6.3000: Signal Processing

Discrete Fourier Transform 2

- Frequency Resolution
- Circular Convolution

April 02, 2026

Last Time

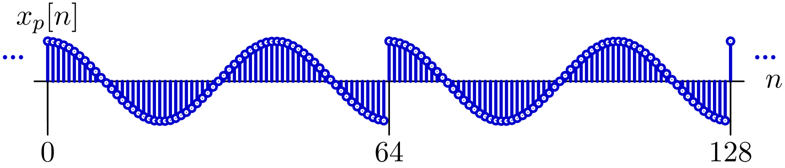
Define the Discrete Fourier Transform (DFT).

Compare the DFT to other Fourier representations.

	analysis	**synthesis**
DTFS:	$X[k] = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] e^{-j\frac{2\pi k}{N}n}$	$x[n] = \sum_{k=\langle N \rangle} X[k] e^{j\frac{2\pi k}{N}n}$
DTFT:	$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}$	$x[n] = \frac{1}{2\pi} \int_{2\pi} X(\Omega) e^{j\Omega n} d\Omega$
DFT:	$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N}n}$	$x[n] = \sum_{n=\langle N \rangle} X[k] e^{j\frac{2\pi k}{N}n}$

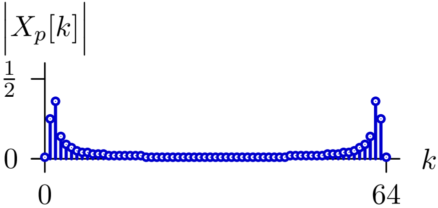
Relation Between DFT and DTFS

The 64-point DFT of $x_2[n] = \cos \frac{3\pi n}{64}$



is equal to the 64-point DTFS of the **periodic extension** of $x_2[n]$.

$$x_p[n] = x_2[n \bmod 64]$$

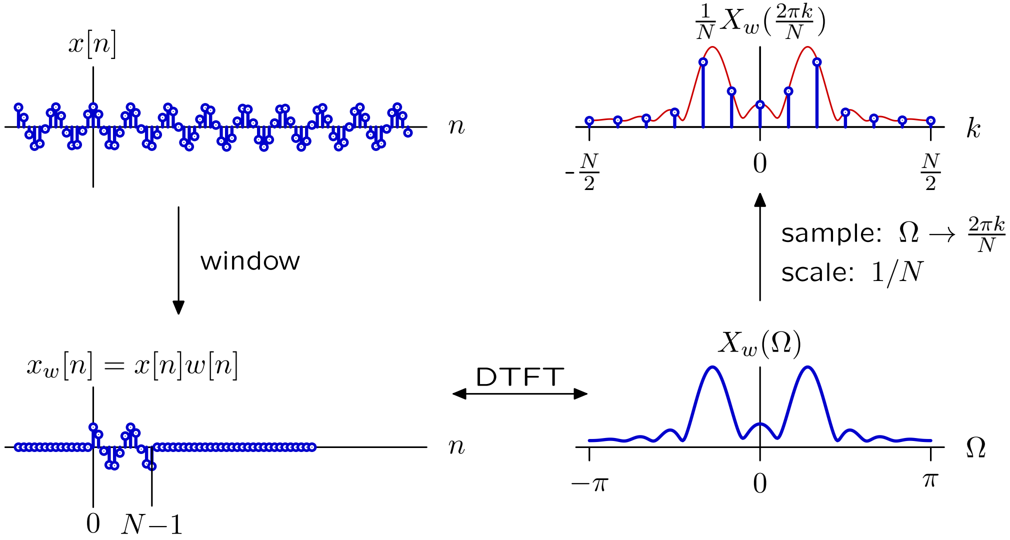


From this perspective, the large number of non-zero frequency components in the DFT of x_2 are needed to generate the step discontinuity at $n = 64$.

Relation Between DFT and DTFT

Decreasing the analysis window N decreases frequency resolution.

