0 & n < 0 \end{cases}$
Goertzel’s Algorithm

- Separate ‘filters’ for each $X[k]$
  - can calculate for just a few values of $k$
- No large buffer, no coefficient table
- Same complexity for full $X[k]$
  (4$N^2$ mults, 4$N^2 - 2N$ adds)
  - but: can halve multiplies by making the denominator real:

\[
H(z) = \frac{1}{1 - W_N^{-k}z^{-1}} = \frac{1}{1 - 2 \cos \frac{2\pi k}{N} z^{-1} + z^{-2}}
\]

- evaluate only for last step
- 2 real mults per step
2. Fast Fourier Transform FFT

- Reduce complexity of DFT from $O(N^2)$ to $O(N \cdot \log N)$
  - grows more slowly with larger $N$
- Works by decomposing large DFT into several stages of smaller DFTs
- Often provided as a highly optimized library
Decimation in Time (DIT) FFT

- Can rearrange DFT formula in 2 halves:

\[
X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}
\]

\[k = 0..N-1\]

\[
= \sum_{m=0}^{N/2-1} \left( x[2m] \cdot W_N^{2mk} + x[2m + 1] \cdot W_N^{(2m+1)k} \right)
\]

Arrange terms in pairs...

Group terms from each pair

[N/2 pt DFT of x for even n] [N/2 pt DFT of x for odd n]
Decimation in Time (DIT) FFT

We can evaluate an $N$-pt DFT as two $N/2$-pt DFTs (plus a few mults/adds)

But if $\text{DFT}_N \{ \cdot \} \sim O(N^2)$

then $\text{DFT}_{N/2} \{ \cdot \} \sim O((N/2)^2) = 1/4 \ O(N^2)$

$\Rightarrow$ Total computation $\sim 2 \cdot 1/4 \ O(N^2)$

$= 1/2$ the computation (+ε) of direct DFT
One-Stage DIT Flowgraph

\[ X[k] = X_0 \left[ \langle k \rangle \frac{N}{2} \right] + W_N^k X_1 \left[ \langle k \rangle \frac{N}{2} \right] \]

Even points from \( x[n] \)

Odd points from \( x[n] \)

Same as \( X[0..3] \) except for factors on \( X_1[\bullet] \) terms

Twiddle factors: always apply to odd-terms output

Not mirror-image

Classic FFT structure
Multiple DIT Stages

- If decomposing one DFT$_N$ into two smaller DFT$_{N/2}$’s speeds things up ... Why not further divide into DFT$_{N/4}$’s?

- i.e. $X[k] = X_0\left[\frac{k}{N/2}\right] + W_N^k X_1\left[\frac{k}{N/2}\right]$

- make: $X_0[k] = X_{00}\left[\frac{k}{N/4}\right] + W_N^{k/2} X_{01}\left[\frac{k}{N/4}\right]$

- $N/4$-pt DFT of even points

- in even subset of $x[n]$

- Similarly, $X_1[k] = X_{10}\left[\frac{k}{N/4}\right] + W_N^{k/2} X_{11}\left[\frac{k}{N/4}\right]$

- $N/4$-pt DFT of odd points

- from even subset
Two-Stage DIT Flowgraph

\[ x[0] \rightarrow \text{DFT}_{N/4} \rightarrow X_0[0] \]

\[ x[4] \rightarrow \text{DFT}_{N/4} \rightarrow X_0[1] \]

\[ x[2] \rightarrow \text{DFT}_{N/4} \rightarrow X_0[2] \]

\[ x[6] \rightarrow \text{DFT}_{N/4} \rightarrow X_0[3] \]

\[ x[1] \rightarrow \text{DFT}_{N/4} \rightarrow X_1[0] \]

\[ x[5] \rightarrow \text{DFT}_{N/4} \rightarrow X_1[1] \]

\[ x[3] \rightarrow \text{DFT}_{N/4} \rightarrow X_1[2] \]

\[ x[7] \rightarrow \text{DFT}_{N/4} \rightarrow X_1[3] \]

