CMPE 352
Signal Processing & Algorithms

Spring 2019

Sedat Ölçer

13 May, 2019

The Fast Fourier Transform (FFT)

1. Start with the DFT equation for $G[k]$:

$$G[k] = \sum_{n=0}^{N-1} g[n] e^{-jk\Omega_0 n} \qquad 0 \le k \le N-1 \qquad (\Omega_0 = \frac{2\pi}{N})$$

2. Assume *N* to be even. Split the time samples into even/odd sequences ($N' = N/2$):

$$G[k] = \underbrace{\sum_{m=0}^{N'-1} g_e[m] e^{-jm\Omega_0' k}}_{G_e[k]} + e^{-j\Omega_0 k} \underbrace{\sum_{m=0}^{N'-1} g_o[m] e^{-jm\Omega_0' k}}_{G_o[k]} \qquad 0 \le k \le N-1 \qquad (\Omega_o' = \frac{2\pi}{N'})$$

(obtained from the <u>even</u> samples) (obtained from the <u>odd</u> samples)

3. Use the following facts: $G_e[k+N'] = G_e[k]$, and $G_o[k+N'] = G_o[k]$ $\quad$ (N' −periodic)

and: $e^{-j(k+N')\Omega_0} = -e^{-jk\Omega_0}$

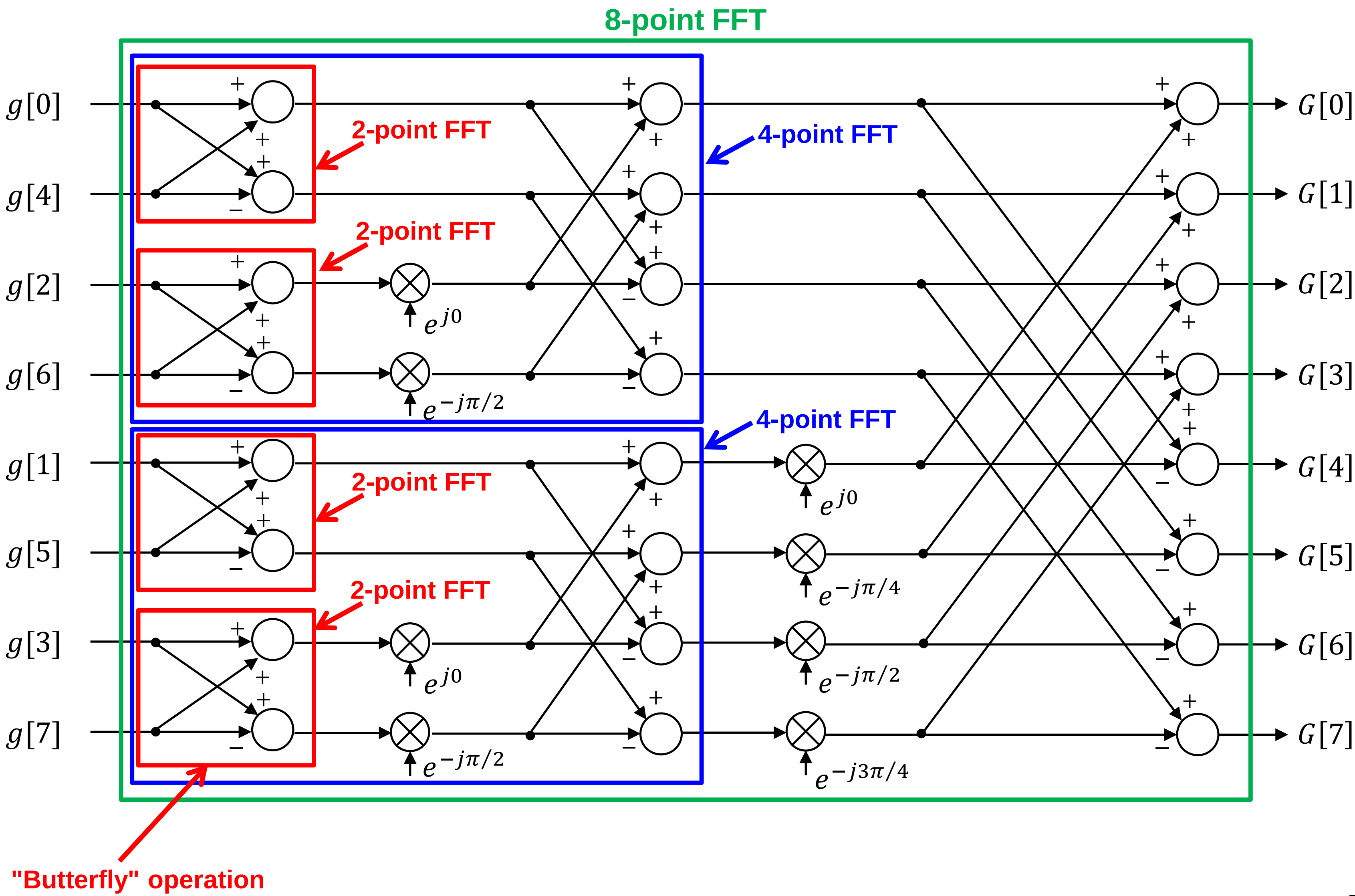
4. Rewrite $G[k]$ as:

First N' values of $G[k]$:	$G[k] = G_e[k] + e^{-j\Omega_0 k} G_o[k]$	for $0 \le k \le N'-1$
Second N' values of $G[k]$:	$G[k+N'] = G_e[k] - e^{-j\Omega_0 k} G_o[k]$	for $0 \le k \le N'-1$

5. Assuming N' to be even, obtain each of $G_e[k]$ and $G_o[k]$ through a N' −point DFT (apply the above algorithm for each DFT)

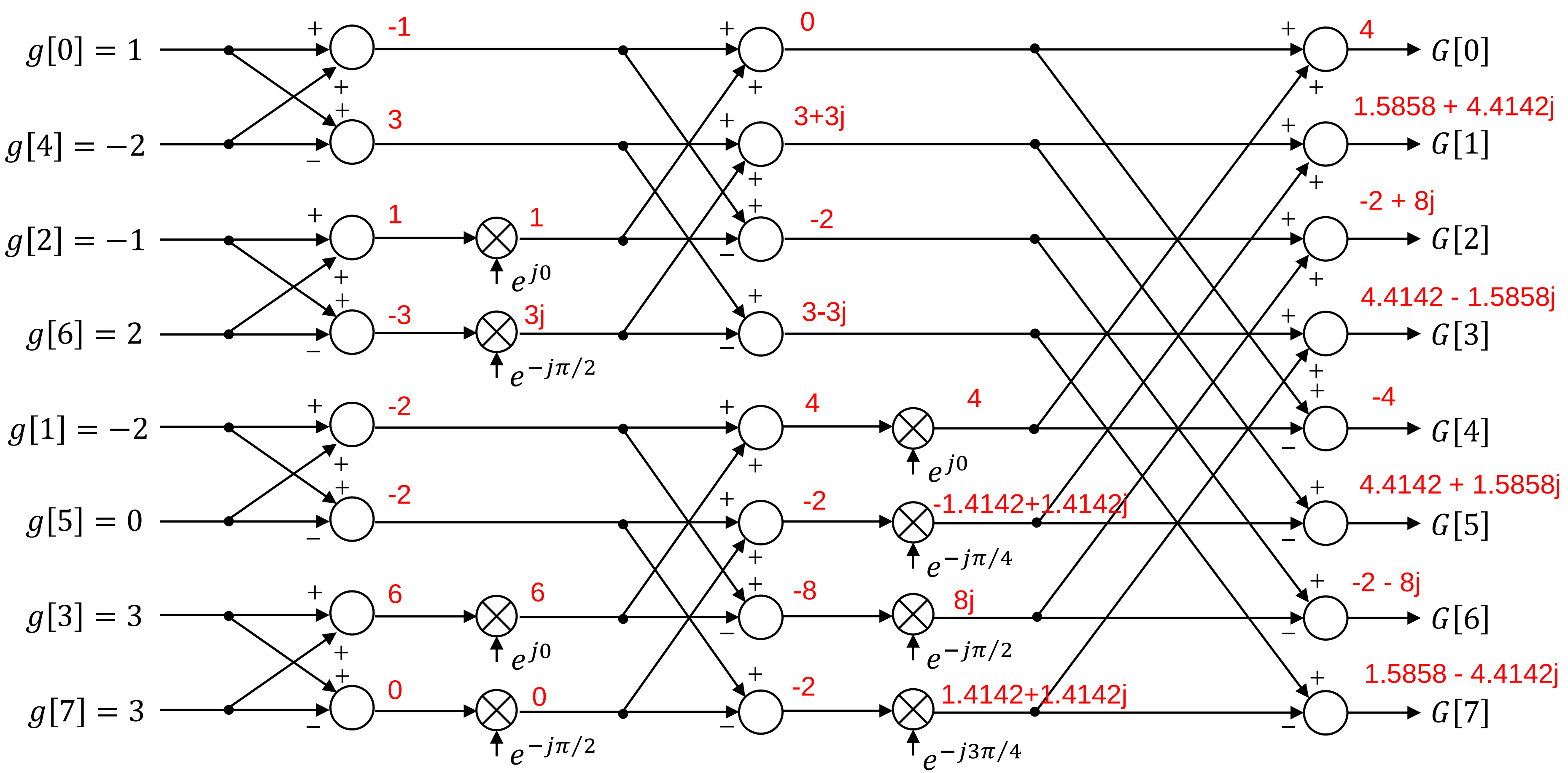