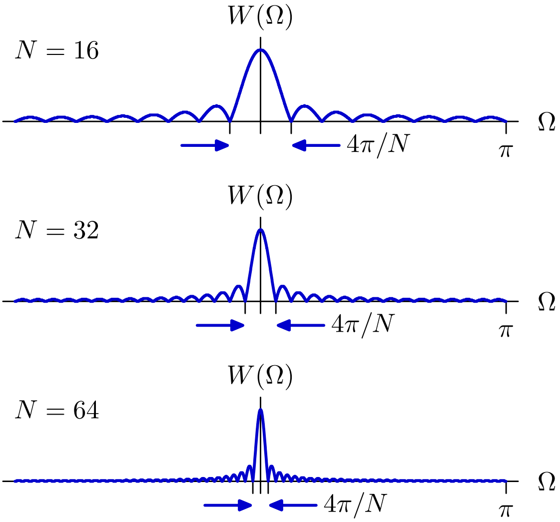
$N = 12$



Frequency Resolution

Frequency blurring is fundamental to the way the DFT works.

Longer windows provide finer frequency resolution.

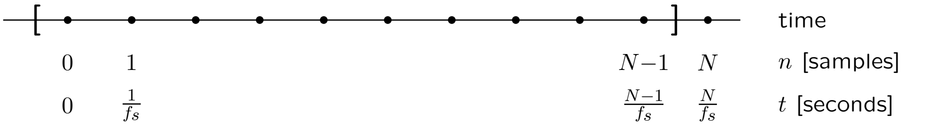


The width of the central lobe is inversely related to window length.

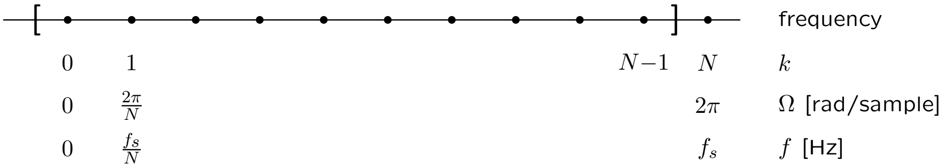
Length of Analysis Window N

The DFT provides a parameter (N) to customize performance.

The time window is divided into N samples numbered $n = 0$ to $N-1$.



Discrete frequencies are similarly numbered as $k = 0$ to $N-1$.

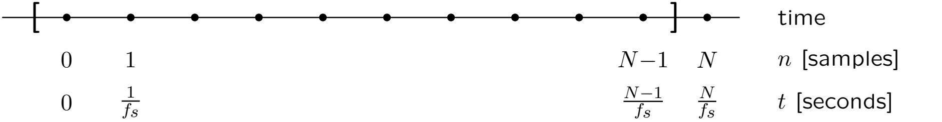


N determines both the length of the window in time and the frequency resolution of the result.

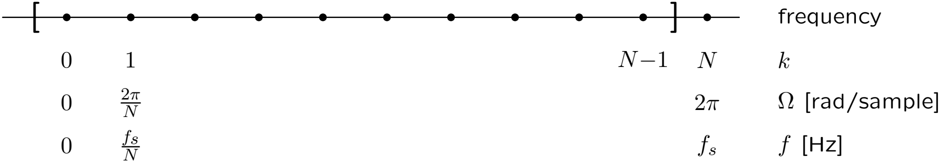
Length of Analysis Window N

The DFT provides a parameter (N) to customize performance.

The time window is divided into N samples numbered $n = 0$ to $N-1$.



Discrete frequencies are similarly numbered as $k = 0$ to $N-1$.



Which is better: big or small values of N?

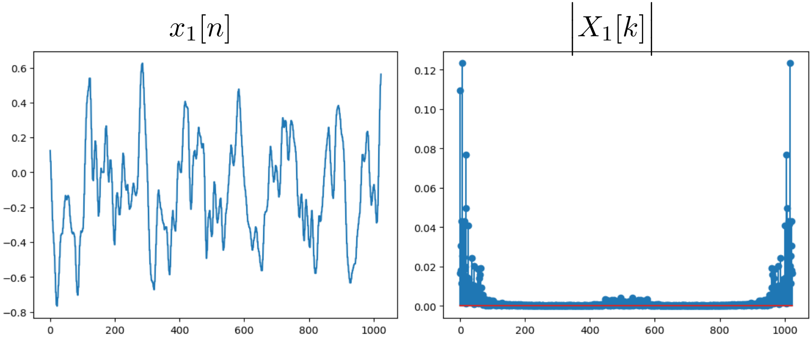
Frequency Analysis

Example: Determine the frequency content of the following sound.

cello: DEb3.wav ($f_s = 44,100\,$Hz)