\[ W_{N/2}^0 \rightarrow X_0[0] \]

\[ W_{N/2}^0 \rightarrow X_1[0] \]

\[ W_N^1 \rightarrow X[1] \]

\[ W_N^2 \rightarrow X[2] \]

\[ W_N^3 \rightarrow X[3] \]

\[ W_N^4 \rightarrow X[4] \]

\[ W_N^5 \rightarrow X[5] \]

\[ W_N^6 \rightarrow X[6] \]

\[ W_N^7 \rightarrow X[7] \]
Multi-stage DIT FFT

- Can keep doing this until we get down to 2-pt DFTs:

\[ X[0] = x[0] + x[1] \]
\[ X[1] = x[0] - x[1] \]

\[ N = 2^M \text{-pt DFT reduces to } M \text{ stages of twiddle factors & summation} \]

\[ O(N^2) \text{ part vanishes} \]

\[ \rightarrow \text{real mults } < M \cdot 4N, \text{ real adds } < 2 \cdot M \cdot 2N \]

\[ \rightarrow \text{complexity } \sim O(N \cdot M) = O(N \cdot \log_2 N) \]
FFT Implementation Details

- Basic butterfly (at any stage):
  \[
  W_N^r X_{X0}[r] + \frac{N}{2} X_X[r] = e^{-j \frac{2\pi (r+N/2)}{N}} X_{X0}[r] + \frac{N}{2} X_X[r+N/2]
  \]

- Can simplify:
  \[
  W_N^r X_{X1}[r] + \frac{N}{2} X_X[r+N/2] = -W_N^r X_{X1}[r] + \frac{N}{2} X_X[r+N/2]
  \]
  \[
  W_N^r = e^{-j \frac{2\pi r}{N}}
  \]
  \[
  = e^{-j \frac{2\pi N/2}{N}}
  \]
  \[
  = -W_N^r
  \]
  i.e. SUB rather than ADD
-1’s absorbed into summation nodes

$W_N^0$ disappears

‘in-place’ algorithm: sequential stages
FFT for Other Values of N

- Having $N = 2^M$ meant we could divide each stage into 2 halves = “radix-2 FFT”
- Same approach works for:
  - $N = 3^M$  radix-3
  - $N = 4^M$  radix-4 - more optimized radix-2
  - etc...
- Composite $N = a \cdot b \cdot c \cdot d \rightarrow$ mixed radix  
  (different $N/r$ point FFTs at each stage)
  - .. or just zero-pad to make $N = 2^M$
Inverse FFT

- Recall IDFT: \[ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk} \]

- Thus:
  \[ Nx'[n] = \sum_{k=0}^{N-1} \left( X[k] W_N^{-nk} \right)^* = \sum_{k=0}^{N-1} X'[k] W_N^{nk} \]

- Hence, use FFT to calculate IFFT:
  \[ x[n] = \frac{1}{N} \left[ \sum_{k=0}^{N-1} X'[k] W_N^{nk} \right]^* \]
DFT of Real Sequences

- If $x[n]$ is pure-real, DFT wastes mult’s
- Real $x[n] \rightarrow$ Conj. symm. $X[k] = X^*[-k]$
- Given two real sequences, $x[n]$ and $w[n]$ call $y[n] = j \cdot w[n]$ , $v[n] = x[n] + y[n]$

\[ X[k] = \frac{1}{2}(V[k]+V^*[-k]) \] , \[ W[k] = -j \frac{1}{2}(V[k]-V^*[-k]) \]

- i.e. compute DFTs of two $N$-pt real sequences with a single $N$-pt DFT
3. Short-Time Fourier Transform (STFT)

- Fourier Transform (e.g. DTFT) gives spectrum of an entire sequence:
- How to see a time-varying spectrum?
- e.g. slow AM of a sinusoid carrier:

\[ x[n] = \left( 1 - \cos \frac{2 \pi n}{N} \right) \cos \omega_0 n \]
Fourier Transform of AM Sine

- Spectrum of whole sequence indicates modulation indirectly...
  