6. Repeat until the 2 −point DFT is reached (hence *N* must be a power of 2)

Example: 8-Point FFT



Example: 8-Point FFT

g = [1, -2, -1, 3, -2, 0 , 2, 3];



Matlab FFT function fft(.)

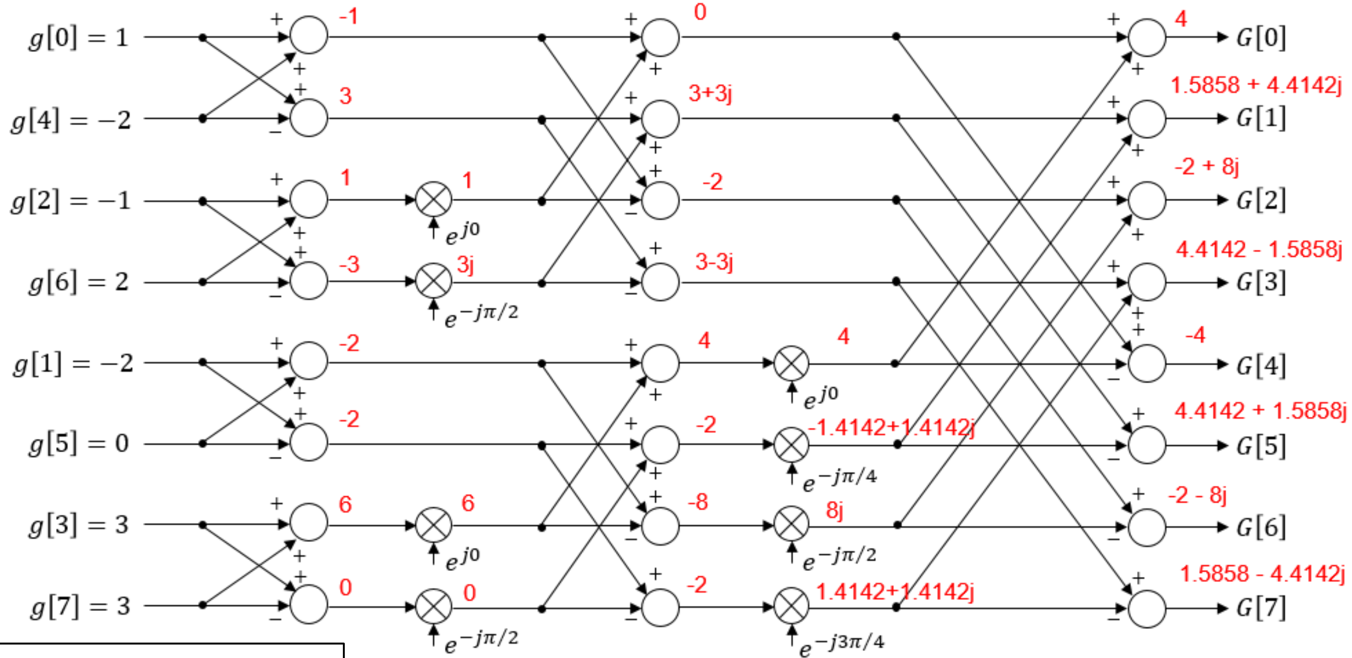
This MATLAB function returns the discrete Fourier transform (DFT) of vector x, computed with a fast Fourier transform (FFT) algorithm.

```
Y = fft(x)
Y = fft(X,n)
Y = fft(X,[],dim)
Y = fft(X,n,dim)
```