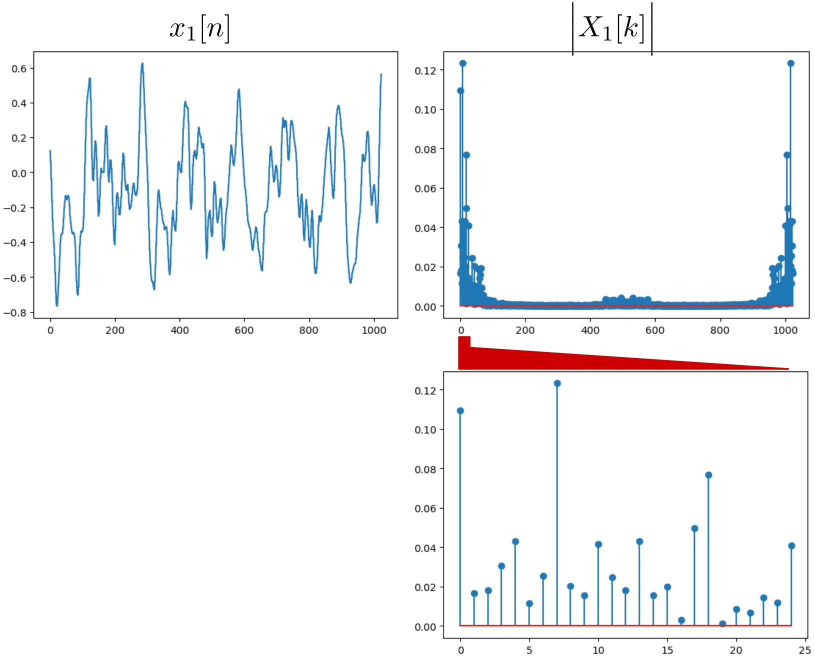
Frequency Analysis

Extract 1024 samples and calculate DFT.



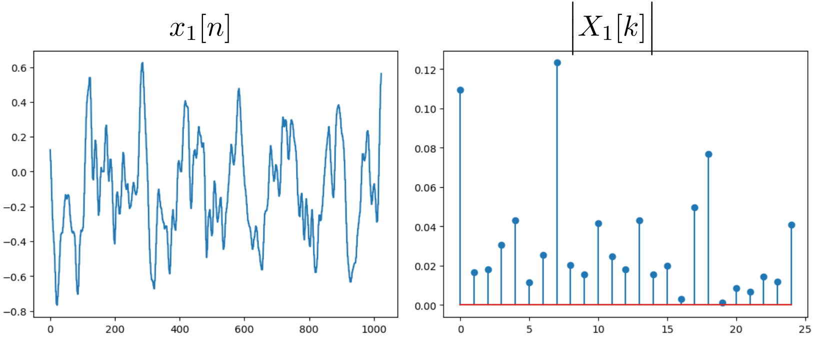
Frequency Analysis

Information about pitch is at low frequencies. Zoom in on $k = 0$ to 24.



Check Yourself

The magnitude of the DFT is largest at $k = 7$ ($f_s = 44100\,\text{Hz}$).

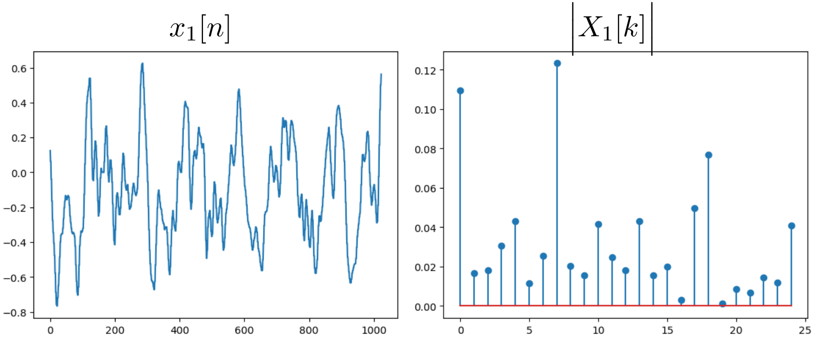


What is the corresponding frequency in Hz?

1. 293.66 Hz
2. 301.46 Hz
3. 146.83 Hz
4. 150.73 Hz
5. None of the above

Frequency Analysis

Information about pitch is at low frequencies. Zoom in on $k = 0$ to 24.



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

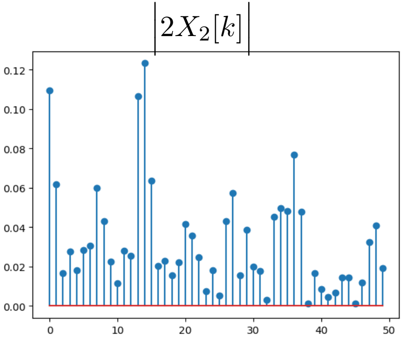
We can increase N to increase the number of analyzed frequencies.

