6.300 Signal Processing

Week 8, Lecture B:
Discrete Fourier Transform (II)

- Resolution in Time and Frequency
- Circular convolution

Lecture slides are available on CATSOOP:

https://sigproc.mit.edu/fall24

Discrete Fourier Transform

A new Fourier representation for DT signals:

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

Synthesis equation

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

Analysis equation

The DFT has a number of features that make it particular convenient

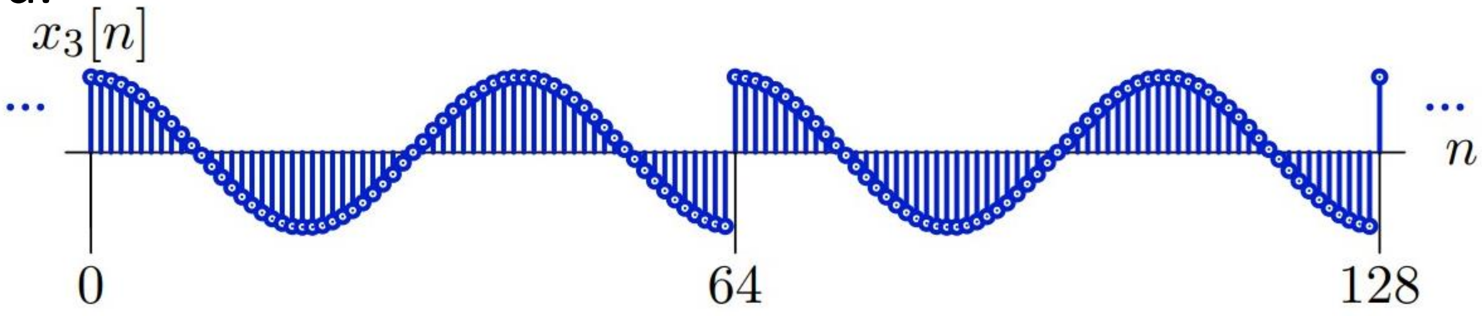
- It is not limited to periodic signals.
- It is discrete in both domains, making it computationally feasible

The FFT (**F**ast **F**ourier **T**ransform) is an algorithm for computing the DFT efficiently.

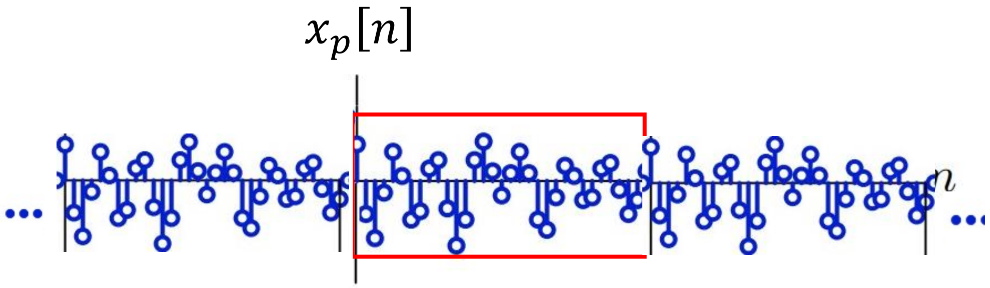
Two Ways to Think About DFT

We can think about the DFT in two different ways:

1. Think about DFT as **Fourier series** of N samples of the signal, periodically extended.

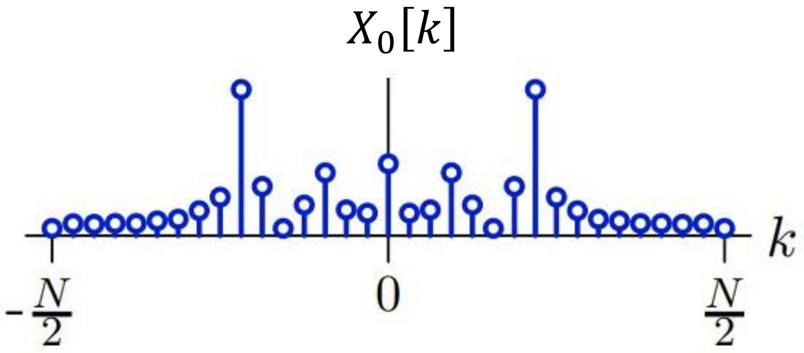


We can see why DFT of a single sinusoid is not concentrated in a single k component



iDFT

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



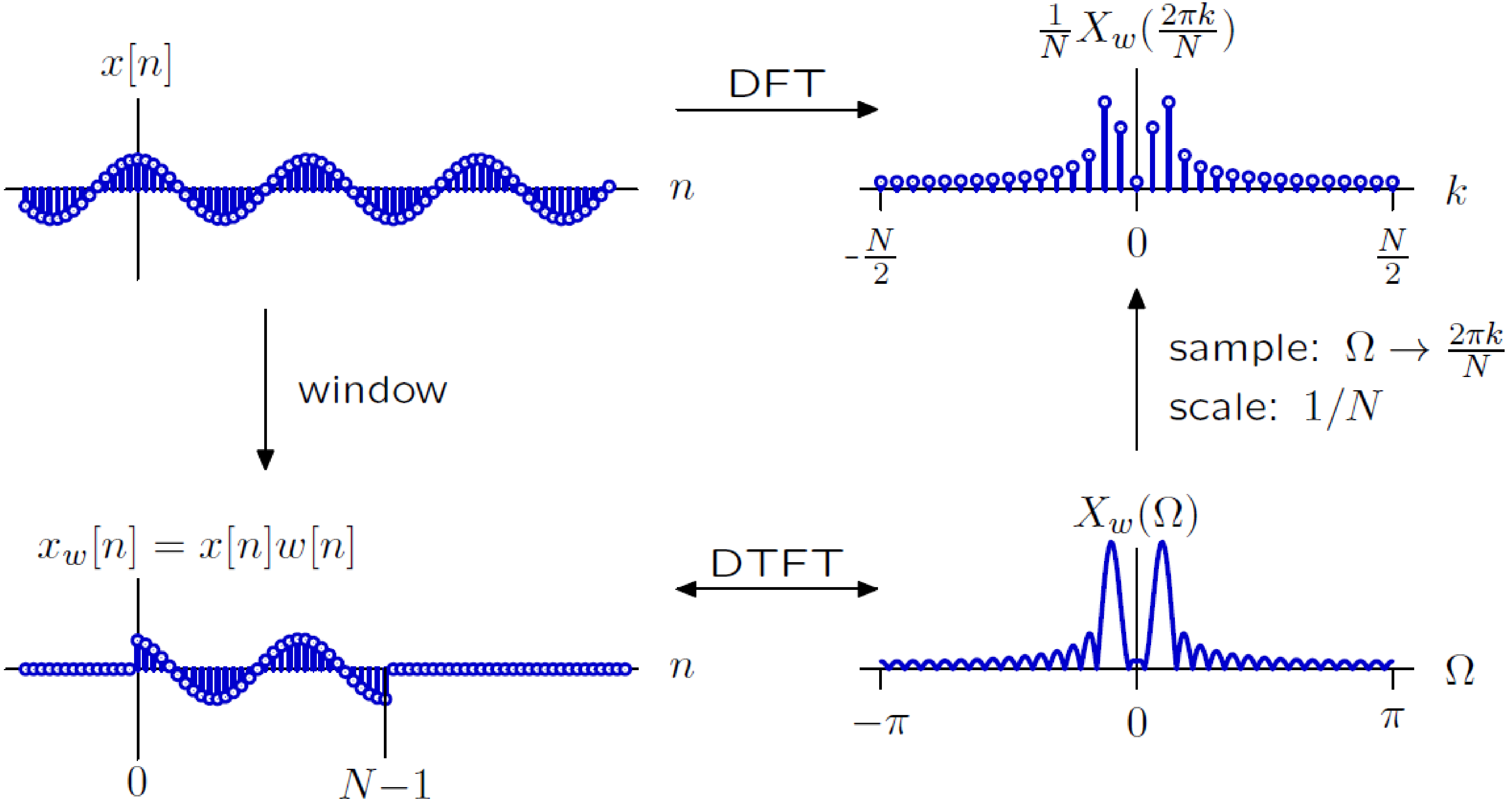
Two Ways to Think About DFT

We can think about the DFT in two different ways:

1. Think about DFT as **Fourier series** of N samples of the signal, periodically extended.

2. Think about DFT as the scaled **Fourier transform** of a “windowed” version of the original signal.

DFT: Relation to DTFT

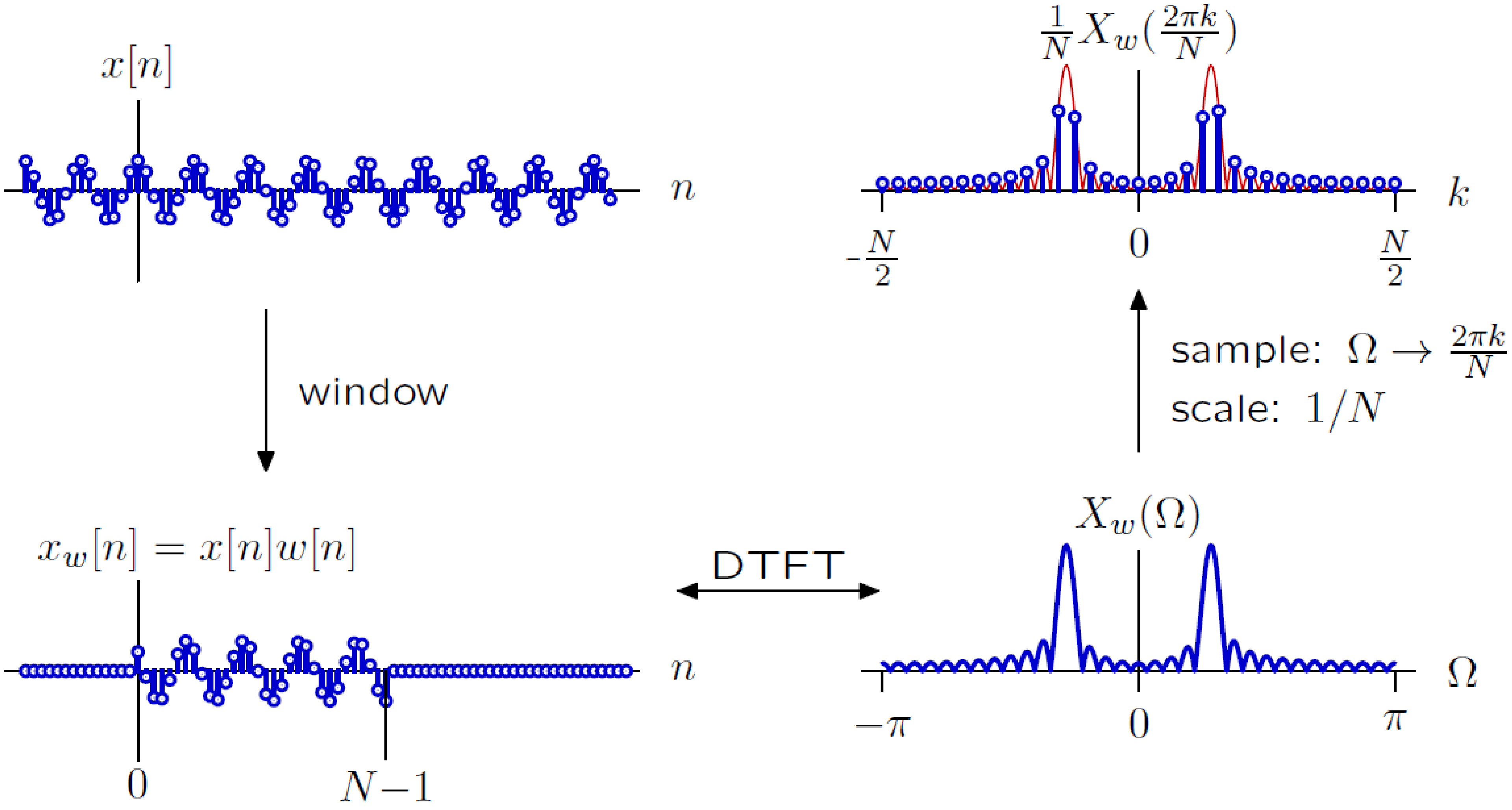


While sampling and scaling are important, it is the **windowing** that most affects frequency content.

DFT: Relation to DTFT

Decreasing the analysis window N decreases frequency resolution.

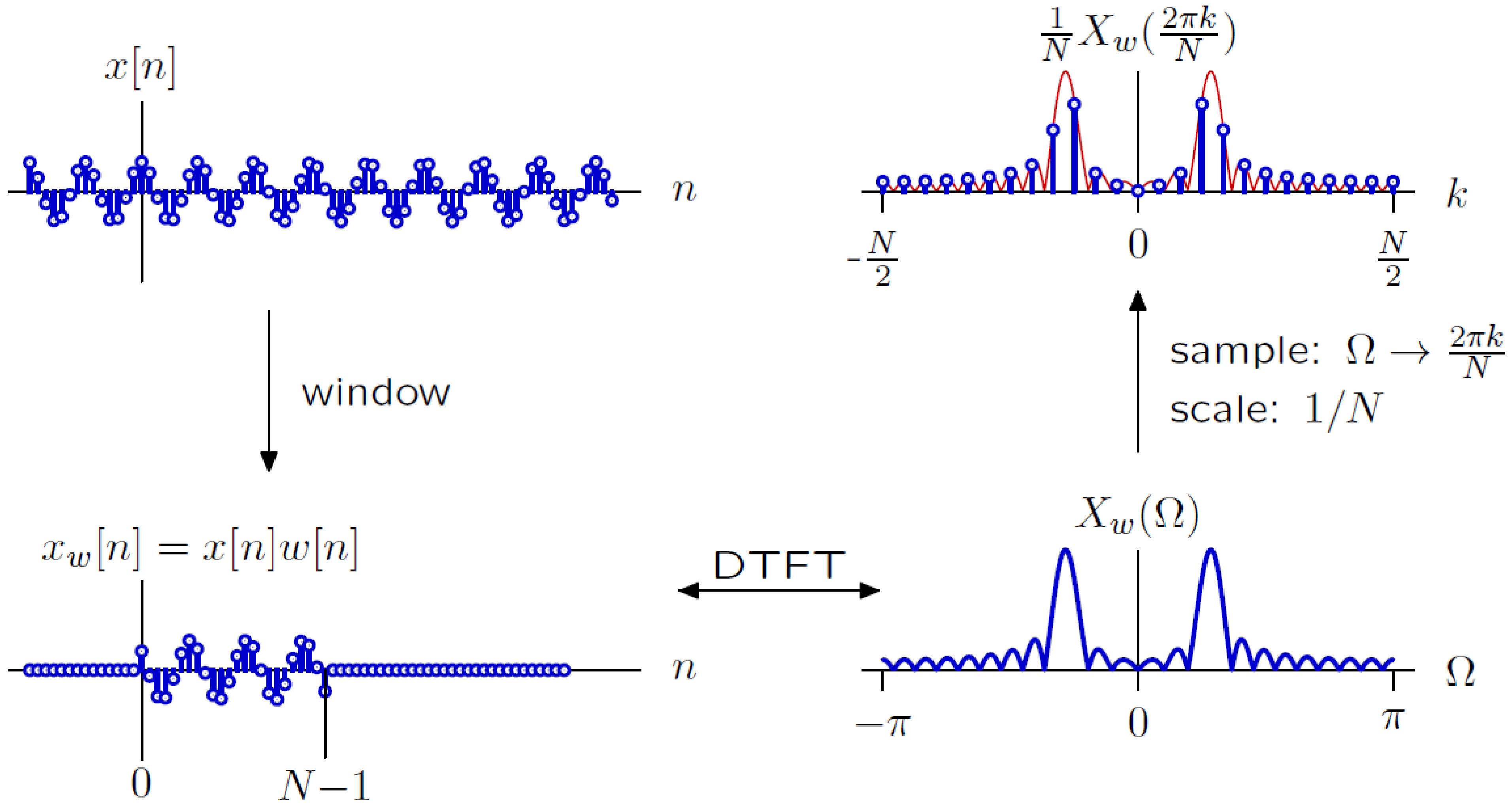
$N = 32$



DFT: Relation to DTFT

Decreasing the analysis window N decreases frequency resolution.

