INTRODUCTION TO THE SHORT-TIME FOURIER TRANSFORM (STFT)

Richard M. Stern

18-491 lecture

April 22, 2020

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Why consider short-time Fourier transforms?

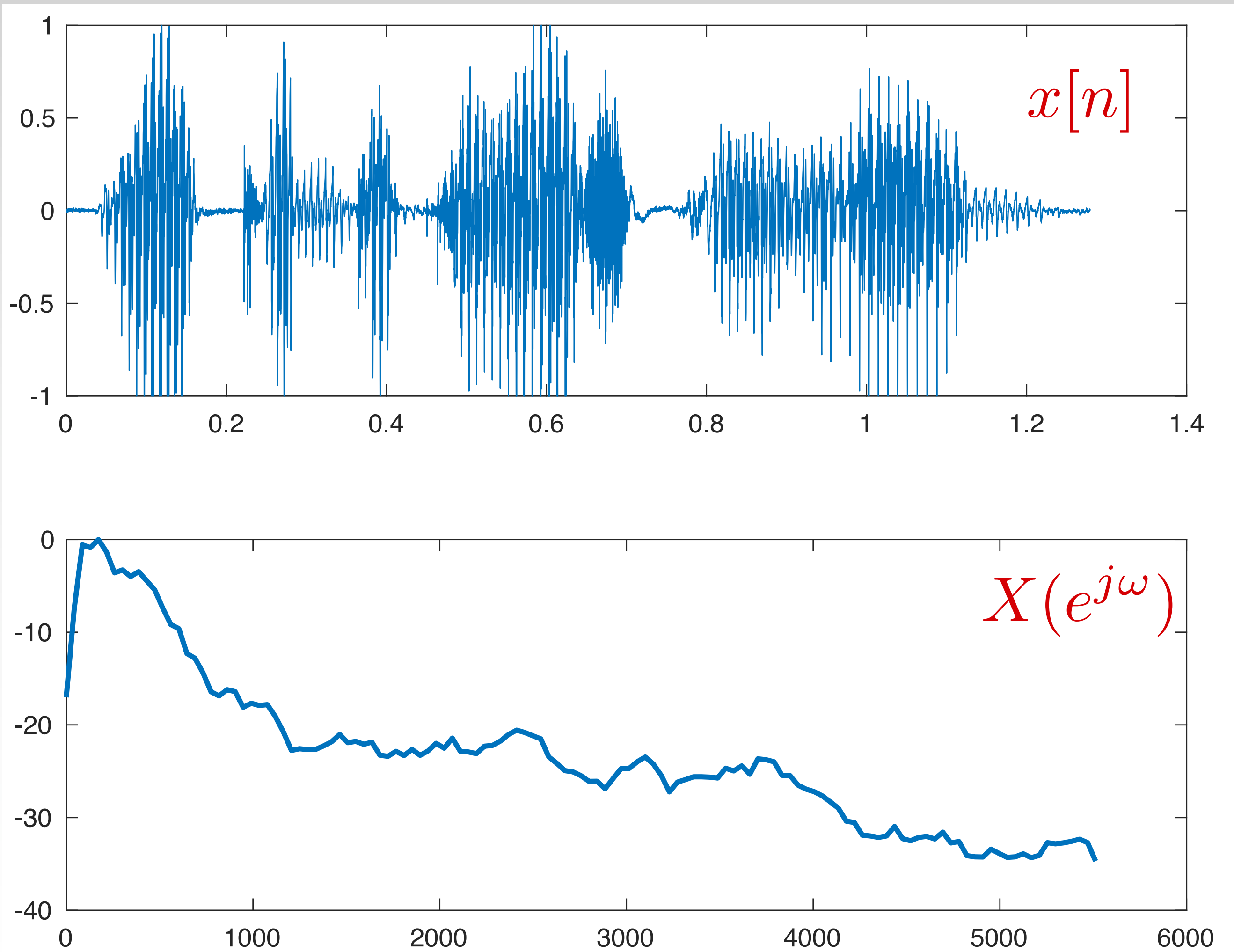
- **Conventional DTFT sums over all time:**