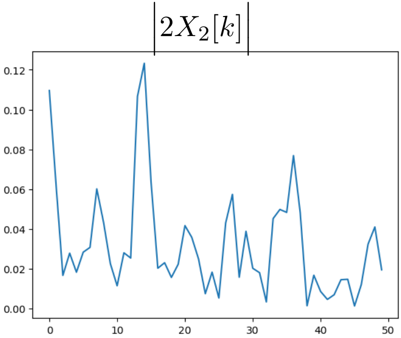
Two methods to increase N:

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

Zero Padding

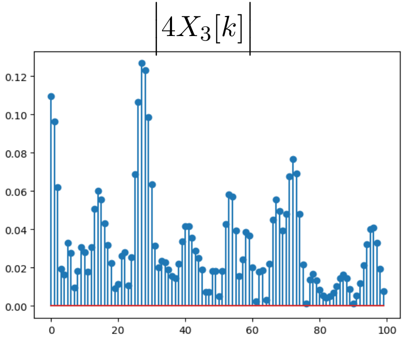
Lengthen by a factor of 2 (N=2048).

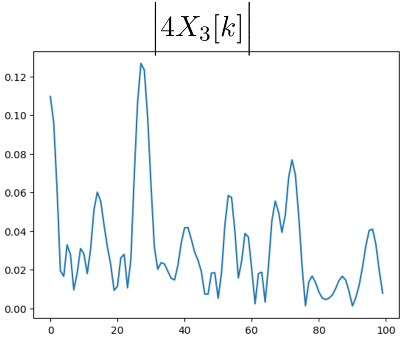




Zero Padding

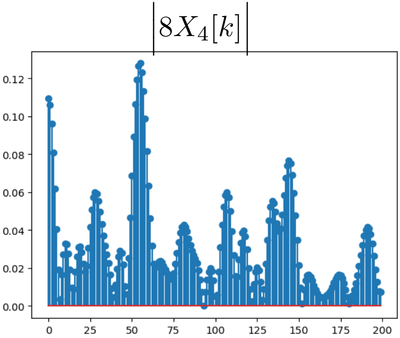
Lengthen by a factor of 4 (N=4096).

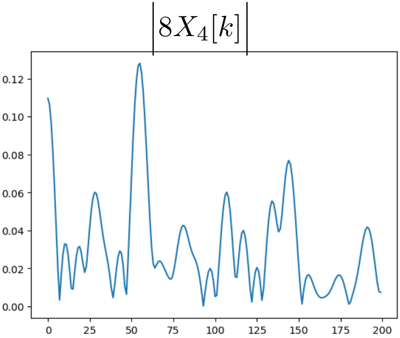




Zero Padding

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





Peak is now at $k = 55$.

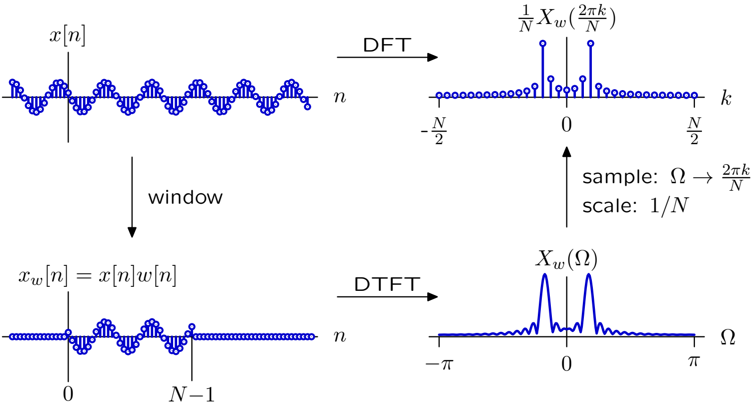
$$f_o = \frac{k_o}{N} f_s = \frac{55}{8 \times 1024} 44100 \approx 296 \text{ Hz}$$

compared to our previous estimate of 301.46 Hz.

But the peak is still between D (293.66 Hz) and E-flat (311.13 Hz).

Can we sharpen the peak so that it falls on D or E-flat but not both?