- ... as cancellation between closely-tuned sines

\[
2cAcB = cA+B +cA-B
\]

\[
\frac{N\sin2\pi kn}{N} - \frac{N\sin2\pi(k-1)n}{N}
\]

\[
-\frac{N\sin2\pi(k+1)n}{2N}
\]
Fourier Transform of AM Sine

- Sometimes we’d rather separate modulation and carrier:
  \[ x[n] = A[n]\cos\omega_0 n \]
  - \( A[n] \) varies on a different (slower) timescale

- One approach:
  - chop \( x[n] \) into short sub-sequences ..
  - .. where slow modulator is \( \sim \) constant
  - DFT spectrum of pieces \( \rightarrow \) show variation
FT of Short Segments

- Break up \( x[n] \) into successive, shorter chunks of length \( N_{FT} \), then DFT each:

\[
\begin{align*}
\text{\( x[n] \)} & \quad \text{\( N_{FT} \)} \\
\text{\( x_0[n] \)} & \quad \text{\( x_1[n] \)} & \quad \text{\( x_2[n] \)} & \quad \text{\( x_3[n] \)} & \quad \text{\( x_4[n] \)} & \quad \text{\( x_5[n] \)} & \quad \text{\( x_6[n] \)} & \quad \text{\( x_7[n] \)} \\
\text{\( x_0[k] \)} & \quad \text{\( x_1[k] \)} & \quad \text{\( x_2[k] \)} & \quad \text{\( x_3[k] \)} & \quad \text{\( x_4[k] \)} & \quad \text{\( x_5[k] \)} & \quad \text{\( x_6[k] \)} & \quad \text{\( x_7[k] \)} \\
\end{align*}
\]

Shows amplitude modulation of \( \omega_0 \) energy
The Spectrogram

- Plot successive DFTs in time-frequency:

  \[
  X_i[k] \quad X_1[k] \quad X_2[k] \quad X_3[k] \quad X_4[k] \quad X_5[k] \quad X_6[k] \quad X_7[k]
  \]

  \[
  |X_i[k]| \quad k
  \]

  \[
  |X[k,n]| \quad k
  \]

  \[
  0 \quad 128 \quad 256 \quad 384 \quad 512 \quad 640 \quad 768 \quad 896 \quad 1024
  \]

  \[
  0 \quad 20 \quad 40 \quad 60 \quad 80 \quad 100 \quad 120
  \]

  \[
  \text{time hopsize (between successive frames)} = 128 \text{ points}
  \]

- This image is called the Spectrogram
Short-Time Fourier Transform

- Spectrogram = STFT magnitude plotted on time-frequency plane
- STFT is (DFT form):

\[
X[k,n_0] = \sum_{n=0}^{N_{FT}-1} x[n_0 + n] \cdot w[n] \cdot e^{-j\frac{2\pi kn}{N_{FT}}}
\]

- intensity as a function of time & frequency
STFT Window Shape

- $w[n]$ provides ‘time localization’ of STFT
  - e.g. rectangular
  - selects $x[n]$, $n_0 \leq n < n_0+N_W$

- But: resulting spectrum has same problems as windowing for FIR design:

$$X(e^{j\omega}, n_0) = \text{DTFT}\{x[n_0+n] \cdot w[n]\}$$

$$= \int_{-\pi}^{\pi} e^{j\theta n_0} X(e^{j\theta})W(e^{j(\omega-\theta)}) d\theta$$

spectrum of short-time window is convolved with (twisted) parent spectrum
**STFT Window Shape**

- e.g. if $x[n]$ is a pure sinusoid,

  $$X(e^{j\omega}) \ast W(e^{j\omega}) = W(e^{j\omega})$$

  - **blurring (mainlobe)**
  - + **ghosting (sidelobes)**

- Hence, use **tapered window** for $w[n]$

  e.g. Hamming

  $$w[n] = 0.54 + 0.46 \cos\left(2\pi \frac{n}{2M+1}\right)$$

  - **sidelobes** $<-40$ dB
STFT Window Length

- **Length of** $w[n]$ **sets temporal resolution**

  - Short window measures only local properties
  - Longer window averages spectral character