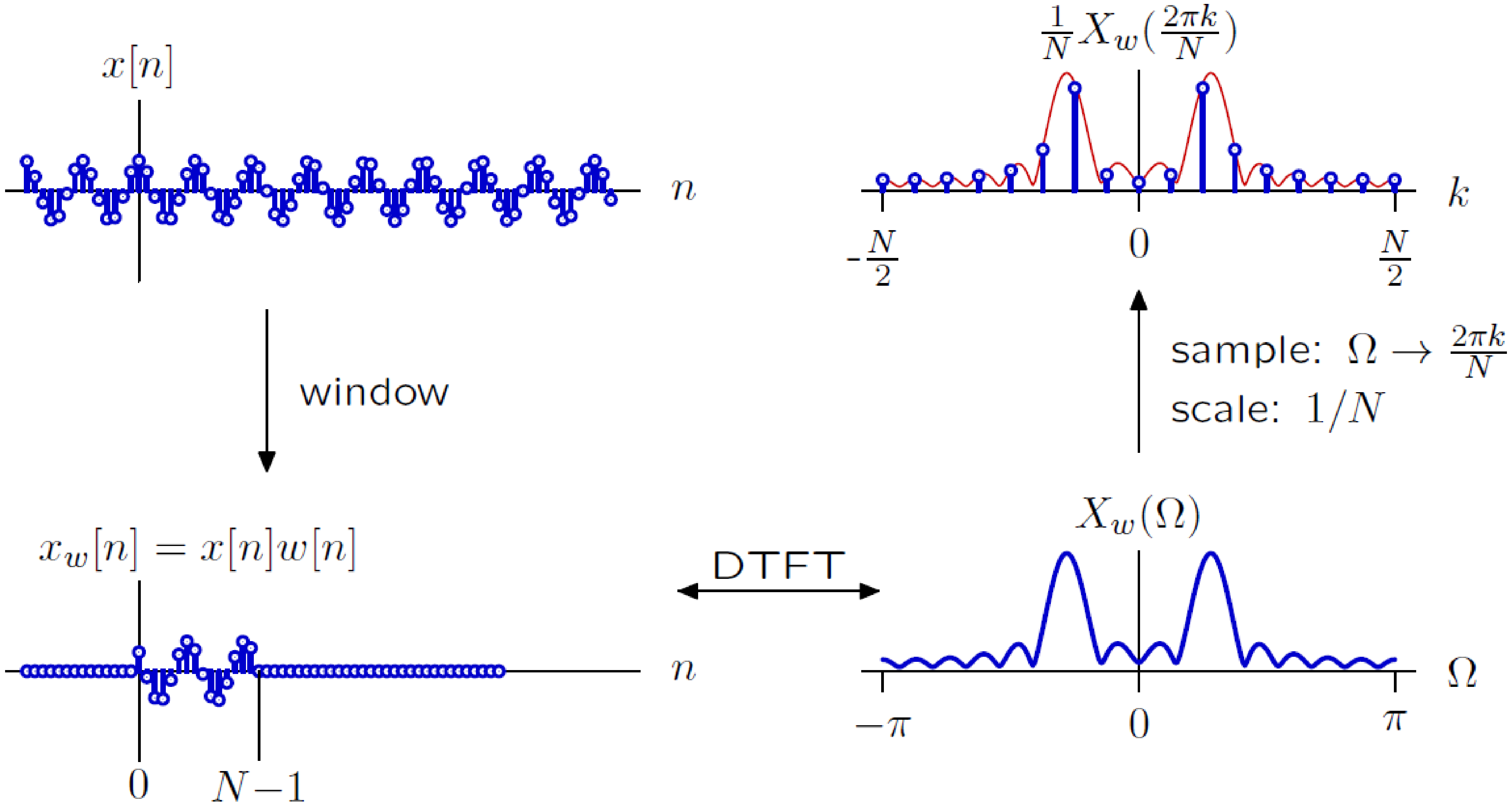
$N = 24$



DFT: Relation to DTFT

Decreasing the analysis window N decreases frequency resolution.

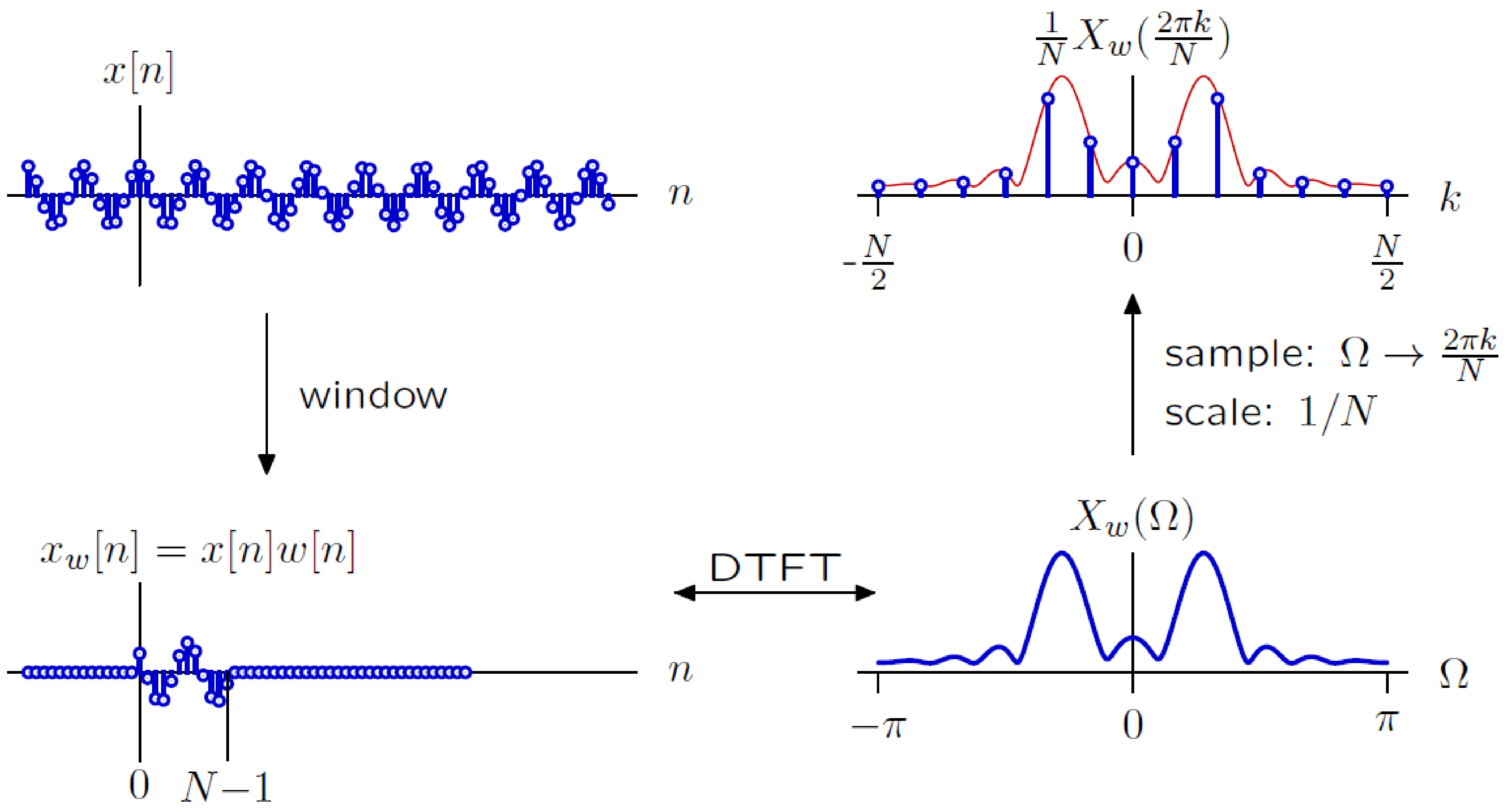
$N = 16$



DFT: Relation to DTFT

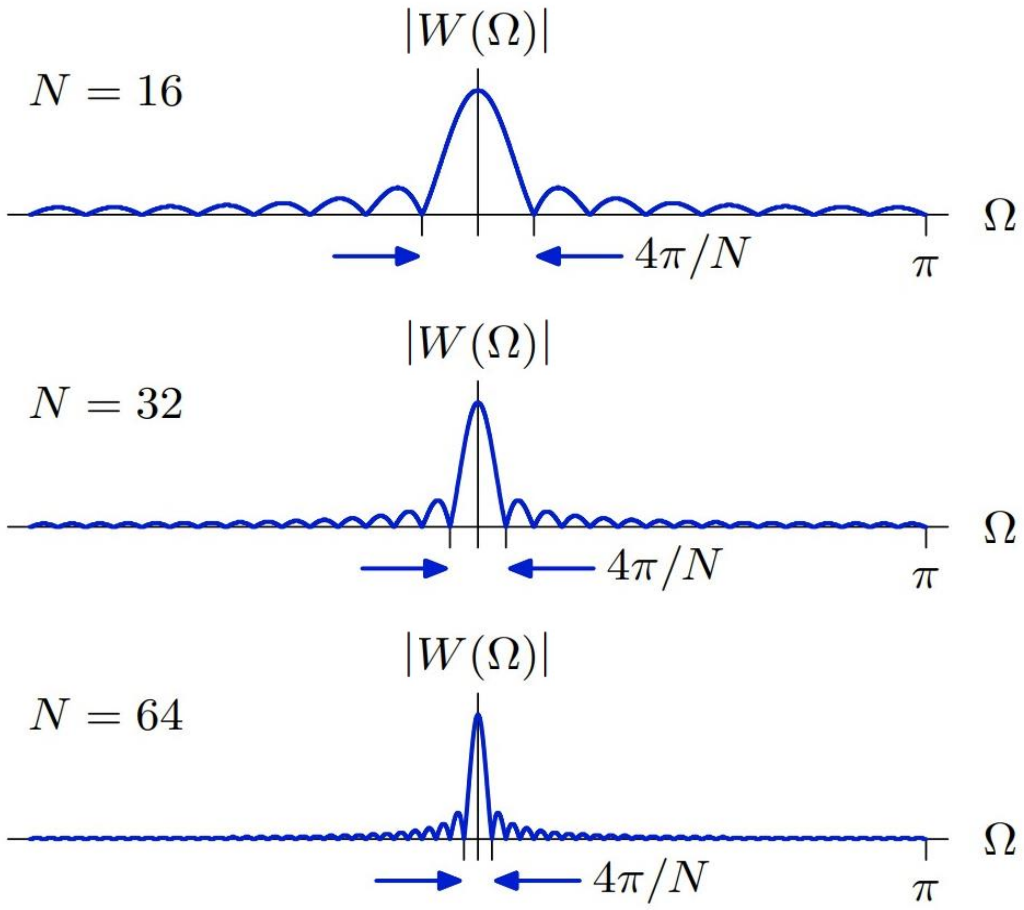
Decreasing the analysis window N decreases frequency resolution.

$N = 12$



Frequency Resolution

Frequency blurring is fundamental to the way DFT works. Longer windows provide finer frequency resolution.



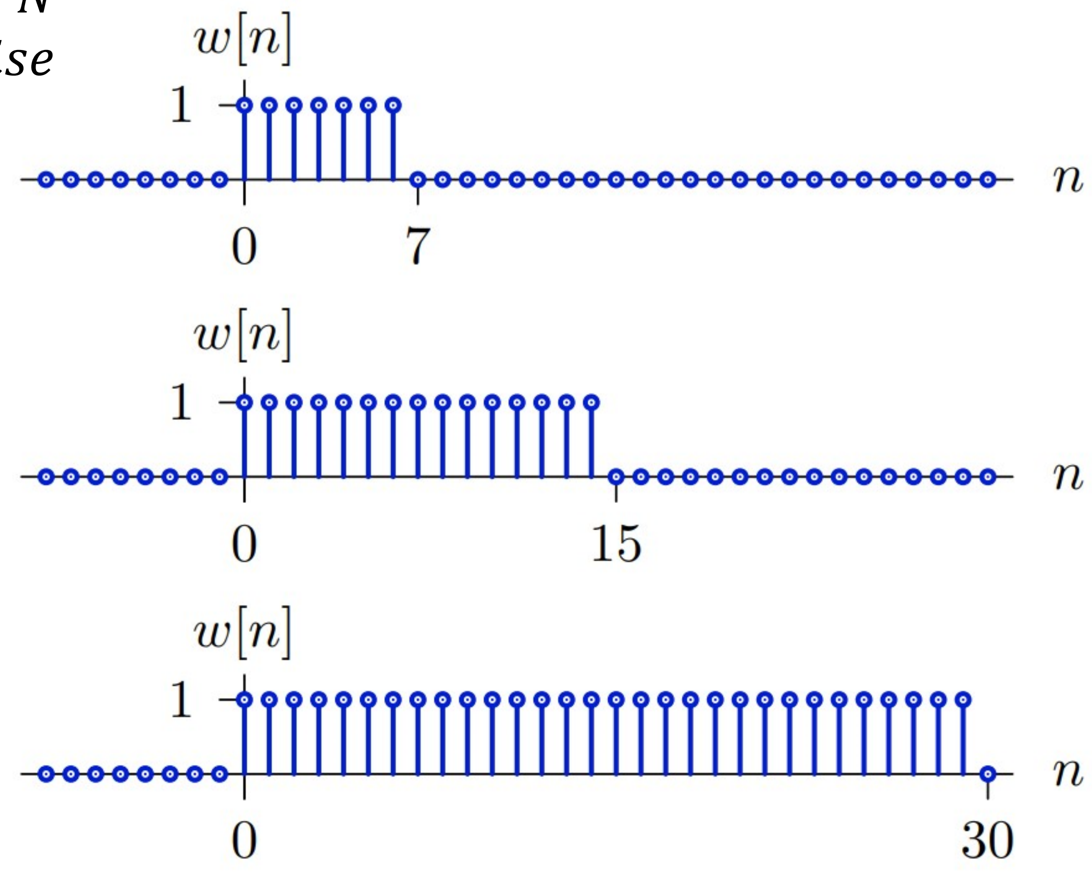
$$W(\Omega) = \frac{\sin(\Omega \frac{N}{2})}{\sin(\frac{\Omega}{2})} e^{-j\Omega\frac{N-1}{2}}$$

The width of the central lobe is inversely related to window length N.

Spectral Blurring & Time/Frequency Tradeoff

However, longer windows provide less temporal resolution.

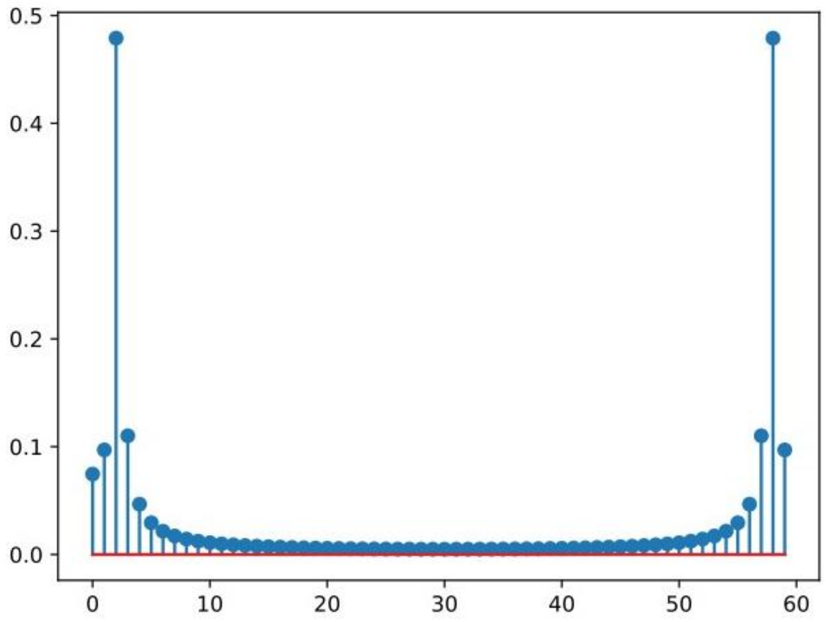
$$w[n] = \begin{cases} 1 & 0 \leq n < N \\ 0 & otherwise \end{cases}$$



→ fundamental tradeoff between resolution in frequency and time.

Check yourself

Consider a waveform containing a single, pure sinusoid. This waveform was recorded with a sampling rate of 8kHz, and we have 60 samples of the waveform. Computing the DFT magnitudes, we find:



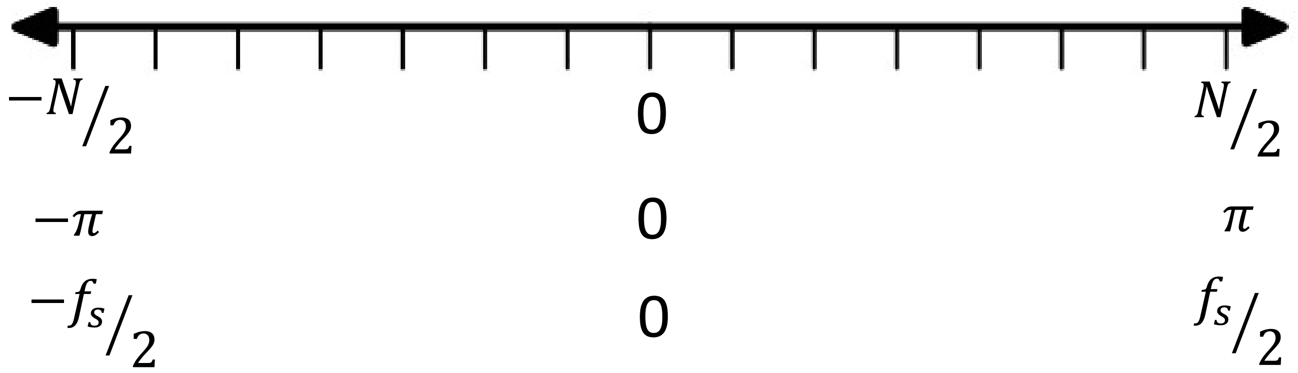
What note is being played? How accurately can we tell?

Frequency Resolution

We only have N distinct samples of the DTFT.

$$\Omega = \frac{2\pi k}{N}$$

$$\Omega = \frac{2\pi f}{f_s}$$



k: integer (frequency)

Ω: rad/sample

f: cycles/second (Hz)

We're uniformly breaking up a range of 2π into N discrete samples: the spacing between samples is **2π/N** . The kth coefficient is associated with **Ω = 2πk/N**

In Hz, the spacing between samples is **fs/N** . Thus, the kth coefficient is associated with a frequency of **f = kfs/N**.