Check Yourself



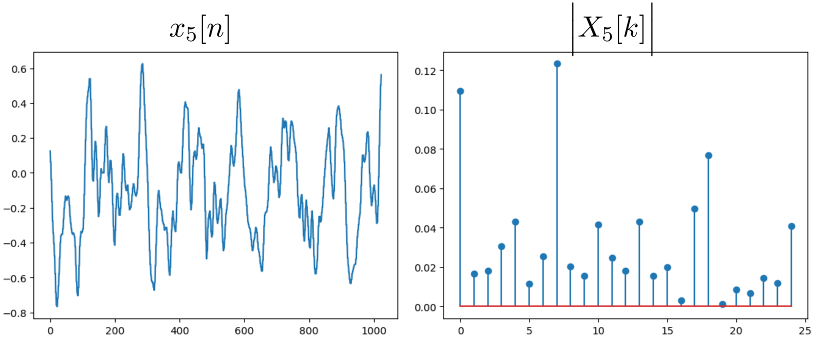
Which of the following is/are true?

1. Zero-padding has no effect on the DTFT of $x_w[n]$.
2. Zero-padding decreases spectral smear in the DTFT.
3. Zero-padding has no effect on the sampled version $X_w(\Omega)$.
4. Zero-padding decreases frequency (Hz) separation of DFT samples.

More Data

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

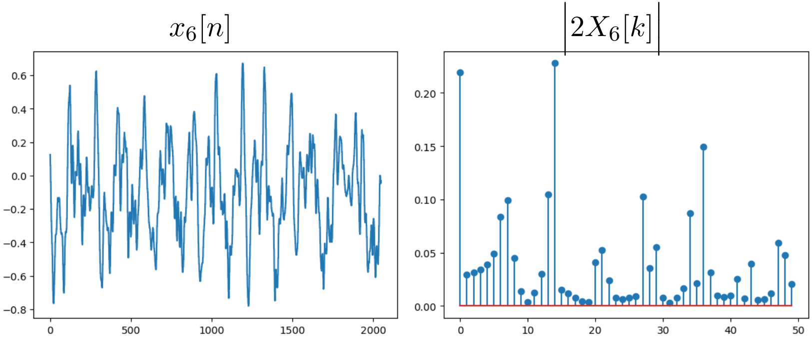
Original (N=1024).



Peak is at $k = 7$ (as before).

More Data

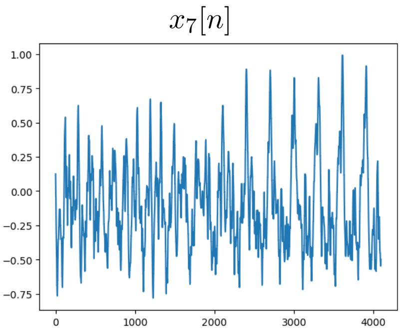
Lengthen by a factor of 2 (N=2048).

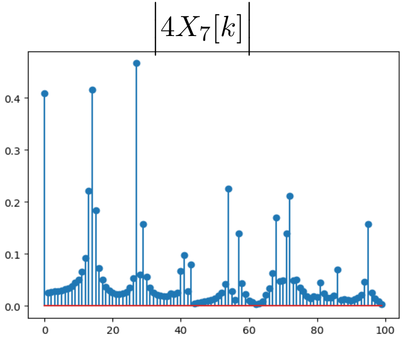


Peak is at $k = 14$.

More Data

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

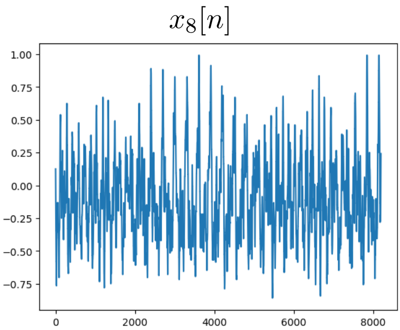


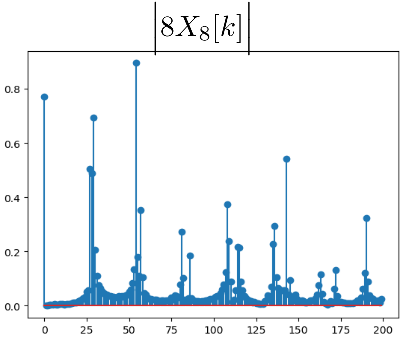


Peak is at $k = 27$.