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

- **An example: “Welcome to DSP-I”**

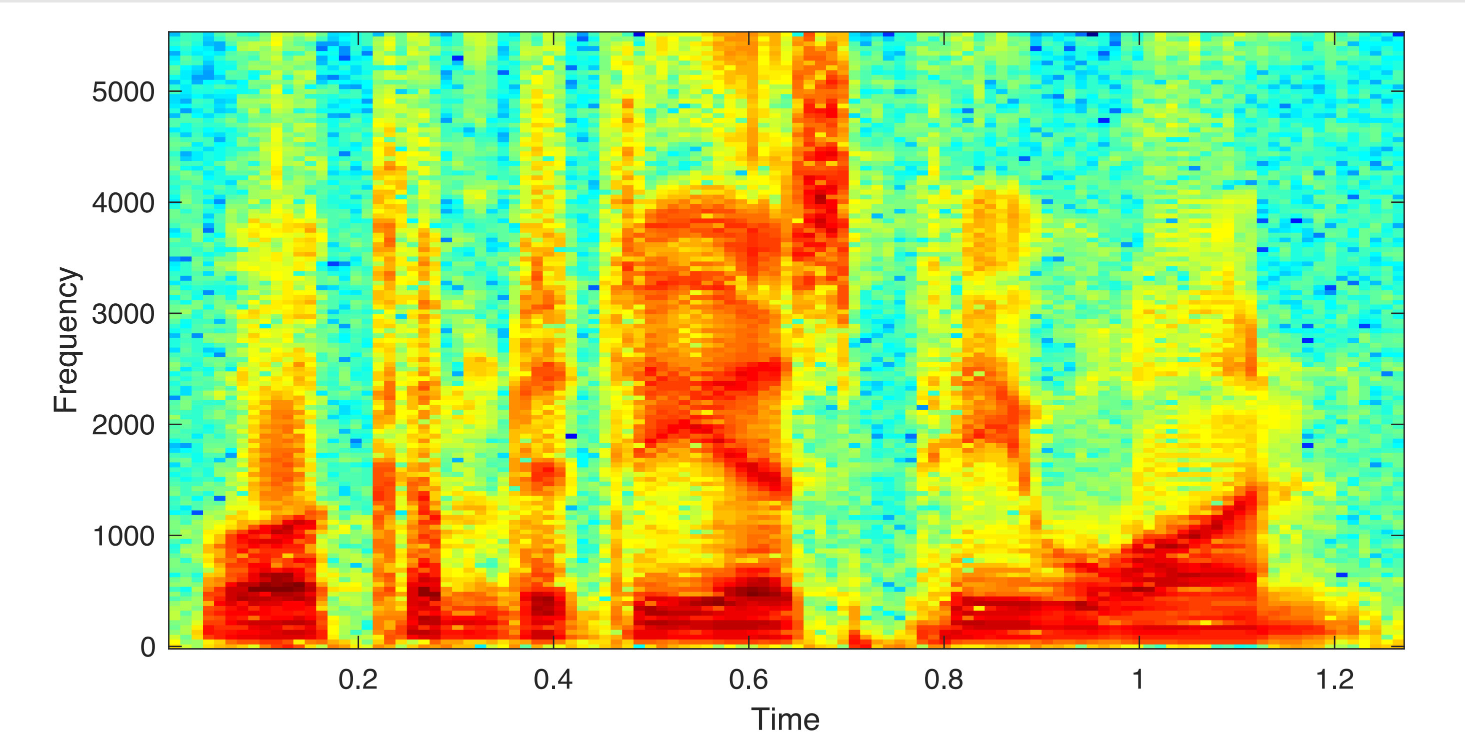
- **The DTFT averages frequency components over time**
  – (from the creation of the universe until ???}

"Welcome to DSP-I" in time and frequency



Why we want the STFT …

- We are more interested in how the frequency components of real sounds like speech and music vary over time
- Example: the spectrogram of “Welcome to DSP-I”