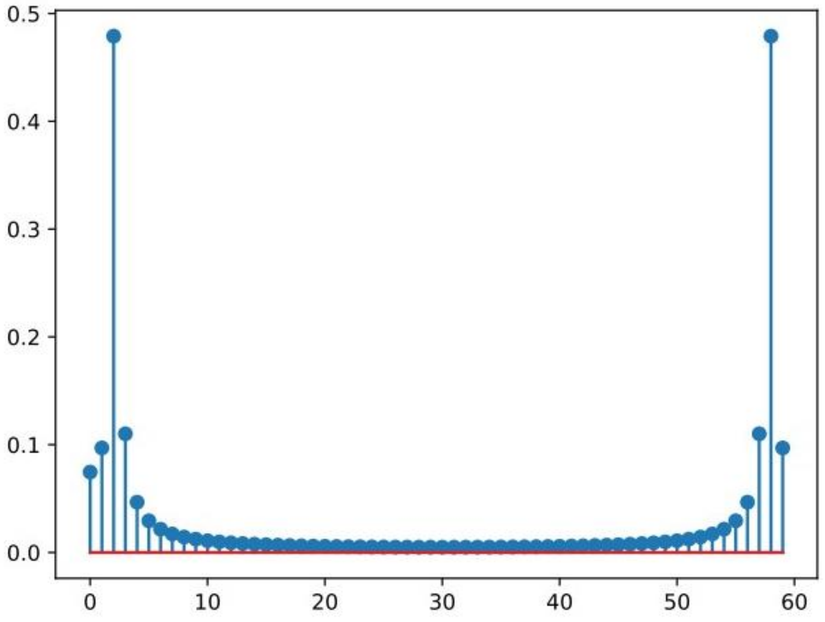
Trade-off: increasing frequency resolution necessarily requires considering more samples of the signal (i.e., increasing N)

Check yourself

Consider a waveform containing a single, pure sinusoid. This waveform was recorded with a sampling rate of 8kHz, and we have 60 samples of the waveform. Computing the DFT magnitudes, we find:

$$If \ \frac{f_s}{N} = 1 \text{ Hz}$$
$$\text{N} = 8000$$

$$If \ \frac{f_s}{N} = 0.1 \text{ Hz}$$
$$\text{N} = 80000$$



$$f = \frac{f_s k}{N} = \frac{8000*2}{60} = 266.7 \text{ (Hz)}$$

$$peak \ @ \ k = 2, -2$$

Middle C?

Frequency resolution:

$$\frac{f_s}{N} = \frac{8000}{60} = 133.3$$

What note is being played? How accurately can we tell?

How many samples do we need to consider in order to be able to determine the frequency of the tone to within 1Hz? Within 0.1Hz?

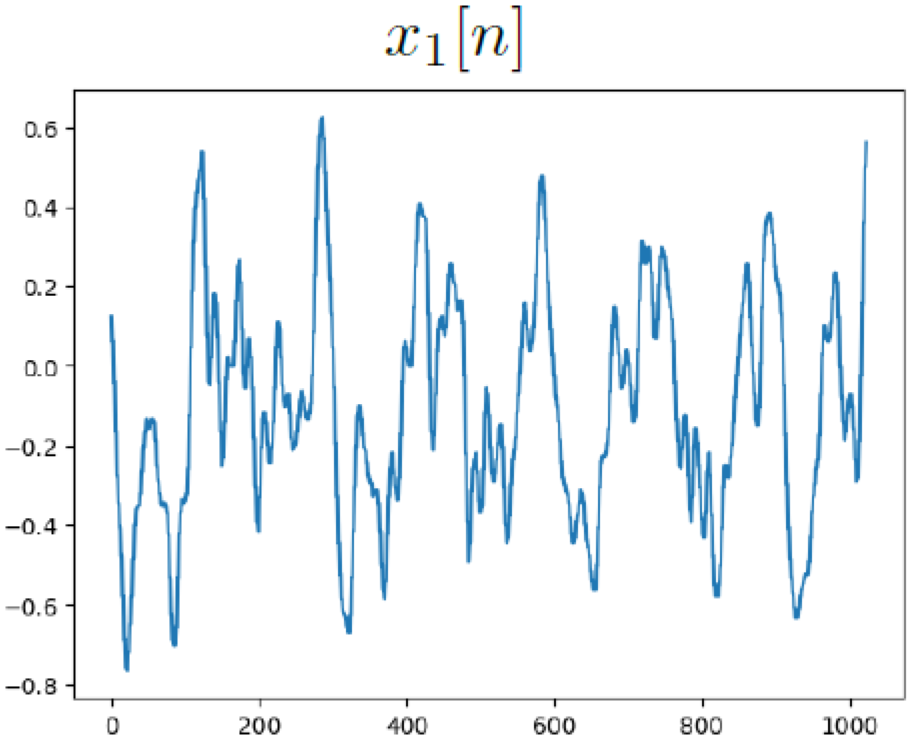
Frequency Resolution

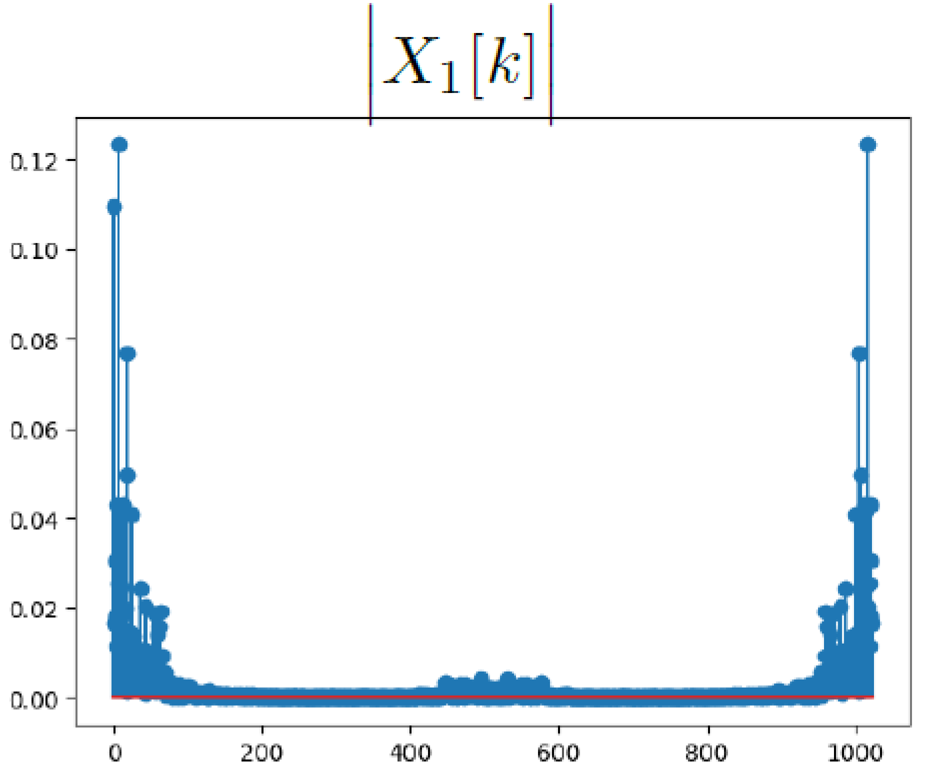
Example: Determine the frequency content of the following sounds.

cello: DEb3.wav (fs = 44, 100 Hz)

Frequency Resolution

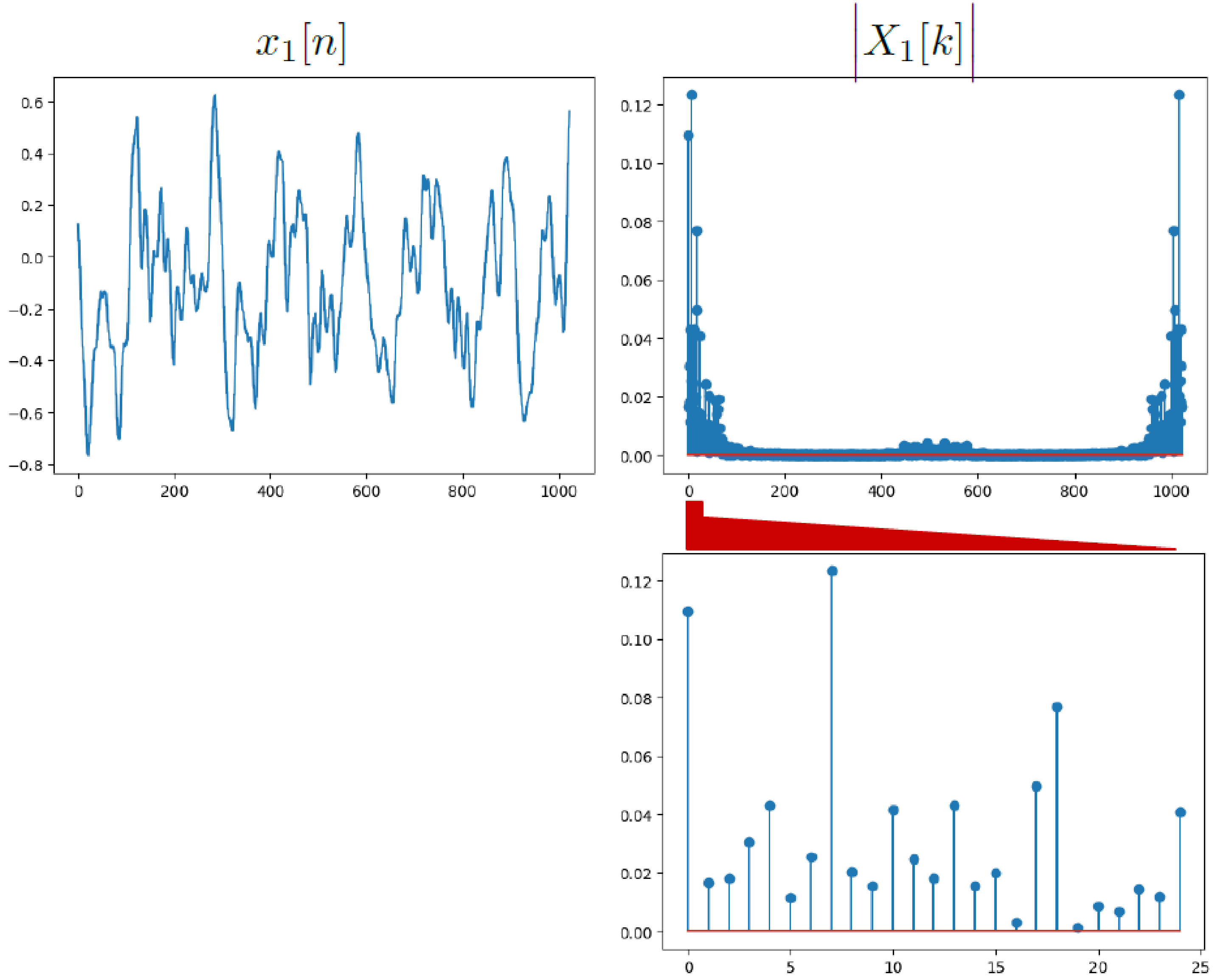
Extract 1024 samples and calculate DFT.





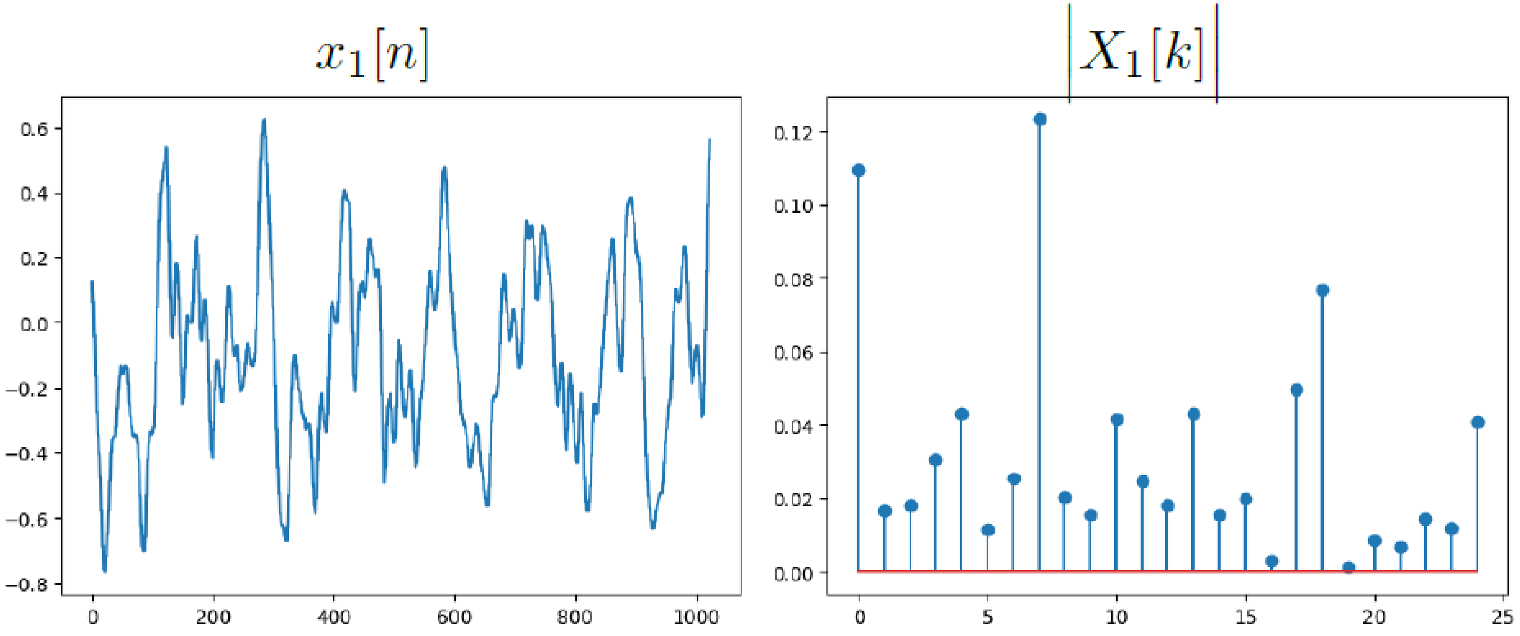
Frequency Resolution

Information about pitch is at low frequencies. Zoom in on $k = 0$ to 25.



Frequency Resolution

Information about pitch is at low frequencies. Zoom in on $k = 0$ to 25.



The biggest amplitude is at $k = 7$.

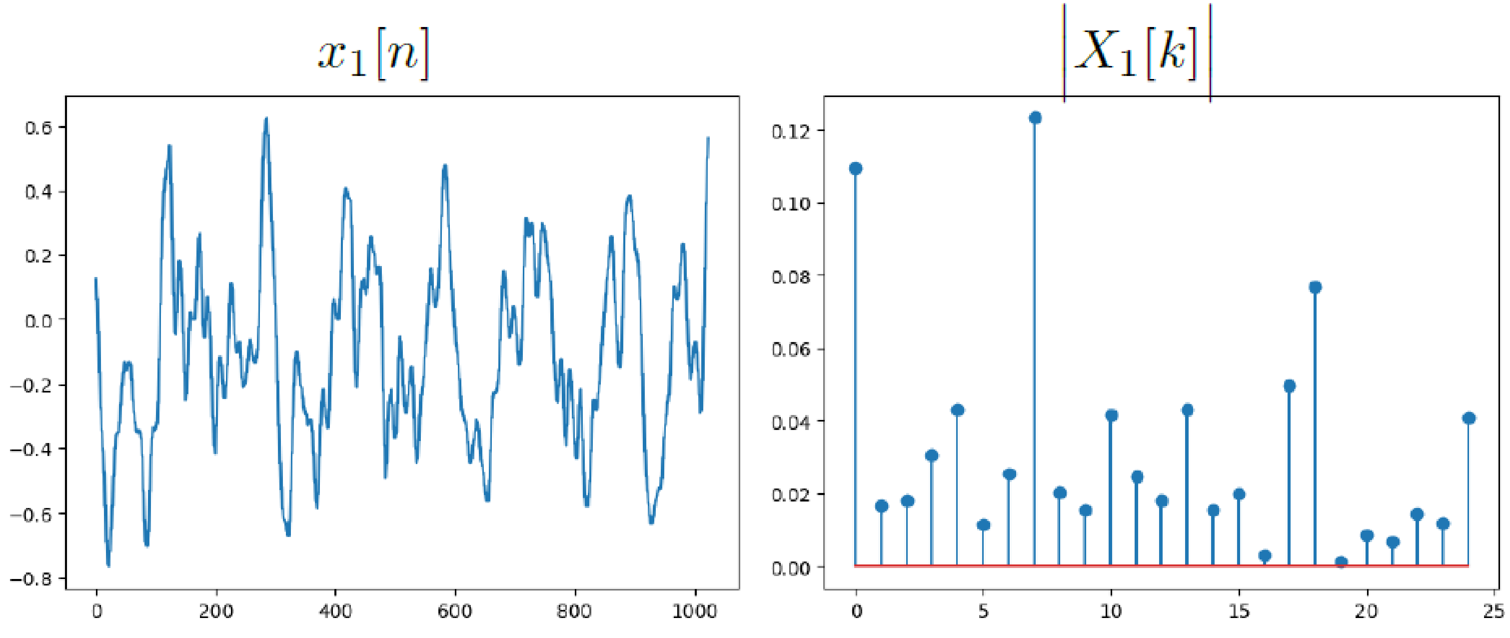
The corresponding frequency (in Hz) follows from proportional reasoning:

$$\frac{f_o}{f_s} = \frac{k_o}{N} \rightarrow f_o = \frac{k_o}{N} f_s = \frac{7}{1024} \times 44100 \approx 301.46\,\text{Hz}$$

This frequency is between D ($293.66\,$Hz) and E-flat ($311.13\,$Hz).

Frequency Resolution

Information about pitch is at low frequencies. Zoom in on $k = 0$ to 25.



The DFT provides integer resolution in k. Therefore, the peak at $k = 7$ could be off by as much as $\pm\frac{1}{2}$.

$$\Delta f = \frac{\Delta k}{N} f_s = \frac{1/2}{1024} \times 44100 \approx 21.5\,\text{Hz}$$

Thus the frequency of the biggest peak is $280 < f_o < 323$, easily including both D ($293.66\,$Hz) and E-flat ($311.13\,$Hz).

Improving Frequency Resolution

We can increase N to increase the number of analyzed frequencies.