More Data

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

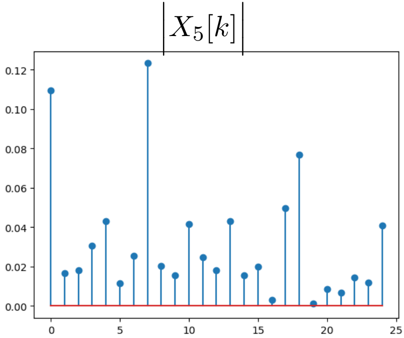


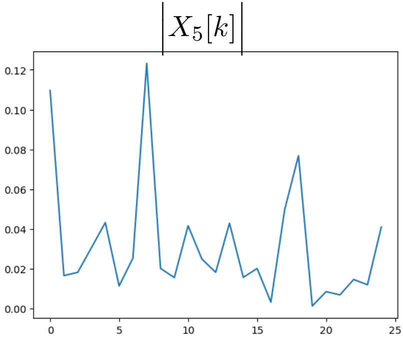


Switching again to line plots ...

More Data

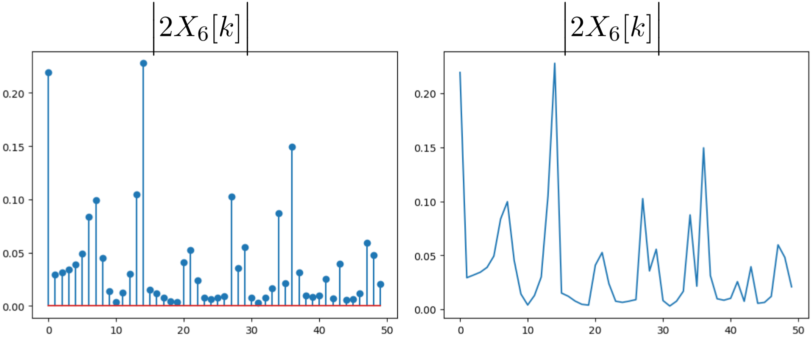
Original (N=1024).





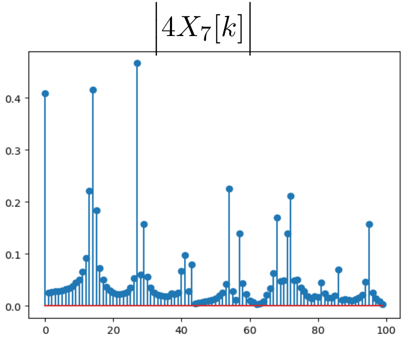
More Data

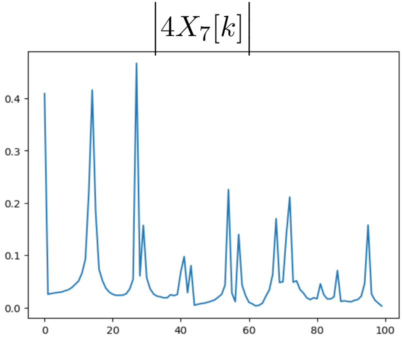
Lengthen by a factor of 2 (N=2048).



More Data

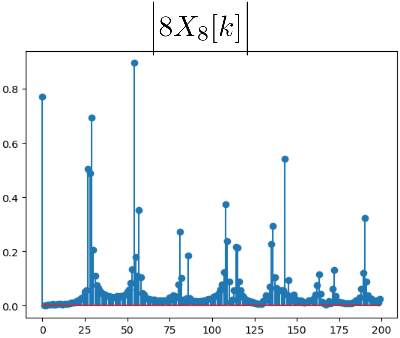
Lengthen by a factor of 4 (N=4096).

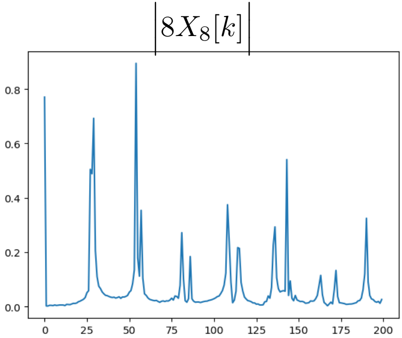




More Data

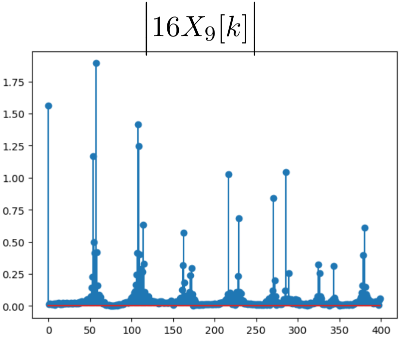
Lengthen by a factor of 8 (N=8192).