- If N is a power of 2, the FFT algorithm is automatically applied
  Otherwise the DFT algorithm is applied

Matlab FFT function fft(.) -- Example

g = [1, -2, -1, 3, -2, 0 , 2, 3];




```
>> x=[1; -2; -1; 3; -2; 0 ; 2; 3];
>> fft(x)

ans =

   4.0000 + 0.0000i
   1.5858 + 4.4142i
  -2.0000 + 8.0000i
   4.4142 - 1.5858i
  -4.0000 + 0.0000i
   4.4142 + 1.5858i
  -2.0000 - 8.0000i
   1.5858 - 4.4142i
```

Applications of the FFT Algorithm

- The Fast Fourier Transform (FFT) algorithm is pervasive in computer engineering and computer science
- Essential for the understanding of signals (design of embedded systems, image processing, face recognition, AI applications, etc.)
- Compression algorithms such JPEG, MPEG, MP3, etc., are all based on the FFT algorithm and its principles
- Data transmission (cell phones (3G/4.5G/5G), ADSL, VDSL, etc.) all implement the FFT algorithm
- ”The FFT is used billions of times everyday”

What did we learn this semester?

- What are signals, what are systems?
- Continuous versus discrete – the analog world and the digital world

- Types of signals: periodic and nonperiodic, even and odd
- Basic signal operations: time shifting, scaling and reversing
- Elementary signals: unit impulse, unit step, exponential, sinusoidal
- **Notion of frequency**, harmonic signals, and bandwidth
- Importance of the complex exponential

- The Fourier Series (periodic signals) – various forms
- Amplitude (or magnitude) and phase spectra

- Signal energy and signal power
- Decibel

- The Fourier Transform (nonperiodic signals)
- Properties
- Amplitude (or magnitude) and phase spectra

- System Frequency response
- Filters

What did we learn this semester?

- Sampling process, A/D conversion
- The Sampling Theorem
- Effect of sampling on signal spectrum – replication, aliasing
- Anti-aliasing low-pass filtering, analog reconstruction
- More on signal sampling and A/D conversion
- Quantizers, quantizing noise, PCM

- Derivation, integration, time-averaging algorithms

- The Discrete Fourier Transform (DFT) algorithm

- The Fast Fourier Transform (FFT) algorithm