The direct (Fourier transform) approach to STFTs

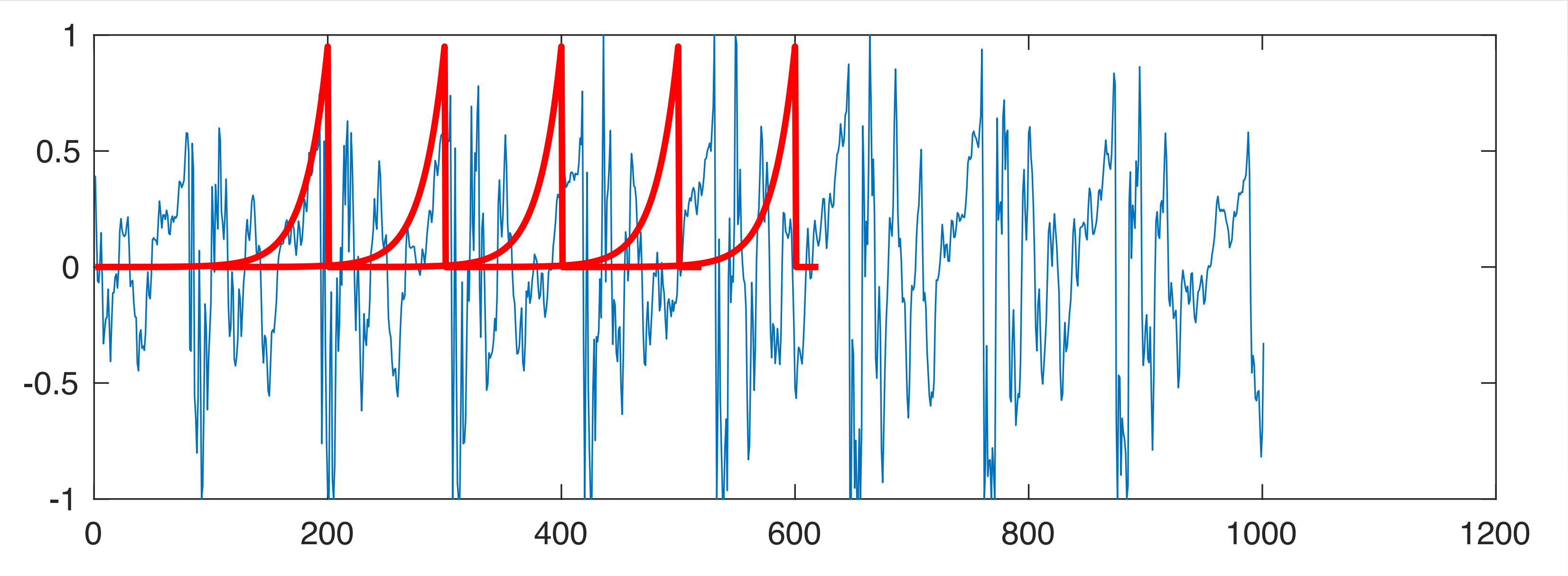
- **Multiply the time function and by a sliding window, and take the DTFT of the product:**

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

- **Comments:**
  - Note that m is a dummy variable and that the window is time-reversed
    - » Notation is consistent with chapter by Nawab and Quatieri in book edited by Lim and Oppenheim; OSPY notation is a little different
  - Results are plotted as a vector function of n, which is called the index of the analysis frame
  - Windows most commonly used are Hamming, rectangular, and exponential

An example with exponential windowing

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



Impact of window size and shape

- **The DTFT of the window is**

$$w[n-m] \Leftrightarrow \sum_{m=-\infty}^{\infty} w[n-m]e^{-j\omega m}$$

- **Letting** $l = m–n$ **and** $m = n–l$**, we obtain**

$$w[n-m] \Leftrightarrow \sum_{l=-\infty}^{\infty} w[m-l]e^{-j\omega n-l} = e^{-j\omega n}W(e^{-j\omega})$$