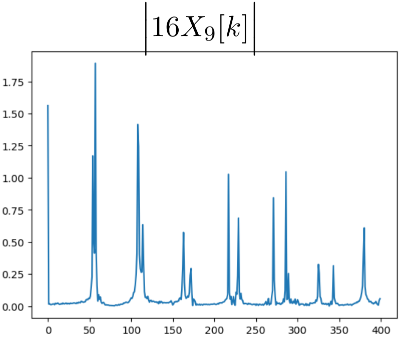




More Data

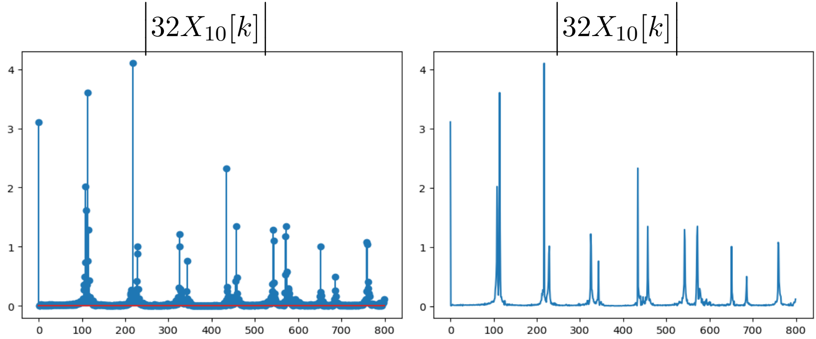
Lengthen by a factor of 16 (N=16,384).





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!

Anything else?

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 numer 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.

Recall the convolution result for the DTFT.

If

$$f_a[n] \overset{\text{DTFT}}{\Longrightarrow} F_a(\Omega)$$

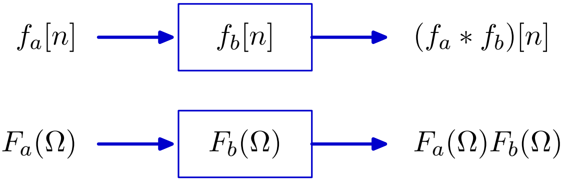
and

$$f_b[n] \overset{\text{DTFT}}{\Longrightarrow} F_b(\Omega)$$

then

$$(f_a * f_b)[n] \overset{\text{DTFT}}{\Longrightarrow} F_a(\Omega)F_b(\Omega)$$

This property is the basis of the **filtering** view of a system:



Regular Convolution

Why does multiplication in frequency correspond to convolution in time?

Let $F(\Omega) = F_a(\Omega) \times F_b(\Omega)$. Find $f[n]$.

$$
\begin{aligned}
f[n] &= \frac{1}{2\pi} \int_{2\pi} F(\Omega) e^{j\Omega n} \, d\Omega \\
&= \frac{1}{2\pi} \int_{2\pi} F_a(\Omega) F_b(\Omega) e^{j\Omega n} \, d\Omega \\
&= \frac{1}{2\pi} \int_{2\pi} F_a(\Omega) \left(\sum_{m=-\infty}^{\infty} f_b[m] e^{-j\Omega m} \right) e^{j\Omega n} \, d\Omega \\
&= \sum_{m=-\infty}^{\infty} f_b[m] \underbrace{\frac{1}{2\pi} \int_{2\pi} F_a(\Omega) e^{j\Omega(n-m)} \, d\Omega}_{f_a[n-m]} \\
&= \sum_{m=-\infty}^{\infty} f_b[m] f_a[n-m] \equiv (f_a * f_b)[n]
\end{aligned}
$$

Multiplying in frequency is equivalent to convolving in time.

Implementing Convolution with DFT

The argument for the DFT is similar to the one for the DTFT.

Let $F[k] = F_a[k] \times F_b[k]$. Find $f[n]$.

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

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

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

No! $f_a[n]$ is only defined for $0 \le n < N$, but $n-m$ can fall outside that range.

Implementing Convolution with DFT

The argument for the DFT is similar to the one for the DTFT.

Let $F[k] = F_a[k] \times F_b[k]$. Find $f[n]$.

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

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

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

Since k is integer, the complex exponential is periodic in $n-m$, period N. Therefore the parenthesized expression is $f_a[(n-m) \bmod N]$, and

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

Implementing Convolution with DFT

Multiplfying DFTs is equivalent to **circular** convolution in time.