Two methods to increase N:

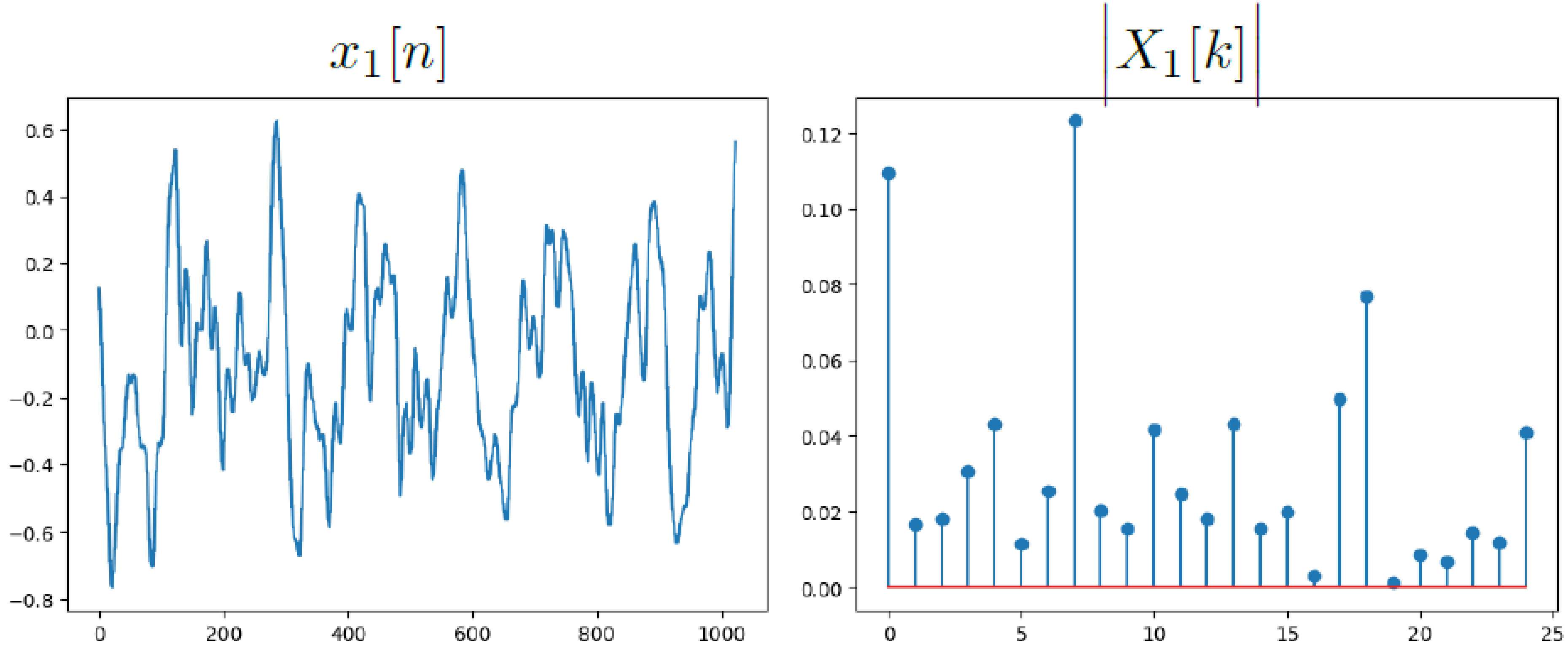
- zero-padding (add zeros to increase length of input)
- increase sample size

Can both methods help us resolve which note was played? What do you think?

Participation question for Lecture

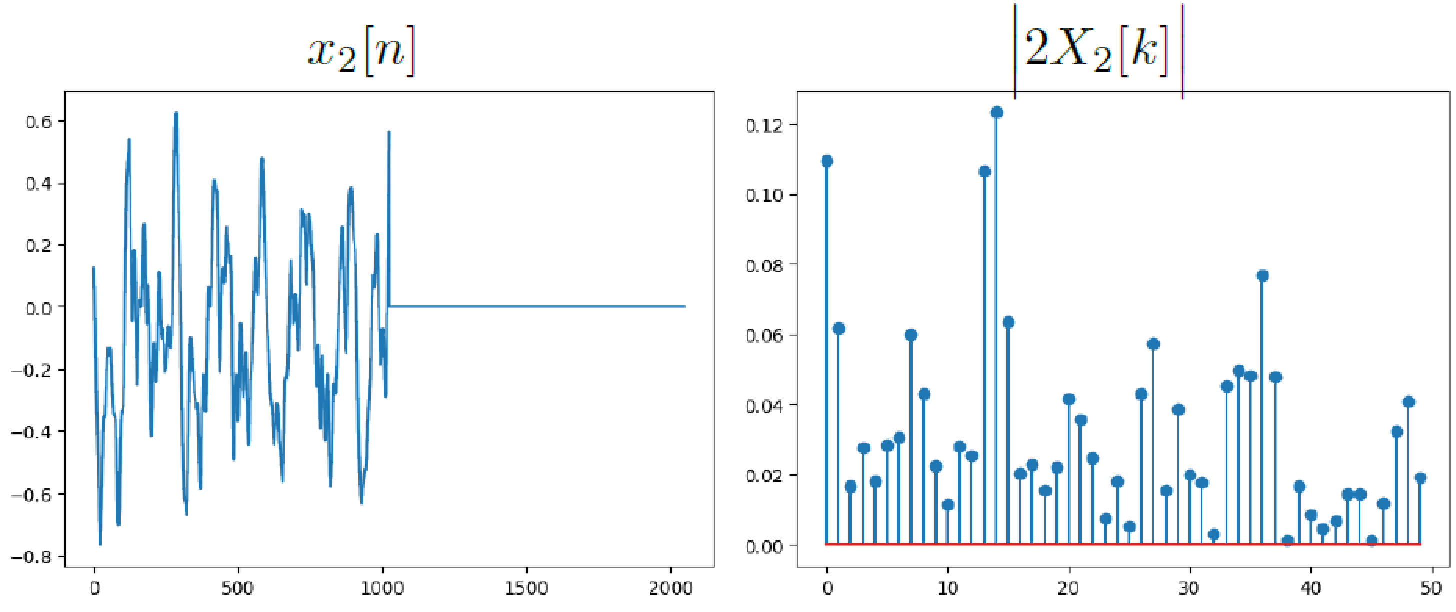
Zero Padding

Original (N=1024).



Zero Padding

What happens if we increase the length of the signal by adding zeros?



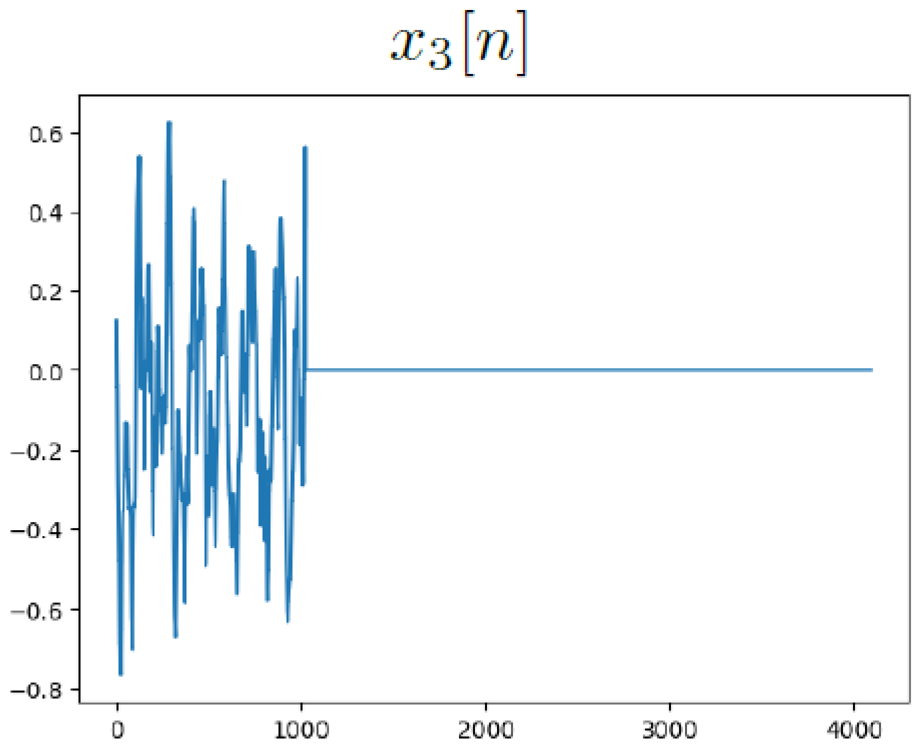
$$X_2[k] = \frac{1}{2N}\sum_{n=0}^{2N-1} x_2[n]e^{-j\frac{2\pi k}{2N}n} = \frac{1}{2N}\sum_{n=0}^{N-1} x_1[n]e^{-j\frac{2\pi k}{2N}n}$$

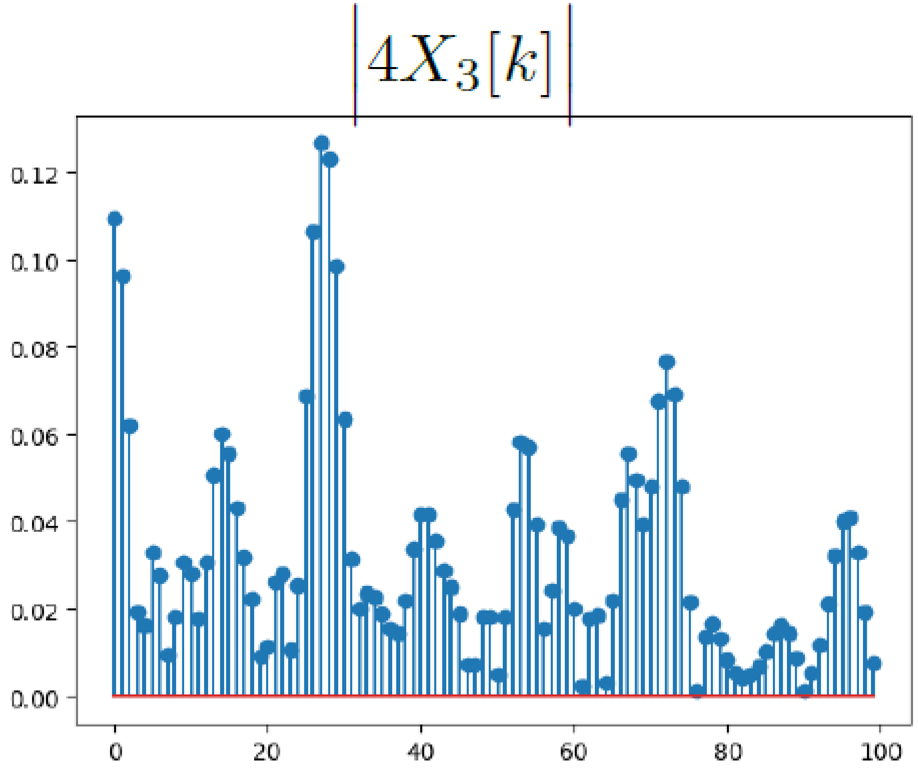
$$= \begin{cases} \frac{1}{2}X_1[k/2] & \text{if } k \text{ is even} \\ \text{new information} & \text{if } k \text{ is odd} \end{cases}$$

Lengthening $x_1[n]$ with zeros stretches the DFT by inserting new coefficients of X_2 between adjacent coefficients of X_1.

Zero Padding

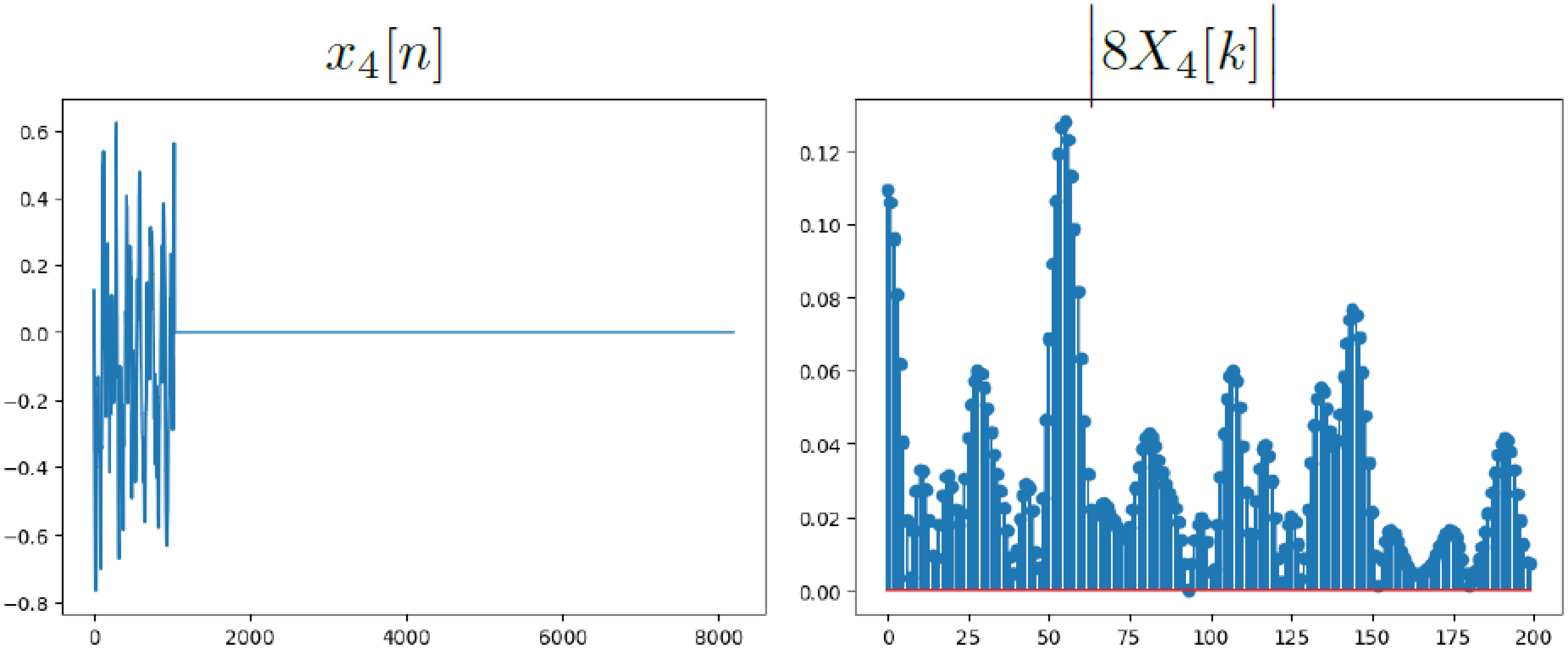
Lengthen by a factor of 4 (N=4096).





Zero Padding

Lengthen by a factor of 8 (N=8192).

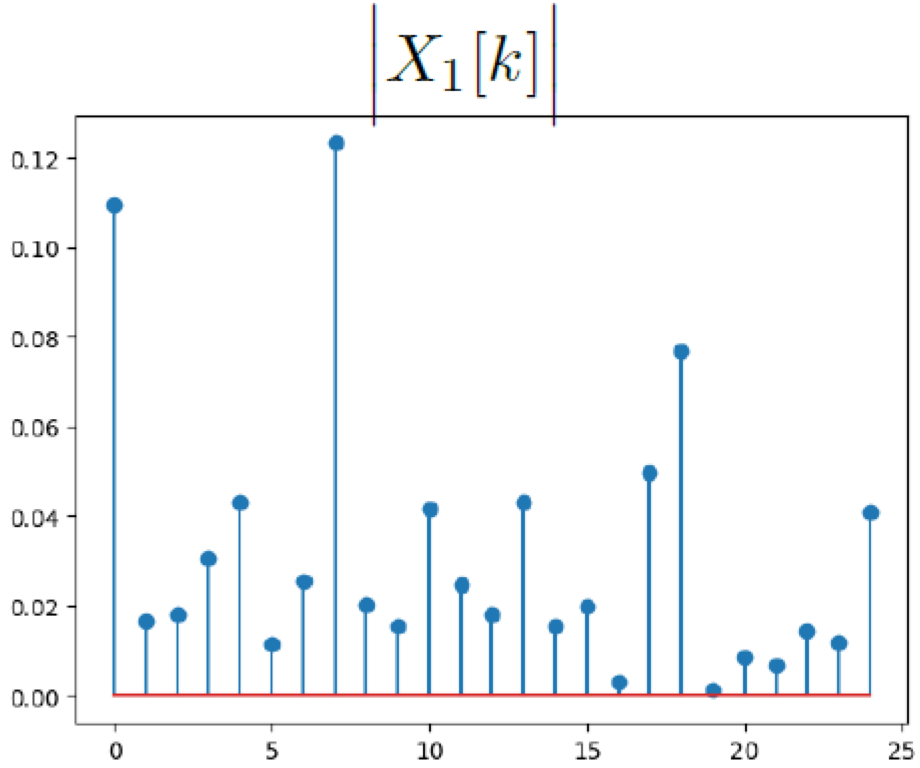


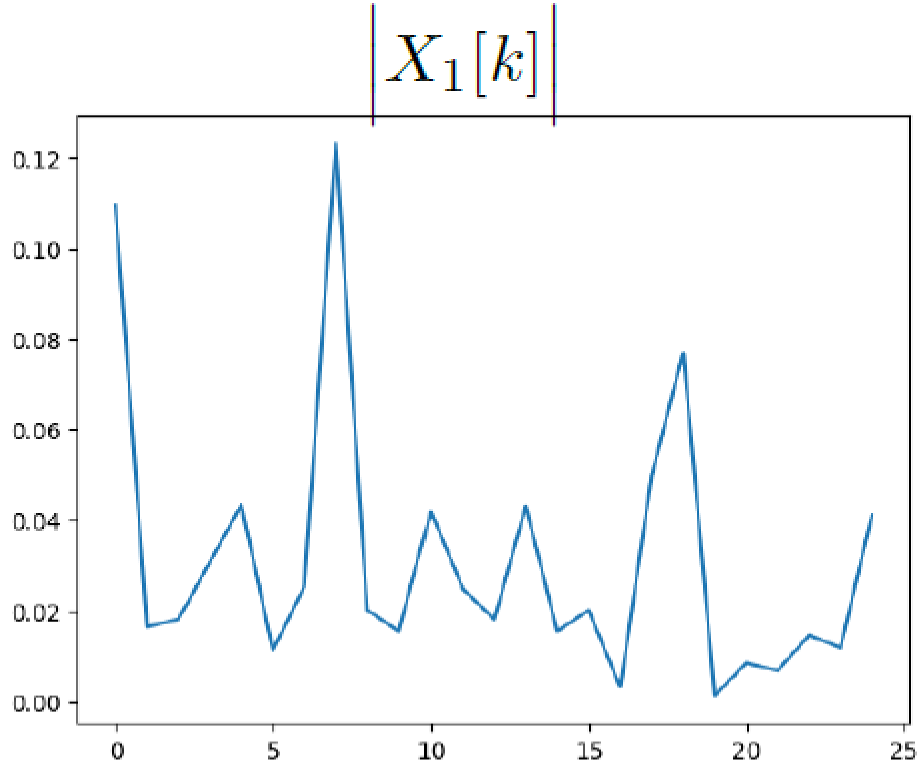
Zero Padding

The stem plots can be distracting when they are close together. (They also take a long time to compute!) Replot using lines (but remember that the signals are DT).