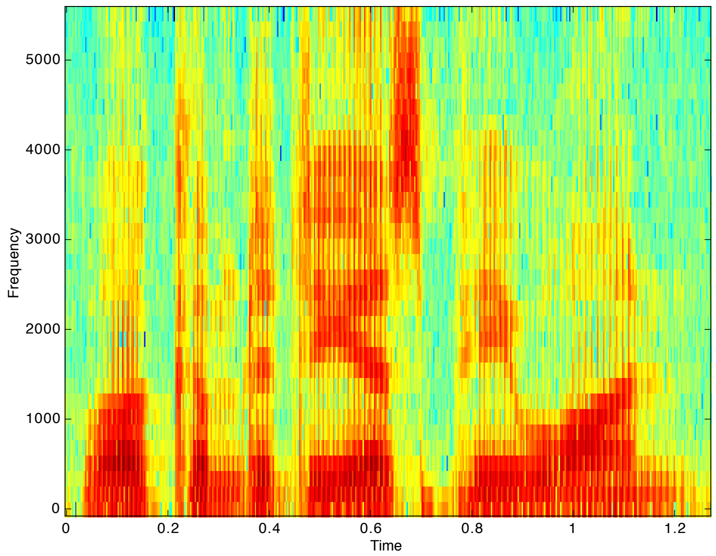
- **Hence …**

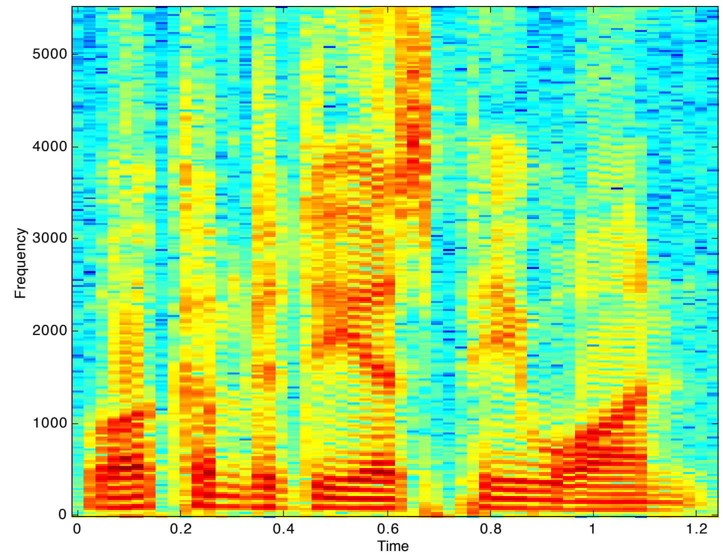
$$w[n-m]x[m] \Leftrightarrow \frac{1}{2\pi}(W(e^{-j\omega})e^{-j\omega n}) \otimes X(e^{j\omega})$$

- **The STFT can be thought of as the circular convolution in frequency of the DTFT of** $x[m]$ **with the DTFT of** $w[n–m]$

Effect of window duration

- **The window duration mediates the tradeoff between resolution in time and frequency:**
- **Short-duration window:** **Long-Duration window:**





- **Best choice of window duration depends on the application**

Can the STFT be inverted?

- Yes, but ….
- Consider the STFT as the transform of the windowed time function:

$$x[m]w[n-m] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X[n,\omega)e^{j\omega n}d\omega$$

- For $n{=}m$ we can write

$$x[n]w[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X[n,\omega)e^{j\omega n}d\omega$$

- Or, of course

$$x[n] = \frac{1}{2\pi w[0]}\int_{-\pi}^{\pi} X[n,\omega)e^{j\omega n}d\omega$$

- So the only absolute constraint for inversion is $w[0] \neq 0$

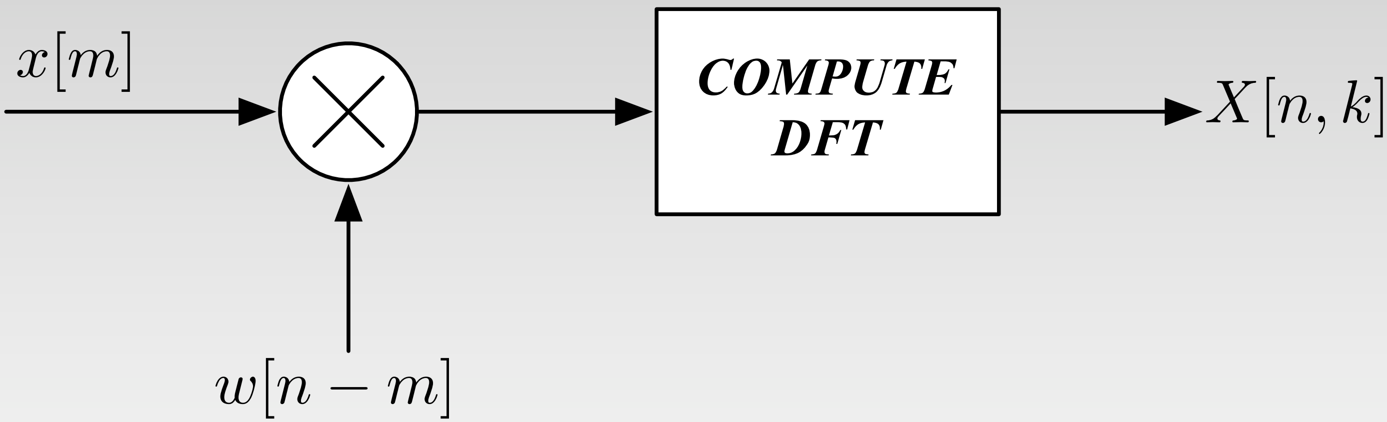
The discrete STFT

- **Normally we would like the STFT to be discrete in frequency as well as time** (for practical reasons)
- We use