- **Window length** $\propto \frac{1}{\text{Mainlobe width}}$

- Shorter window $\rightarrow$ more blurred spectrum

- *more time detail* $\leftrightarrow$ *less frequency detail*
STFT Window Length

- Can illustrate time-frequency tradeoff on the time-frequency plane:

  disks show ‘blurring’ due to window length; area of disk is constant

  → Uncertainty principle: \( \delta f \cdot \delta t \geq k \)

- Alternate tilings of time-freq:

  half-length window → half as many DFT samples
Spectrograms of Real Sounds

- Time-domain
- Successive short DFTs
- Individual t-f cells merge into continuous image
Narrowband vs. Wideband

- Effect of varying window length:

![Waveform comparison](image-url)
Spectrogram in Matlab

```
>> [d, sr] = wavread('mpgrl_sx419.wav');
>> Nw = 256;
>> specgram(d, Nw, sr)
>> caxis([-80 0])
>> colorbar
```

(hann) window length

actual sampling rate (to label time axis)

dB
STFT as a Filterbank

Consider one ‘row’ of STFT:

\[
X_k[n_0] = \sum_{n=0}^{N-1} x[n_0 + n] \cdot w[n] \cdot e^{-j\frac{2\pi kn}{N}}
\]

\[
= \sum_{m=0}^{-(N-1)} h_k[m] x[n_0 - m]
\]

where \( h_k[n] = w[-n] \cdot e^{j\frac{2\pi kn}{N}} \)

- Each STFT row is output of a filter (subsampled by the STFT hop size)
STFT as a Filterbank

- If \( h_k[n] = w[(-)n] \cdot e^{j\frac{2\pi kn}{N}} \)

  then \( H_k(e^{j\omega}) = W(e^{(-)j(\omega-\frac{2\pi k}{N})}) \)

- Each STFT row is the same bandpass response defined by \( W(e^{j\omega}) \), frequency-shifted to a given DFT bin:

\[
\begin{align*}
|W(e^{j\omega})| & \\
|H_1(e^{j\omega})| & \\
|H_2(e^{j\omega})| & \\
\vdots & \\
\end{align*}
\]

A bank of identical, frequency-shifted bandpass filters: “filterbank”
STFT Analysis-Synthesis

- IDFT of STFT frames can reconstruct (part of) original waveform

- e.g. if \( X[k,n_0] = \text{DFT}\{x[n_0 + n] \cdot w[n]\} \) then \( \text{IDFT}\{X[k,n_0]\} = x[n_0 + n] \cdot w[n] \)

- Can shift by \( n_0 \), combine, to get \( \hat{x}[n] \):

\[
\hat{x}[n] = x[n_0 + n] \cdot w[n]
\]

- Could divide by \( w[n-n_0] \) to recover \( x[n] \)...
STFT Analysis-Synthesis

- Dividing by small values of \( w[n] \) is bad

- Prefer to overlap windows:
  
  i.e. sample \( X[k,n_0] \)

  at \( n_0 = r \cdot H \) where \( H = N/2 \) (for example)

- Then \( \hat{x}[n] = \sum_r x[n]w[n - rH] \)
  
  \[ = x[n] \quad \text{if} \quad \sum_{\forall r} w[n - rH] = 1 \]
STFT Analysis-Synthesis

- Hann or Hamming windows with 50% overlap sum to constant

\[
(0.54 + 0.46 \cos(2\pi \frac{n}{N})) + (0.54 + 0.46 \cos(2\pi \frac{n-N}{2N})) = 1.08
\]

- Can modify individual frames of \(X[k,n]\) and then reconstruct
  - complex, time-varying modifications
  - tapered overlap makes things OK
STFT Analysis-Synthesis

- e.g. Noise reduction:

  STFT of original speech

  Speech corrupted by white noise

  Energy threshold mask