Zero Padding

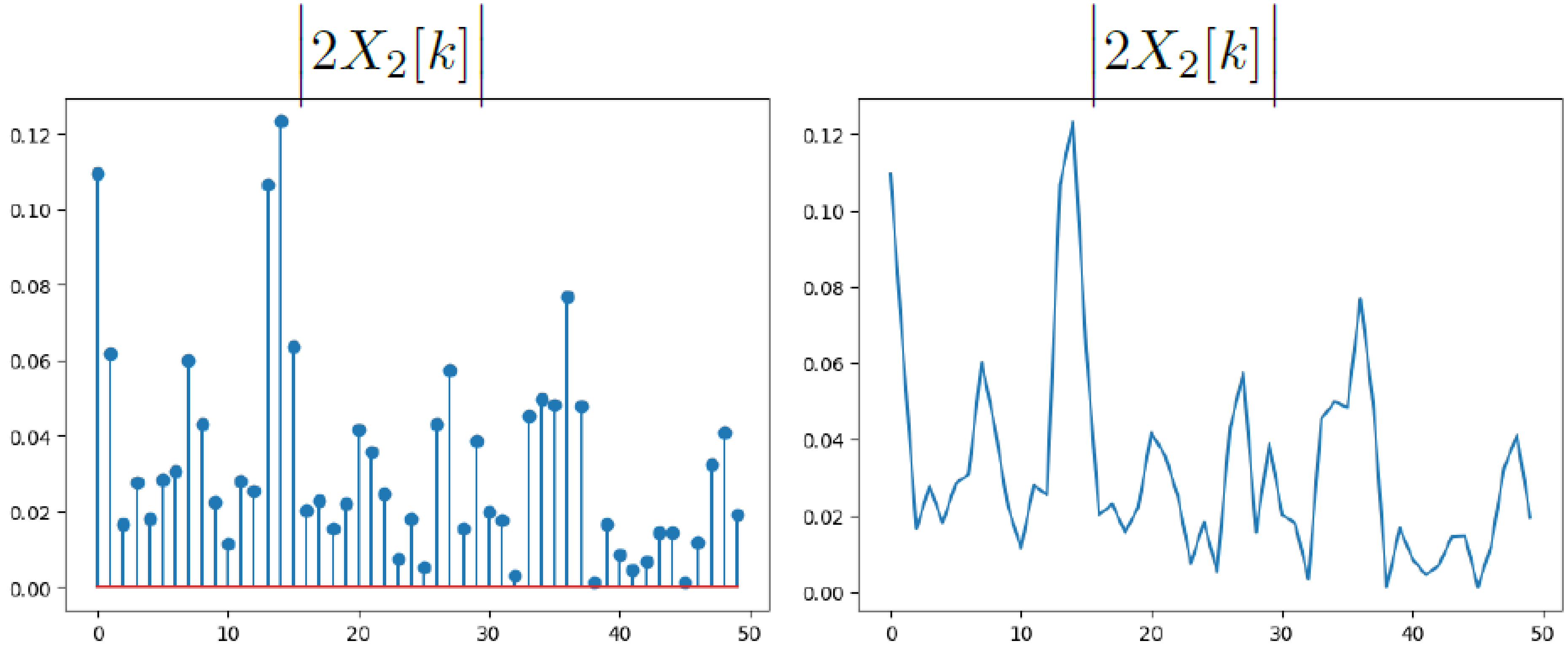
Original (N=1024).





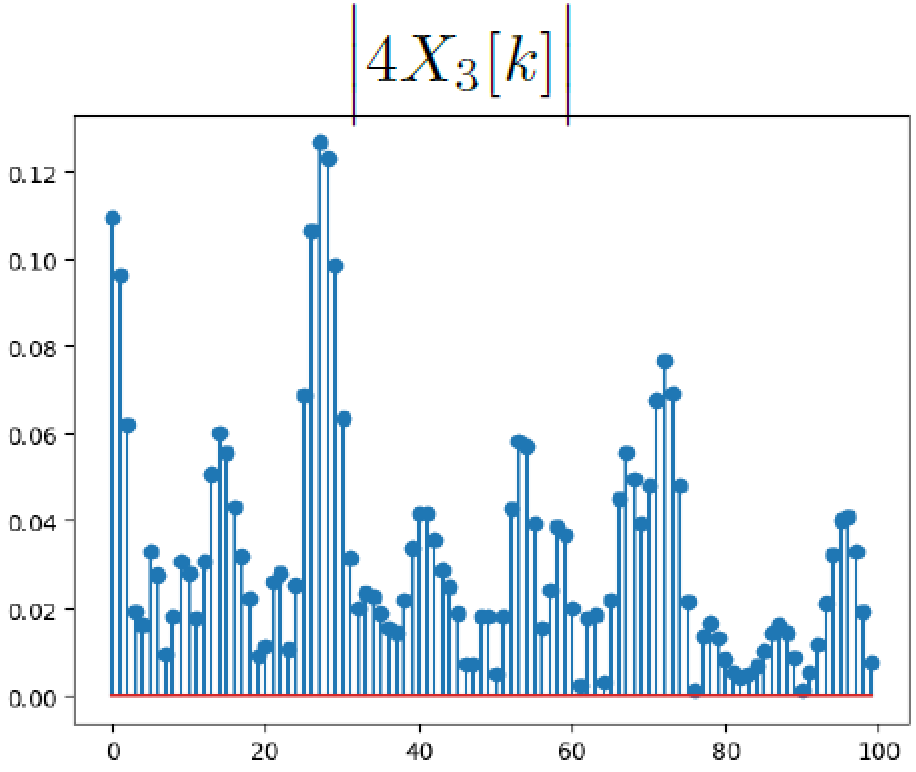
Zero Padding

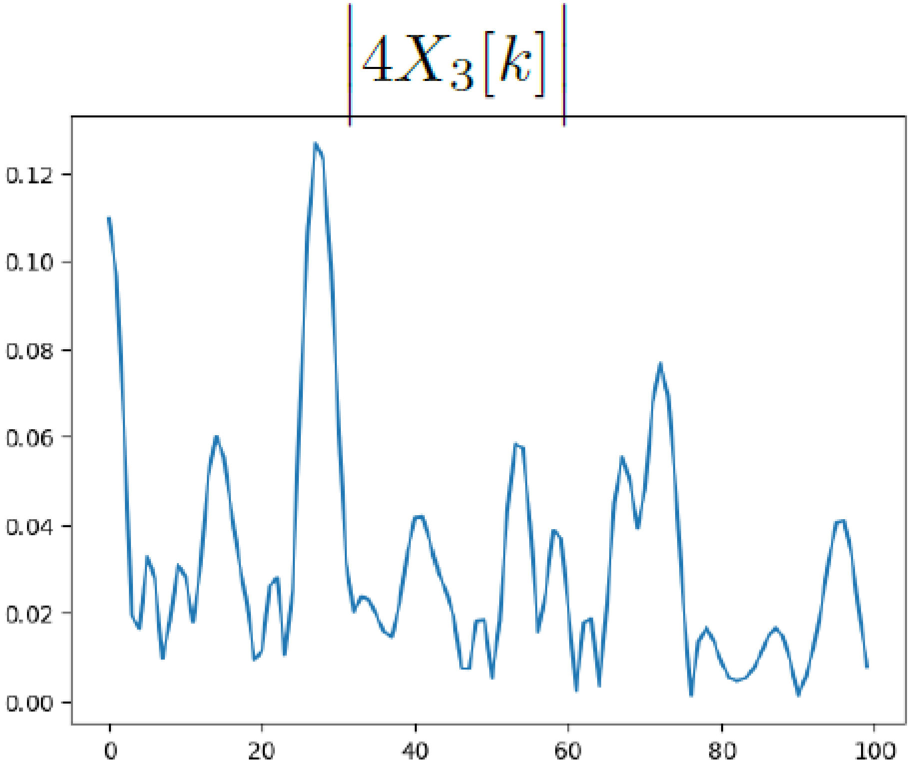
Lengthen by a factor of 2 (N=2048).



Zero Padding

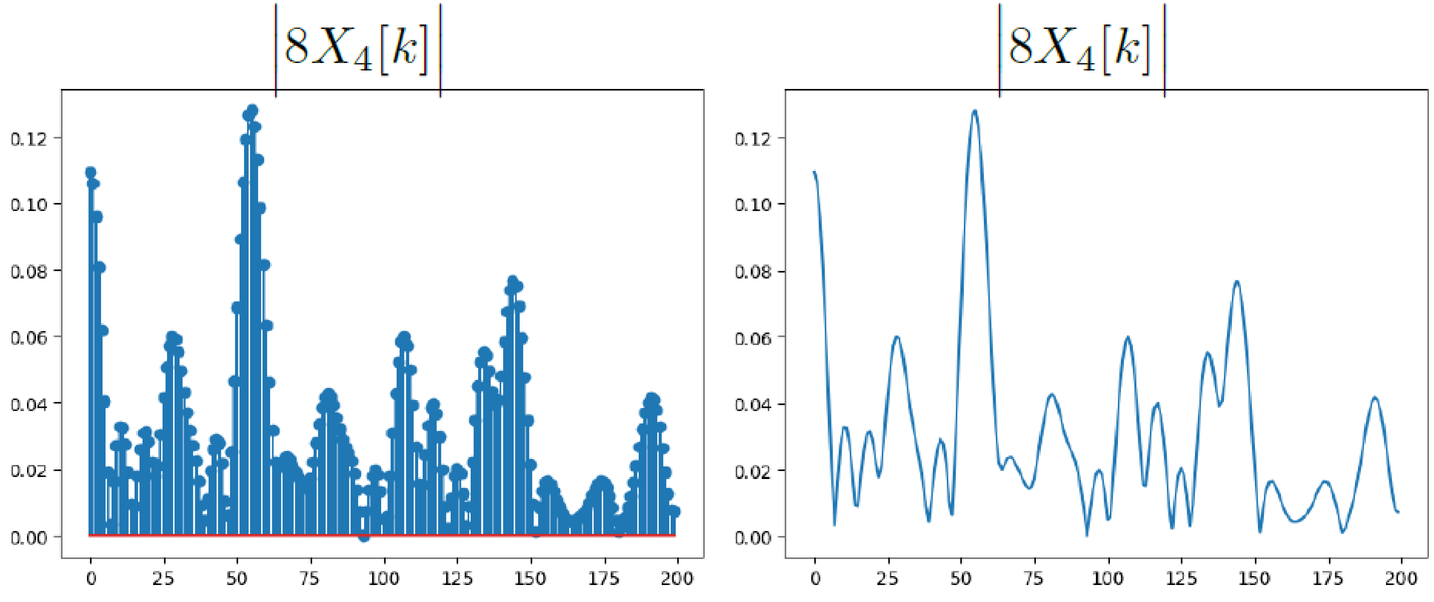
Lengthen by a factor of 4 (N=4096).





Zero Padding

Lengthen by a factor of 8 (N=8192).



Peak is now at $k = 55$.

$$f_o = \frac{k_o}{N} f_s = \frac{55}{8 \times 1024} 44100 \approx 296 \text{ Hz}$$

compared to our previous estimate of 301.46 Hz.

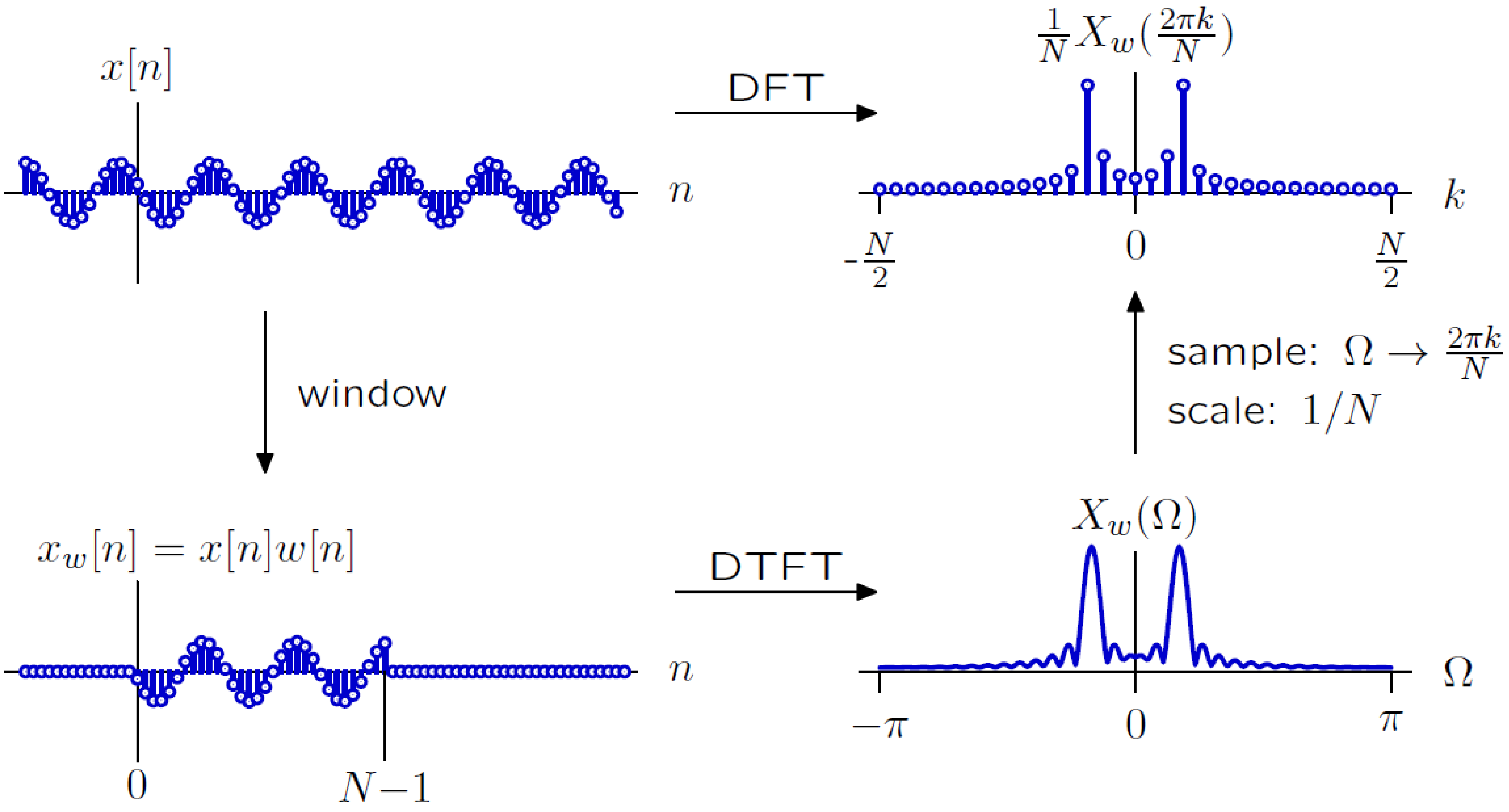
More importantly, frequencies are sampled more densely:

$$\Delta f = \frac{\Delta k}{N} f_s = \frac{1/2}{8 \times 1024} \times 44100 \approx 2.7 \text{ Hz}$$

But we still cannot tell if the note was D or E-flat.

Relation Between DFT and DTFT

Padding with zeros does not increase the length of the "effective" window.
Thus zero padding does not decrease the amount of frequency smearing.