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j2\pi km/N}$$

which is $X[n,\omega)$ evaluated at $\omega_k = 2\pi k/N$

Summary: the Fourier transform implementation of the STFT



- **The Fourier transform implementation of the STFT:**
  – Window input function
  – Take Fourier transform
  – Repeat, after shifting window

There are other ways of computing the STFT!

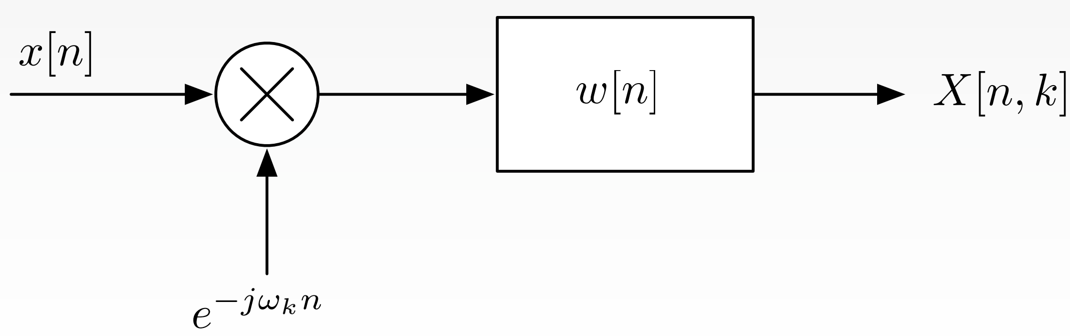
- **Again, the STFT equation is**

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j2\pi km/N}$$

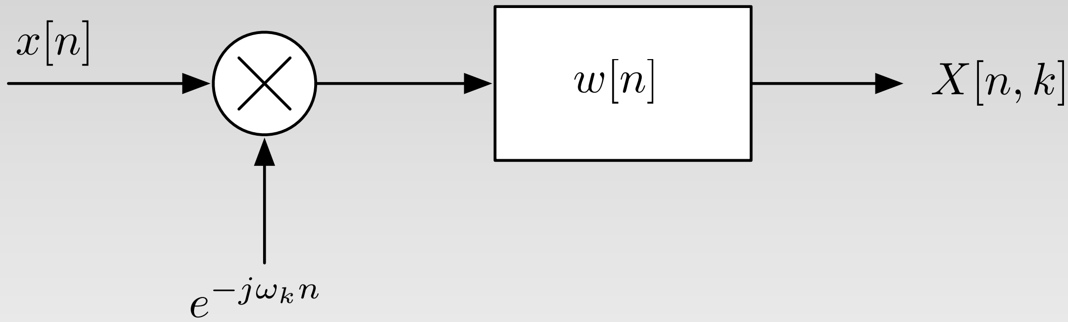
- **Rearranging the terms, we obtain the convolution**