If

$$f_a[n] \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad F_a[k]$$

and

$$f_b[n] \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad F_b[k]$$

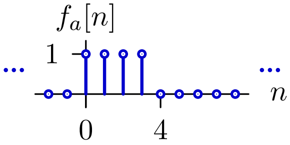
then

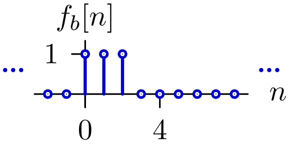
$$\frac{1}{N}(f_a \circledast f_b)[n] \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad F_a[k]F_b[k]$$

where

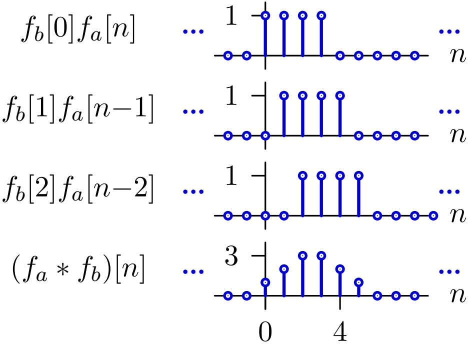
$$(f_a \circledast f_b)[n] = \sum_{m=0}^{N-1} f_b[m] f_a[(n-m) \bmod N]$$

Superposition View of Conventional Convolution

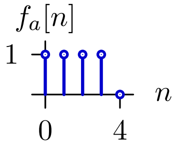


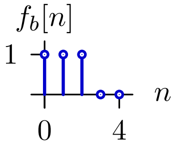


$$(f_a * f_b)[n] = \sum_{m=-\infty}^{\infty} f_b[m] f_a[n-m] = f_b[0] f_a[n] + f_b[1] f_a[n-1] + f_b[2] f_a[n-2]$$



Superposition View of Circular Convolution (N=5)

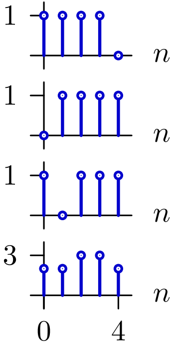




$$\left(f_a \circledast f_b\right)[n] = \sum_{m=0}^{N-1} f_b[m] f_a[(n-m) \bmod N]$$

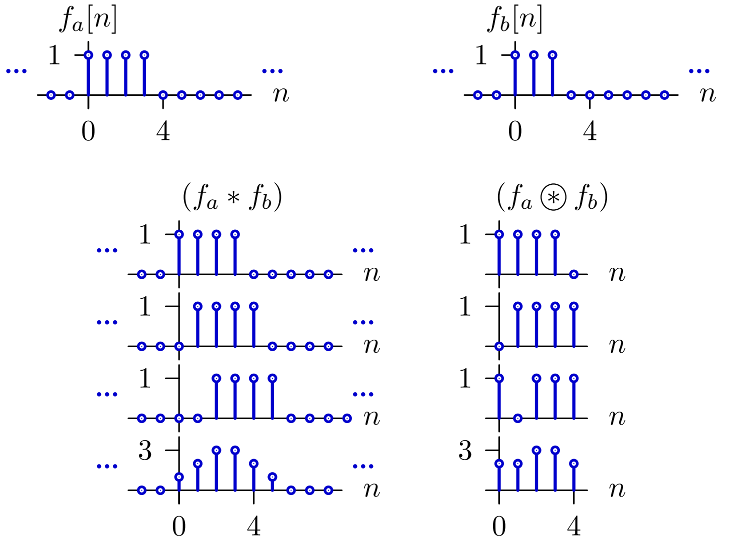
$$= f_b[0] f_a[n \bmod N] + f_b[1] f_a[(n-1) \bmod N] + f_b[2] f_a[(n-2) \bmod N]$$

$f_b[0] f_a[n \bmod N]$	$f_a[0]$	$f_a[1]$	$f_a[2]$	$f_a[3]$	$f_a[4]$
$f_b[1] f_a[(n-1) \bmod N]$	$f_a[4]$	$f_a[0]$	$f_a[1]$	$f_a[2]$	$f_a[3]$
$f_b[2] f_a[(n-2) \bmod N]$	$f_a[3]$	$f_a[4]$	$f_a[0]$	$f_a[1]$	$f_a[2]$
$(f_a \circledast f_b)[n]$					



Side By Side

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



Summary

Today we discussed two important 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.