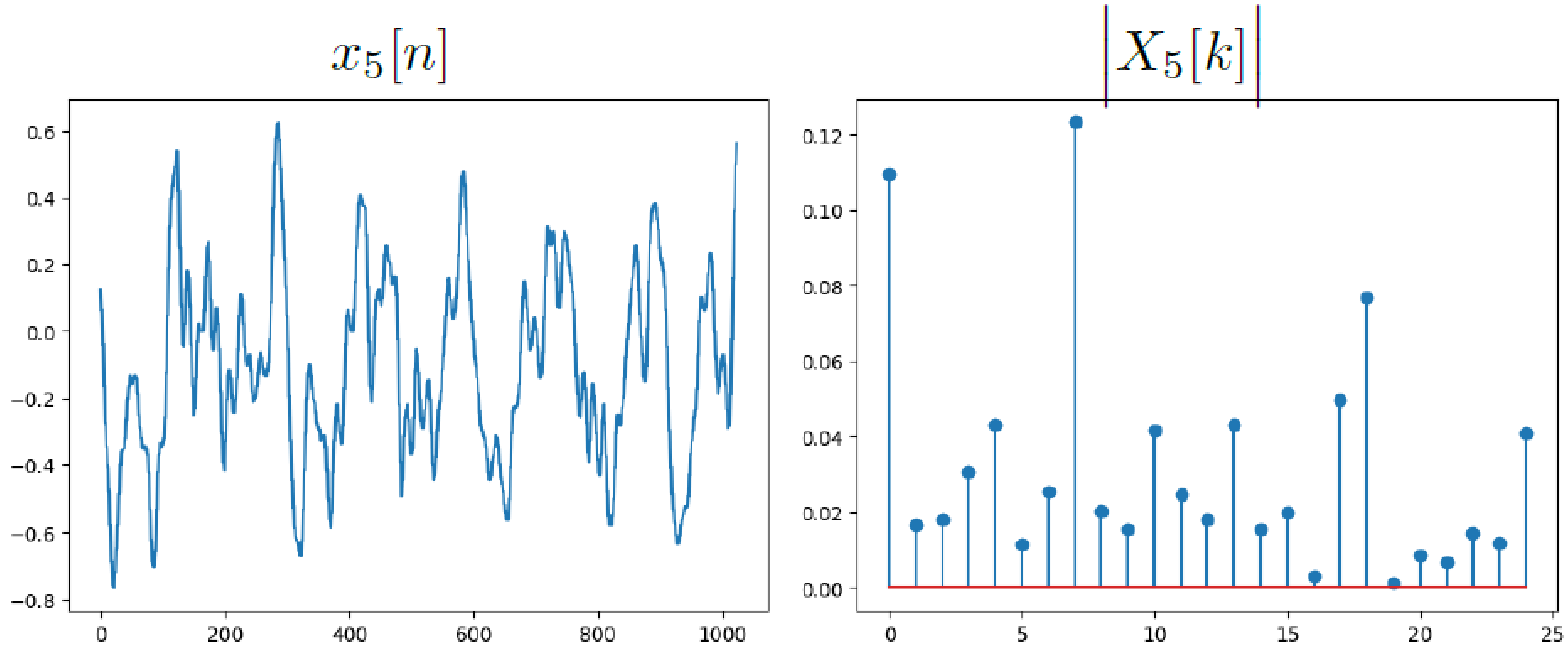
Zero padding adds frequencies but does not sharpen frequency resolution.

More Data

In order to increase frequency resolution, we need to include more data.

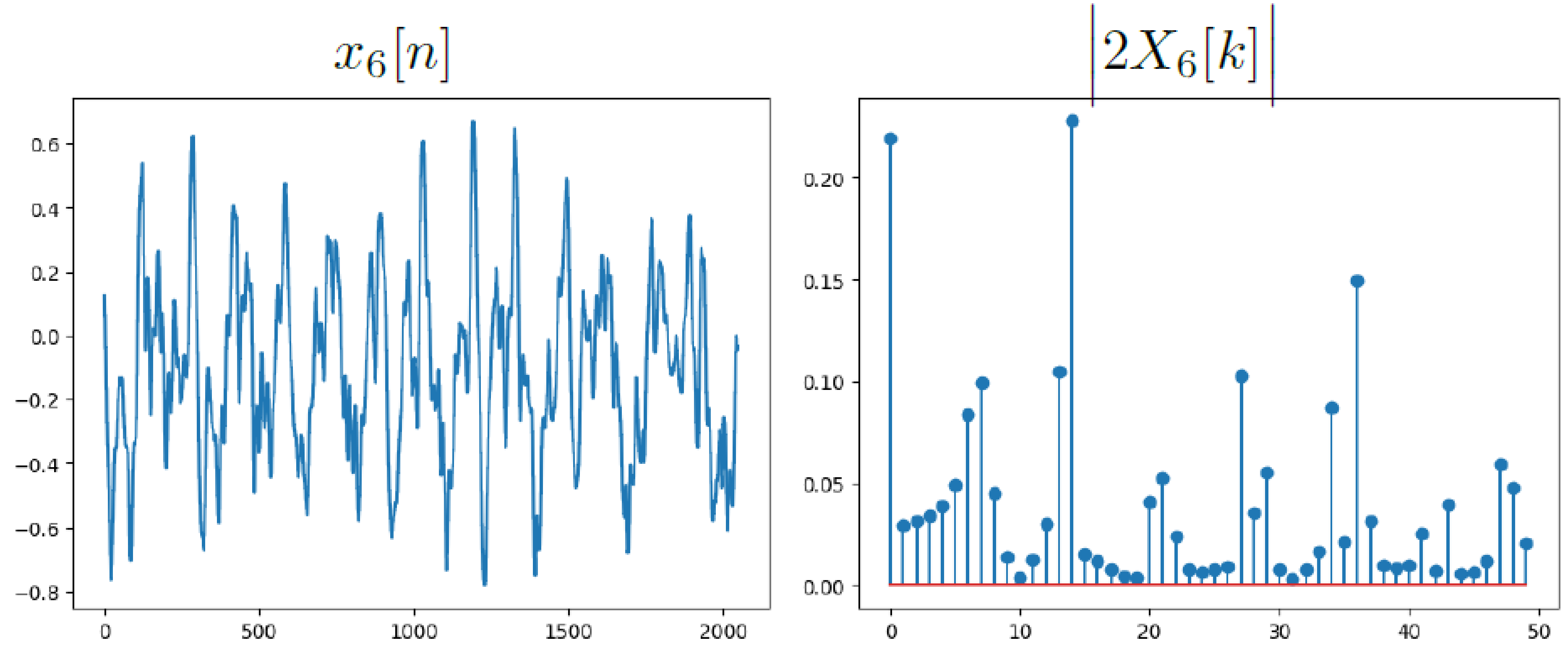
More Data

Original (N=1024).



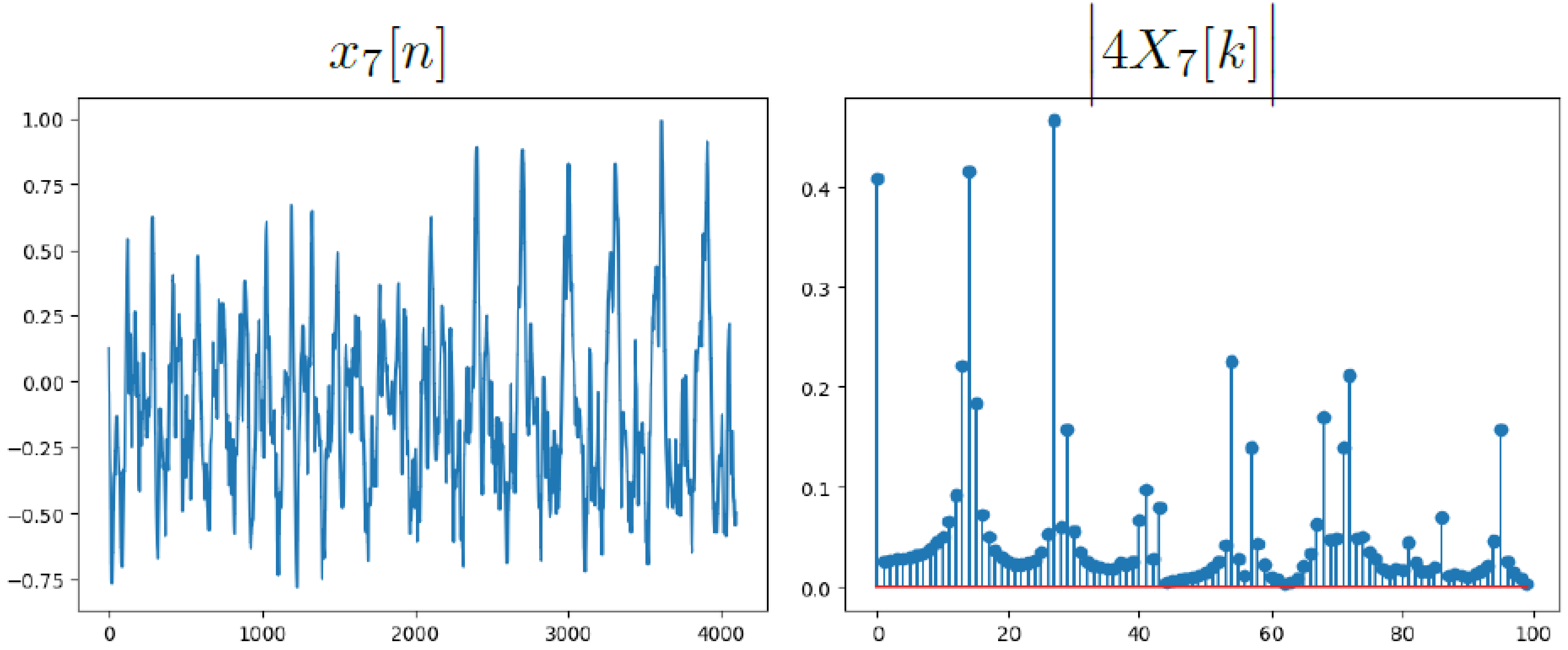
More Data

Lengthen by a factor of 2 (N=2048).



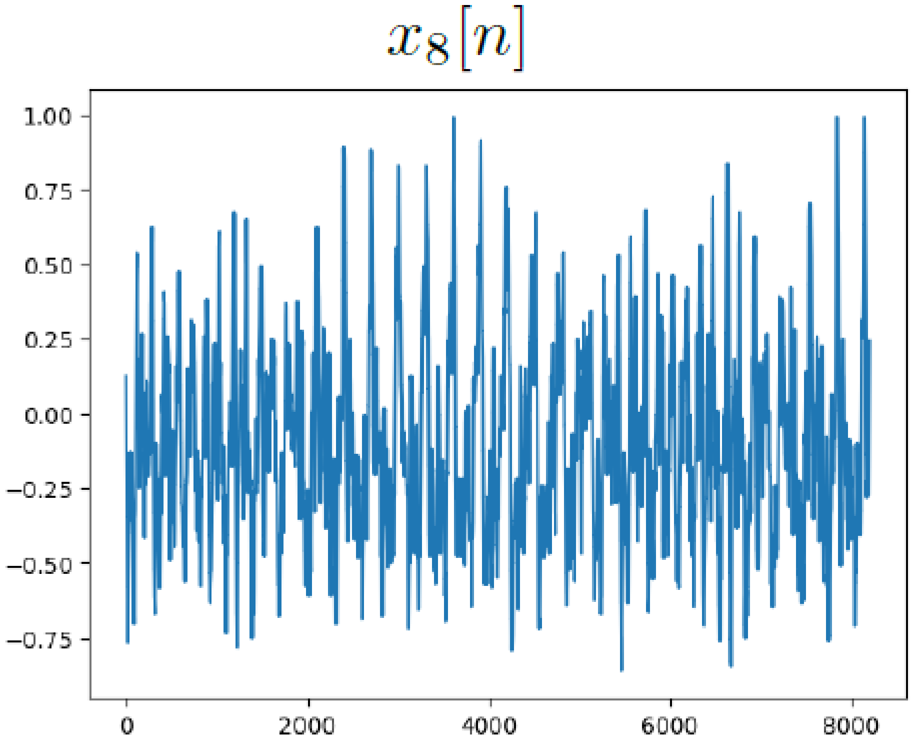
More Data

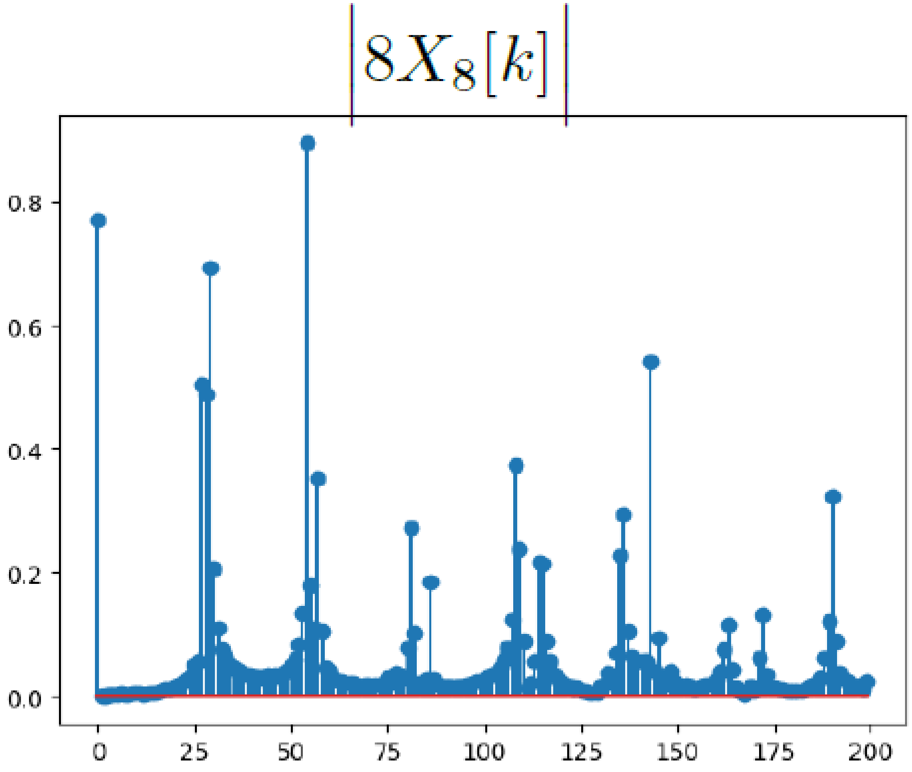
Lengthen by a factor of 4 (N=4096).



More Data

Lengthen by a factor of 8 (N=8192).

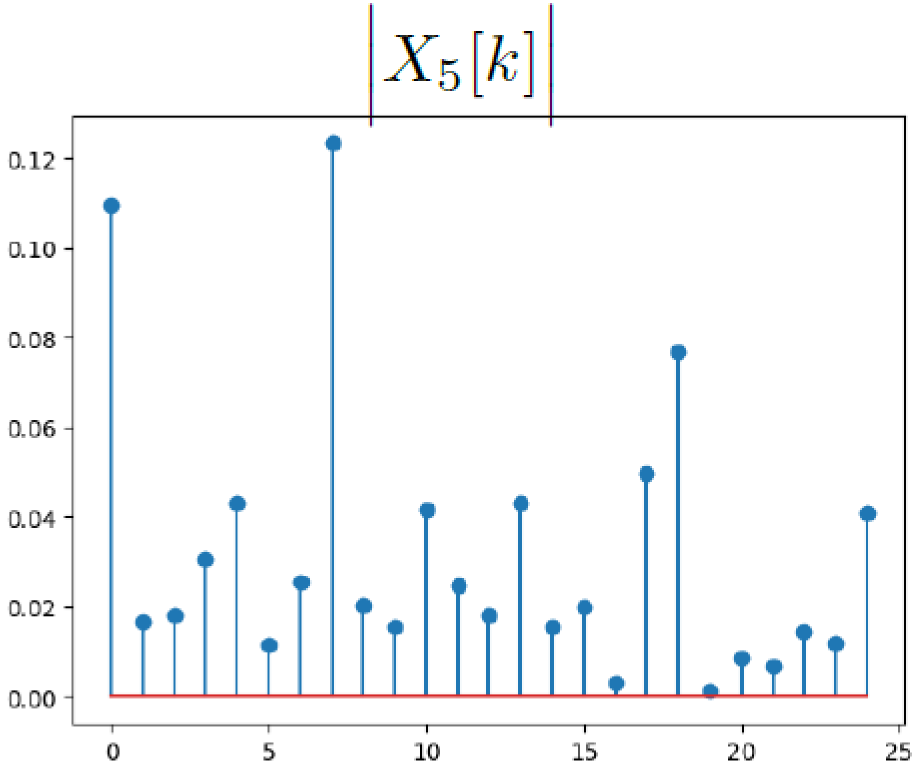


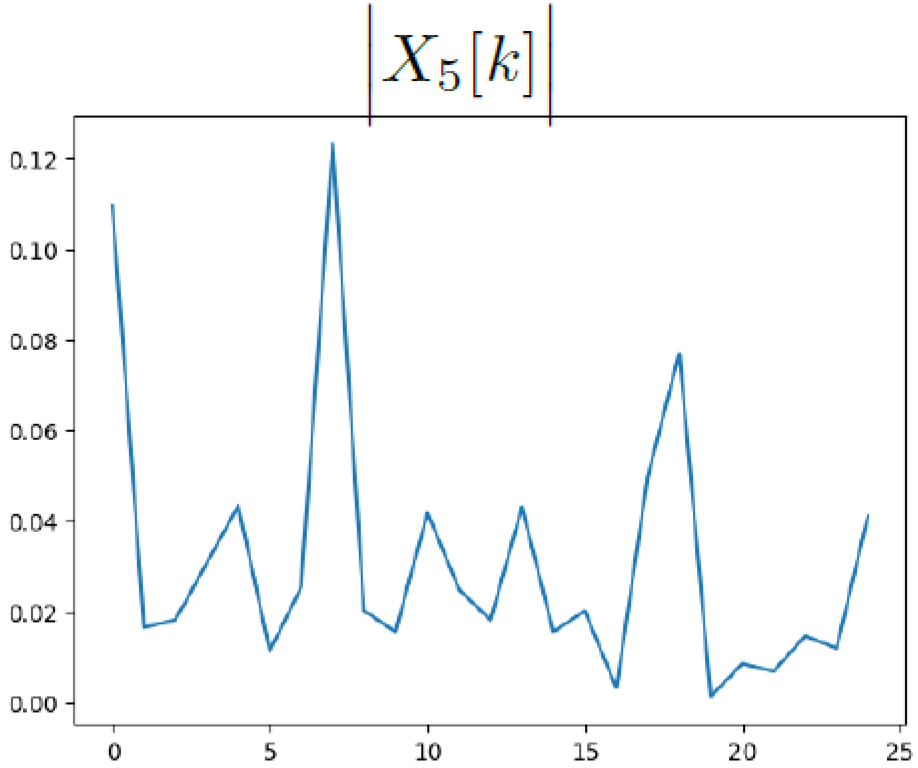


Switching again to line plots ...

More Data

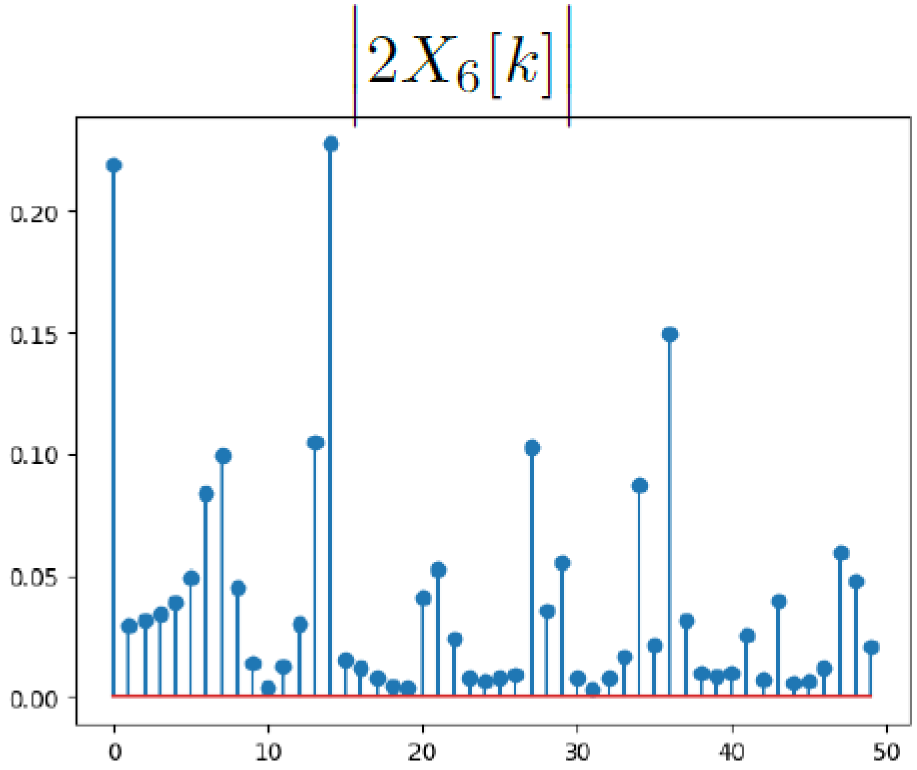
Original (N=1024).

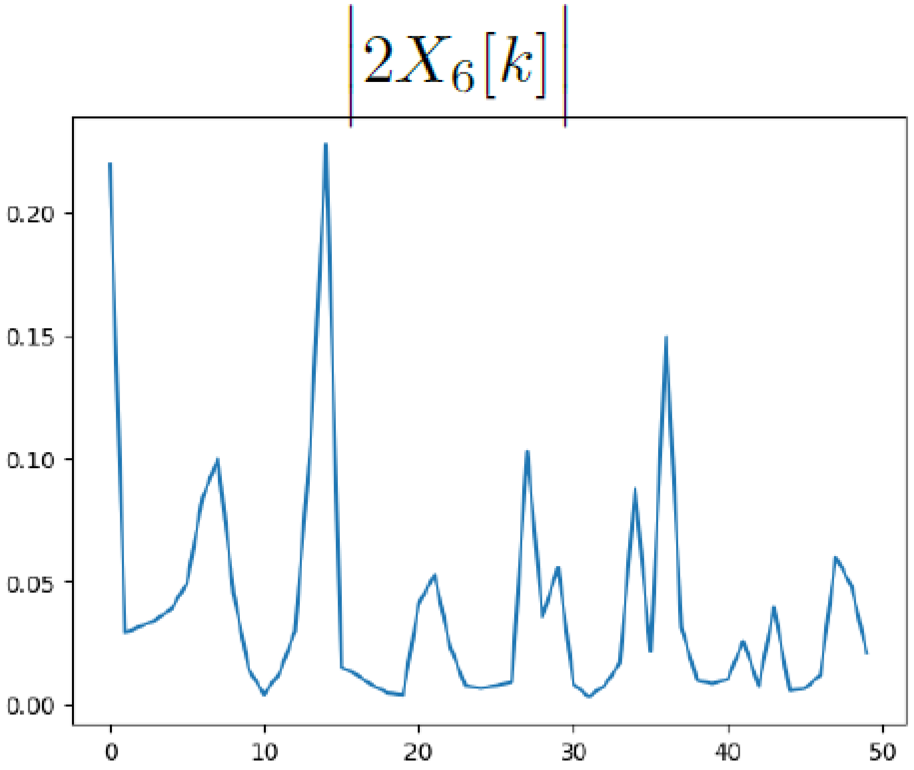




More Data

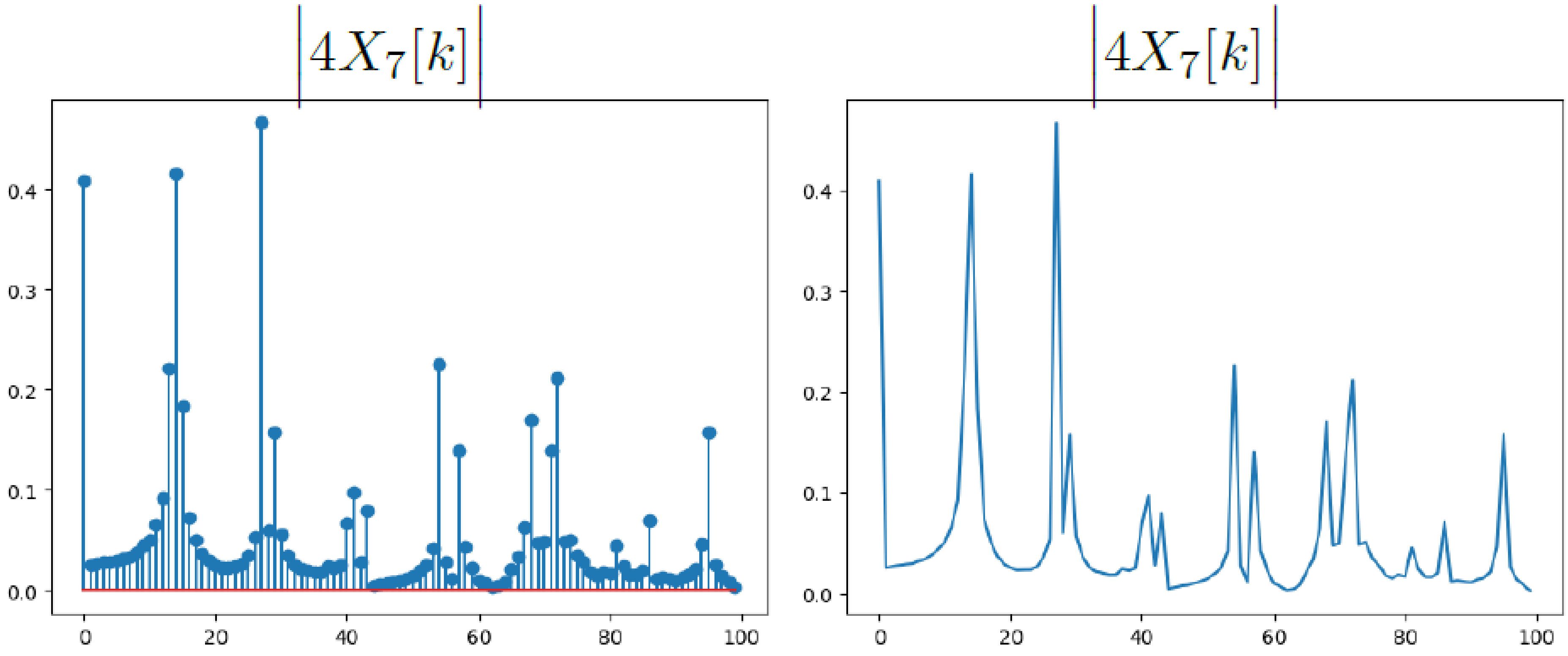
Lengthen by a factor of 2 (N=2048).





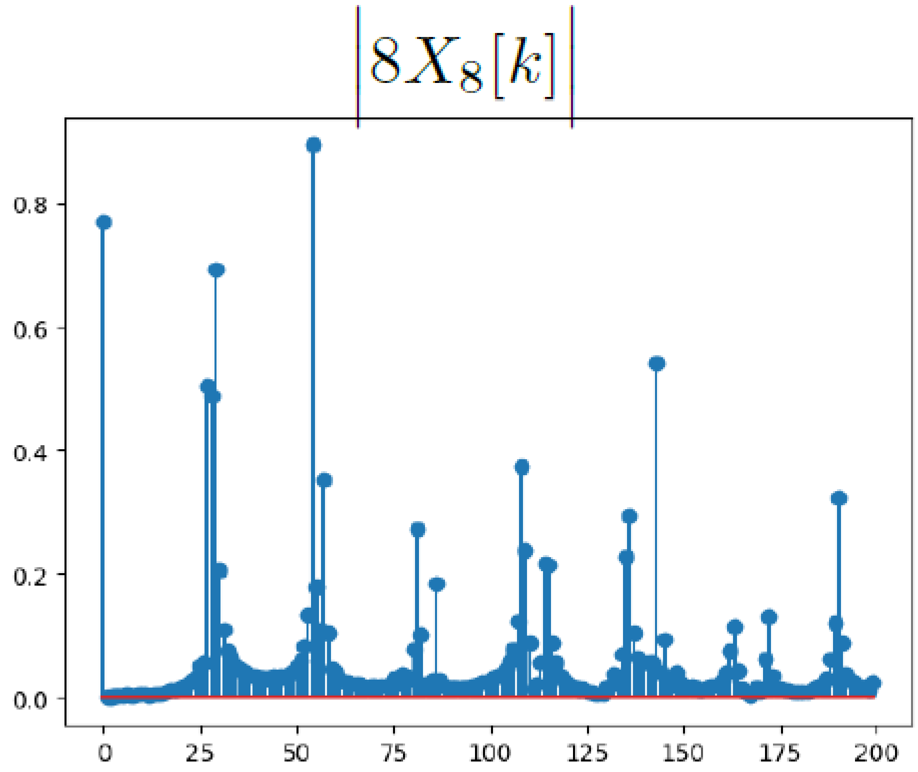
More Data

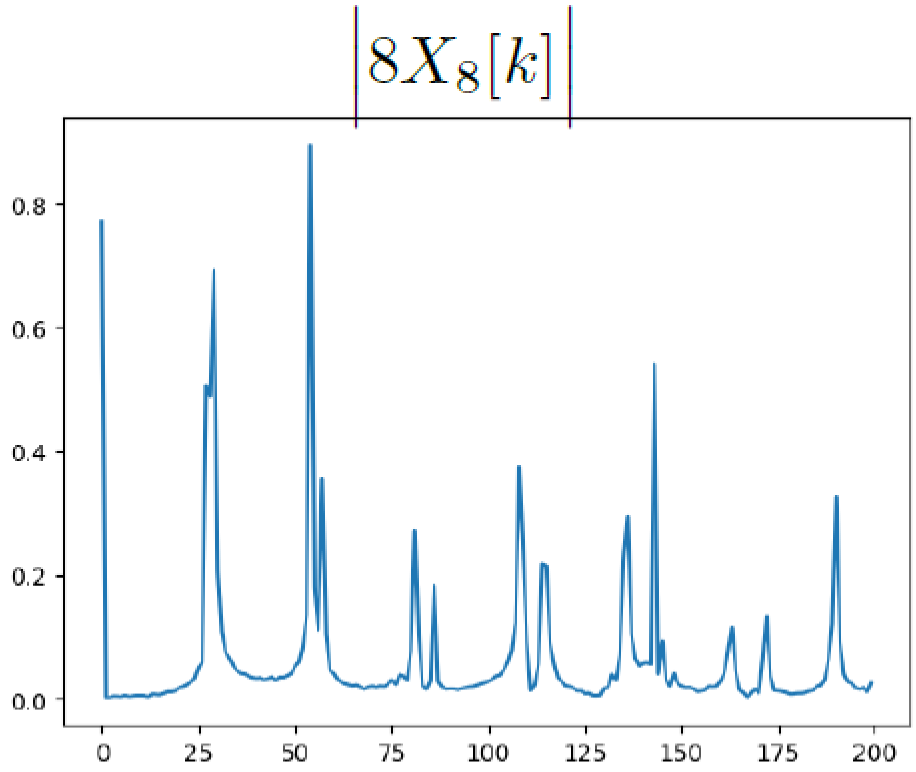
Lengthen by a factor of 4 (N=4096).



More Data

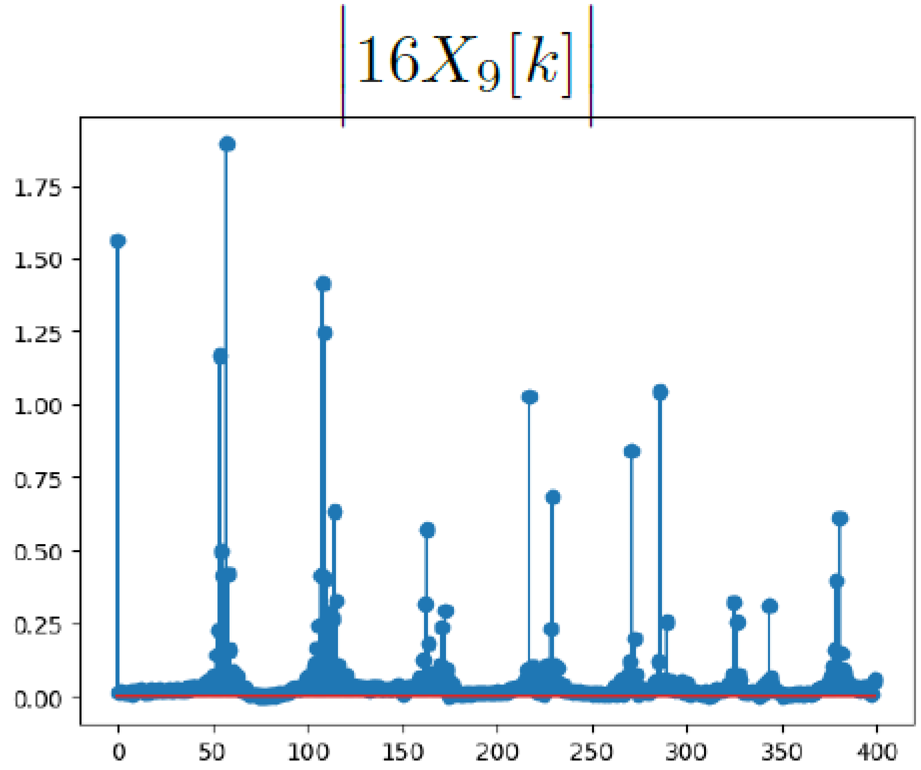
Lengthen by a factor of 8 (N=8192).

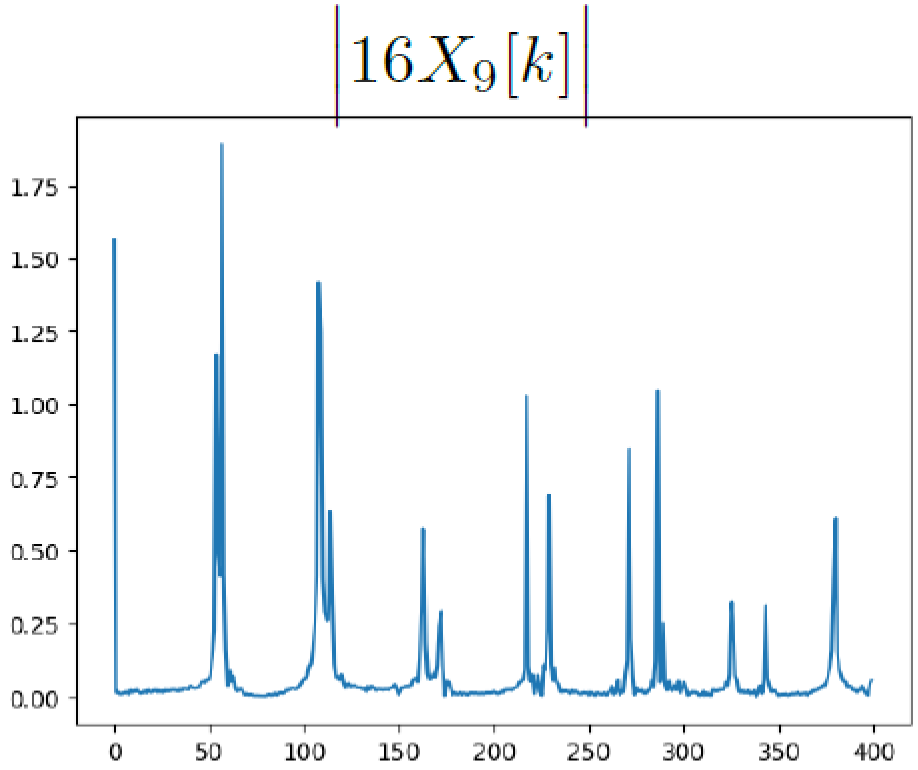




More Data

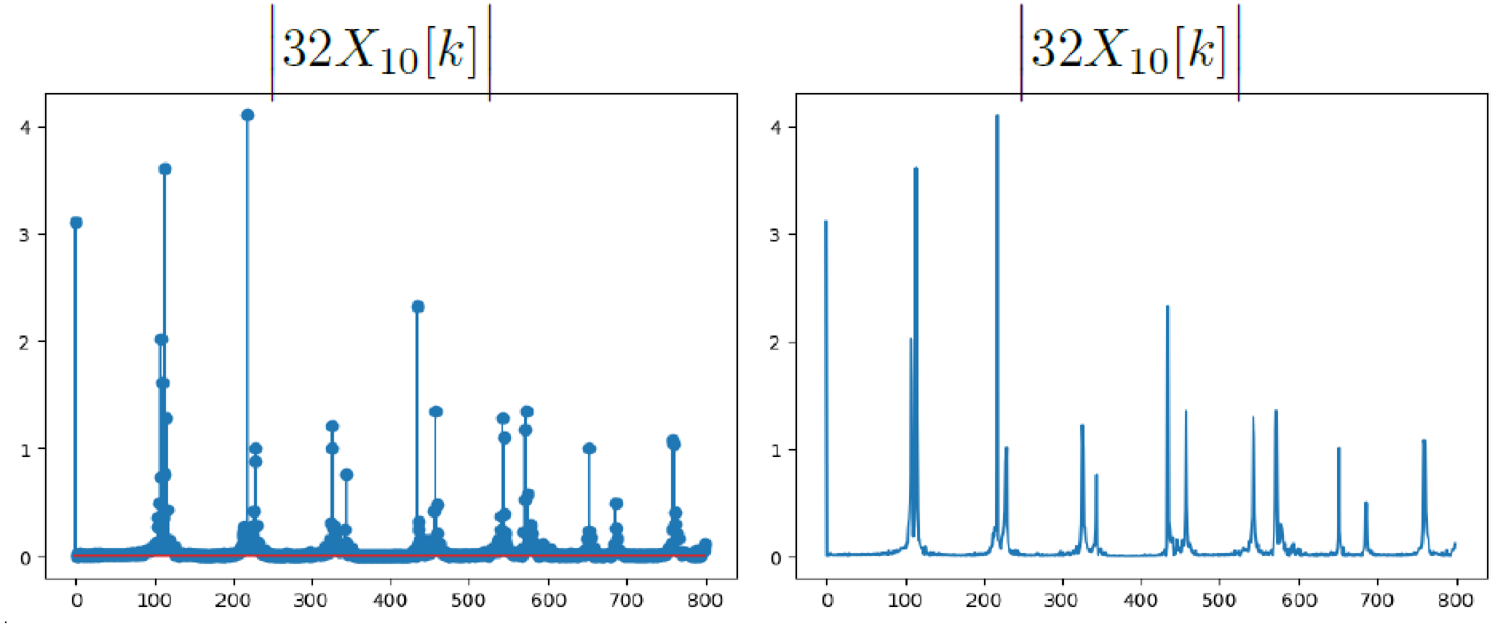
Lengthen by a factor of 16 (N=16,384).





More Data

Lengthen by a factor of 32 (N=32,768).



Clear peaks at $k = 217$ and $k = 228$ ($f = 292.04$ Hz and $f = 306.85$ Hz).
$\rightarrow$ close to D (293.66 Hz) and E-flat (311.13 Hz): both notes are present!

Notice that these are the second harmonics of lower frequencies.
$\rightarrow$ an octave lower than was suggested by the analysis with $N = 1024$.

The fundamental components were not clearly resolved with $N = 1024$ but are clear with $N = 32{,}768$.

Summary: Frequency Resolution

Increasing the length of the analysis by zero padding increases the number of frequency points (because sampling is more dense) but does not increase frequency resolution (because windowing is unchanged).

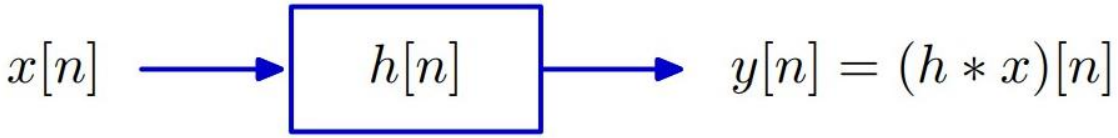
To increase frequency resolution we must increase the number of data that are analyzed.

Implementing Convolution with DFT

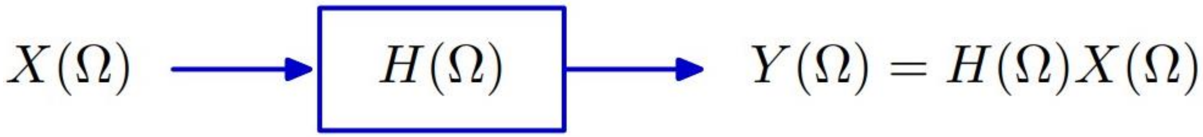
In addition to being useful for characterizing the frequency content of a signal, the DFT can also be used to implement convolution.

Remember we can perform **filtering** in both the time and frequency domains:

Time domain:



Frequency domain:



We can use **DFT** when working in frequency domain!

Regular Convolution

Multiplication in frequency domain correspond to convolution in time domain.

Let $F(\Omega) = F_a\,(\Omega) \cdot F_b\,(\Omega)$, find $f[n]$.

$$f[n] = \frac{1}{2\pi}\int_{2\pi} F(\Omega) \cdot e^{j\Omega n}\, d\Omega \;= \frac{1}{2\pi}\int_{2\pi} F_a\,(\Omega) \cdot F_b\,(\Omega) \cdot e^{j\Omega n}\, d\Omega$$

$$= \frac{1}{2\pi}\int_{2\pi} F_a\,(\Omega) \cdot \left(\sum_{m=-\infty}^{\infty} f_b[m] \cdot e^{-j\Omega m}\right) \cdot e^{j\Omega n}\, d\Omega \;= \sum_{m=-\infty}^{\infty} f_b[m] \frac{1}{2\pi}\int_{2\pi} F_a\,(\Omega) \cdot e^{j\Omega(n-m)}\, d\Omega$$

$$= \sum_{m=-\infty}^{\infty} f_b[m]\, f_a[n-m] \;\equiv (f_b * f_a)[n]$$

$$(x * h)[n] \;\overset{\text{DTFT}}{\Longleftrightarrow}\; H(\Omega)X(\Omega)$$

Implementing Convolution with DFT

Let $F[k] = F_a[k] \cdot F_b[k]$, find $f[n]$.

$$f[n] = \sum_{k=0}^{N-1} F[k] e^{j\frac{2\pi}{N}kn} = \sum_{k=0}^{N-1} F_a[k] \cdot F_b[k] e^{j\frac{2\pi}{N}kn} = \sum_{k=0}^{N-1} F_a[k] \cdot \left(\frac{1}{N} \sum_{m=0}^{N-1} f_b[m] e^{-j\frac{2\pi}{N}km} \right) e^{j\frac{2\pi}{N}kn}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] \left(\sum_{k=0}^{N-1} F_a[k] \cdot e^{j\frac{2\pi}{N}k(n-m)} \right)$$

The expression in the parenthesis looks like $f_a[n-m]$ since

$$f_a[n] = \sum_{k=0}^{N-1} F_a[k] e^{j\frac{2\pi}{N}kn}$$

But $f_a[n]$ was only defined for $0 \le n < N$, and $n - m$ can fall outside that range.

How should we evaluate $f_a[n]$ when n is not between 0 and N−1?

Implementing Convolution with DFT

Let $F[k] = F_a[k] \cdot F_b[k]$, find $f[n]$.

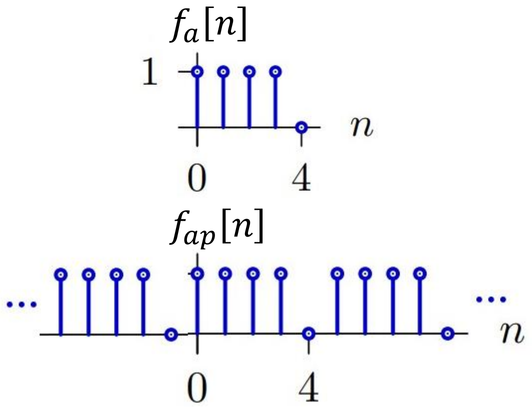
$$f[n] = \sum_{k=0}^{N-1} F[k] e^{j\frac{2\pi}{N}kn} = \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] \left(\sum_{k=0}^{N-1} F_a[k] \cdot e^{j\frac{2\pi}{N}k(n-m)} \right) = \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] \, f_{ap}[(n-m)] = \frac{1}{N} (f_b * f_{ap})[n]$$

The expression in the parenthesis looks like $f_a[n-m]$, but $f_a[n]$ was only defined for $0 \leq n < N$, and $n-m$ can fall outside that range. How should we evaluate $f_a[n]$ when n is not between 0 and N−1?

What is in the parenthesis is inverse DFT, remember iDFT gives the periodically extended signal (slide #3 of today's lecture).

So the expression in the parenthesis equal to $f_{ap}[n-m]$, where $f_{ap}[n]$ is a periodically extended version of $f_a[n]$:

$$f_{ap}[n] = \sum_{i=-\infty}^{\infty} f_a[n+iN] = f_a[n \bmod N]$$

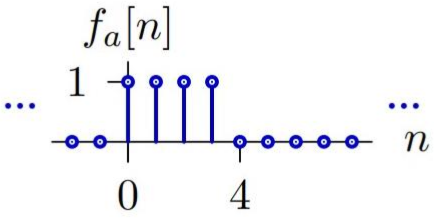


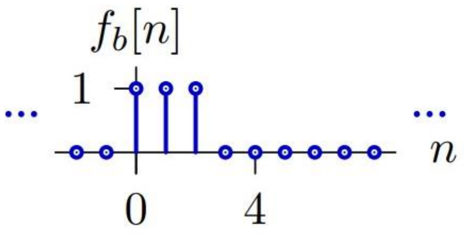
Circular Convolution:

$$f[n] = \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] \, f_a[(n-m) \bmod N] \equiv \frac{1}{N} (f_b \circledast f_a)[n]$$

Circular Convolution

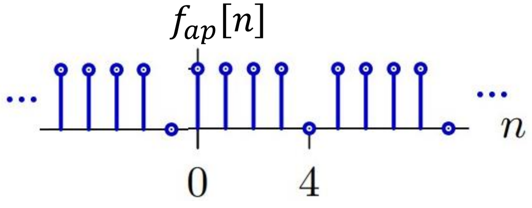
Let us use an example to illustrate the difference between **circular convolution** & **conventional convolution**.





$$(f_b \circledast f_a)[n] = \sum_{m=0}^{N-1} f_b[m]\, f_a[(n-m)\ mod\ N]$$
$$= f_b[0]f_a[n\ mod\ N] + f_b[1]f_a[(n-1)\ mod\ N]$$
$$+ f_b[2]f_a[(n-2)\ mod\ N\ ] = (f_b * f_{ap})[n]$$

$$(f_b * f_a)[n] = \sum_{m=-\infty}^{\infty} f_b[m]\, f_a[n-m] = f_b[0]f_a[n] + f_b[1]f_a[n-1] + f_b[2]f_a[n-2]$$



$(f_b * f_a)$

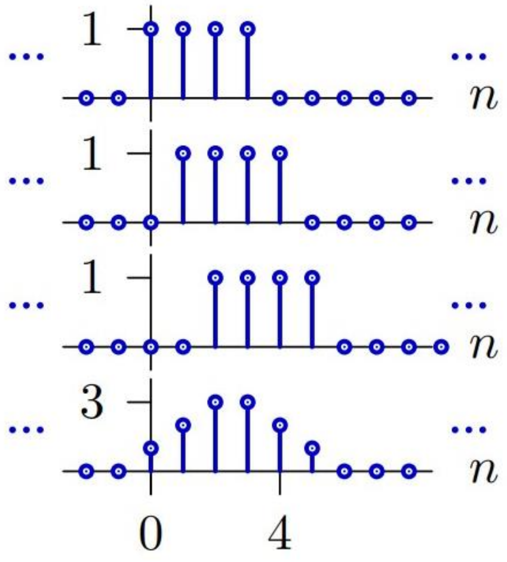
$(f_b \circledast f_a)$

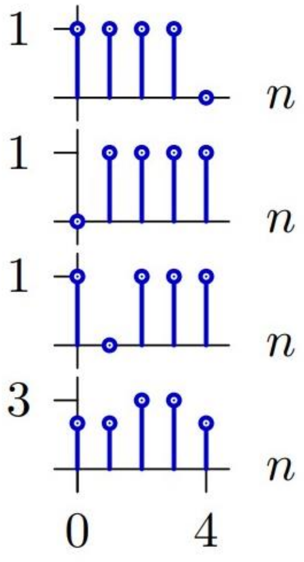
$f_b[0]f_a[n]$

$f_b[1]f_a[n-1]$

$f_b[2]f_a[n-2]$

$(f_b * f_a)[n]$





$f_b[0]f_a[n\ mod\ N]$

$f_b[1]f_a[(n-1)\ mod\ N]$

$f_b[2]f_a[(n-2)\ mod\ N\ ]$

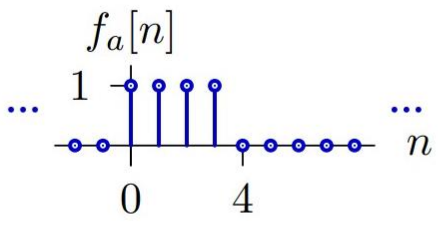
$(f_b \circledast f_a)[n]$

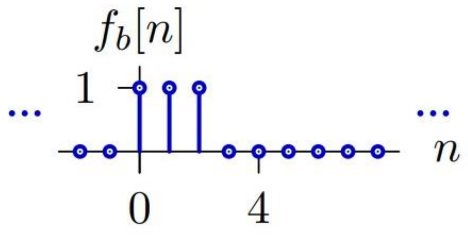
The parts of the conventional convolution that would fall outside the DFT window "alias" to points inside the DFT window.

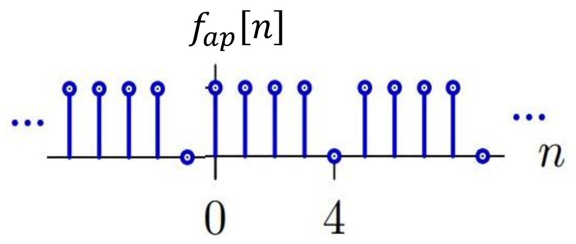
aliasing in time!

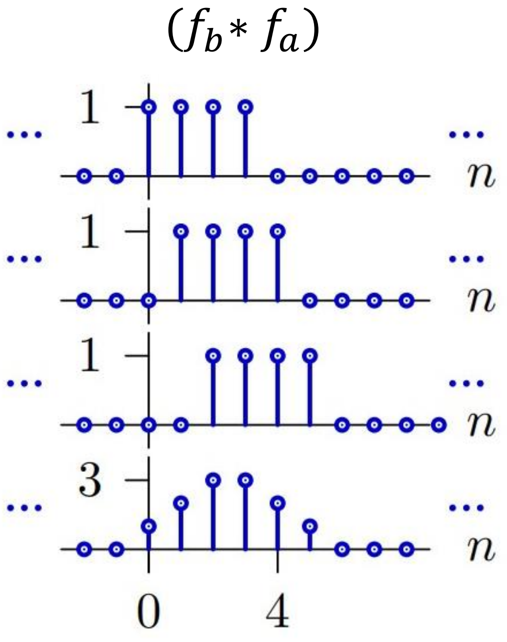
Different Ways to Consider Circular Convolution

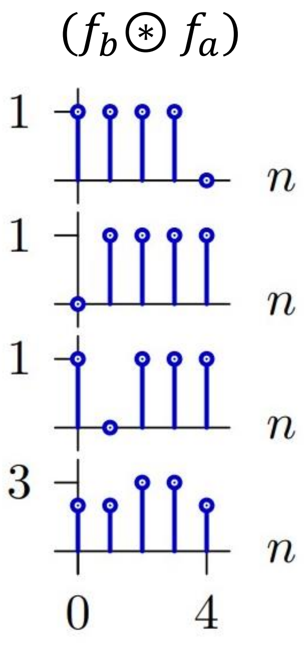
$$(f_b \circledast f_a)[n] = \sum_{m=-\infty}^{\infty} f_b[m]\, f_a[(n-m) \bmod N] = \sum_{m=-\infty}^{\infty} f_b[m]\, f_{ap}[n-m] = (f_b * f_{ap})[n]$$











The result of circular convolution is equivalent to:

- A convolution of one of the signals with a periodically-extended version of the other:

$$(f_b \circledast f_a)[n] = (f_b * f_{ap})[n],\ f_{ap}[n] = \sum_{m=-\infty}^{\infty} f_a[n - mN]$$

- A periodically-extended version (periodic in N) of the result of convolving the two signals:

$$(f_b \circledast f_a)[n] = \sum_{m=-\infty}^{\infty} (f_b * f_a)[n - mN]$$

Implementing Convolution

With DTFT:

If

$$x[n] \overset{DTFT}{\Longleftrightarrow} X(\Omega)$$

and

$$h[n] \overset{DTFT}{\Longleftrightarrow} H(\Omega)$$

then

$$(x * h)[n] \overset{DTFT}{\Longleftrightarrow} H(\Omega)X(\Omega)$$

With DFT:

If

$$x[n] \overset{DFT}{\Longrightarrow} X[k]$$

and

$$h[n] \overset{DFT}{\Longrightarrow} H[k]$$

then

$$\frac{1}{N}(x \circledast h)[n] \overset{DFT}{\Longleftrightarrow} H[k]X[k]$$

Summary

Today we discussed two critical issues in using the DFT.

- Frequency resolution - how the length of a signal determines the ability to discriminate frequencies using the DFT.
- Circular Convolution - how the DFT can be used to carry out time domain operations.