$$X[n,k] = \sum_{n=-\infty}^{\infty} w[n-m](x[m]e^{-j2\pi mk/N})$$

- **This can expressed as the lowpass implementation of the STFT:**



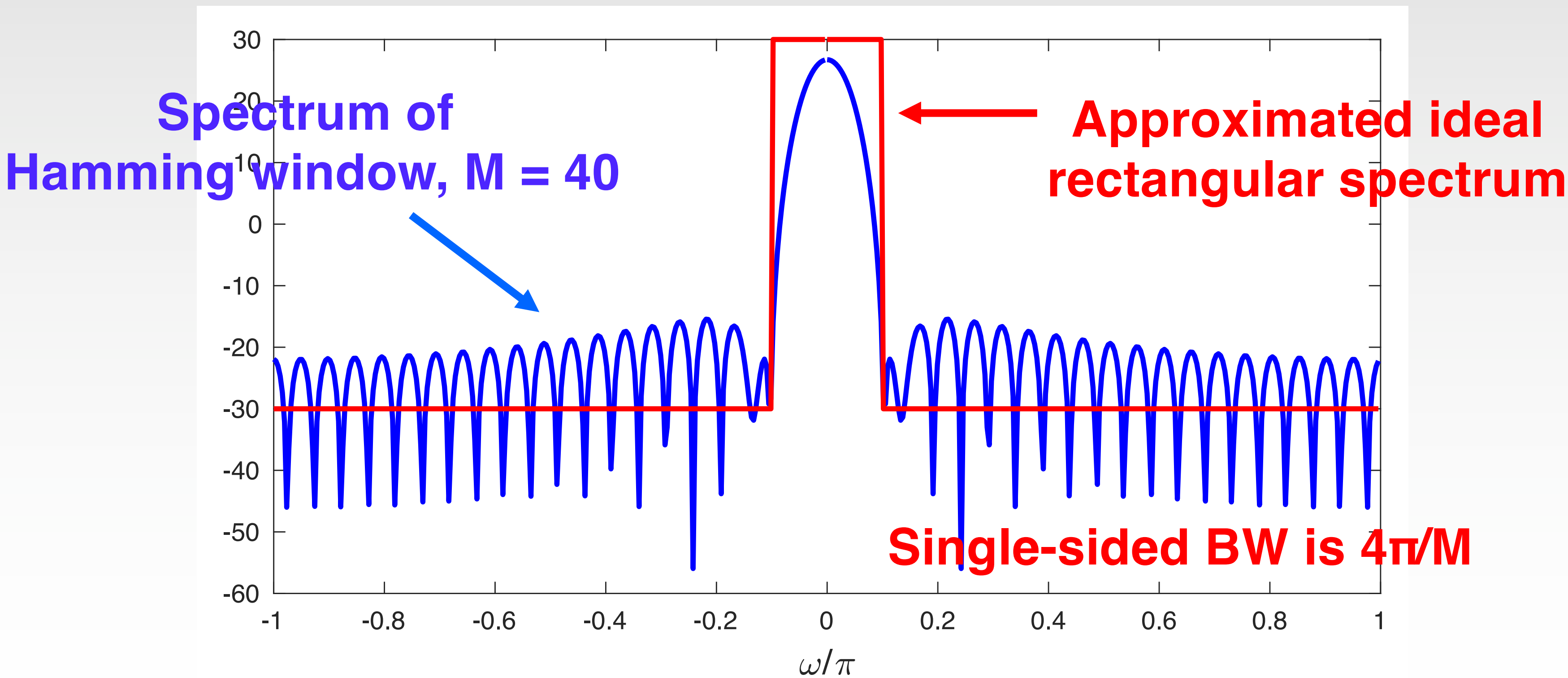
The lowpass implementation of the STFT



- **Note that the frequency response of practical windows $w[n]$ is almost invariably that of a lowpass filter**
- **The lowpass implementation translates the spectrum of $x[n]$ to the left by ω_k radians and passes through a lowpass filter**

The Hamming window as a lowpass filter

- The width of the main lobe of a Hamming window is $8\pi/M$
- We will think of it as if it were an ideal LPF with the same bandwidth



Also, the bandpass implementation of the STFT

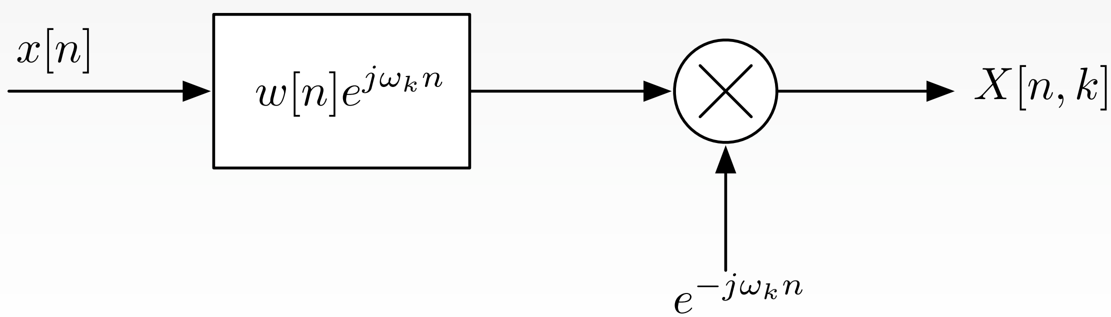
- **The original STFT equation remains**

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j2\pi km/N}$$

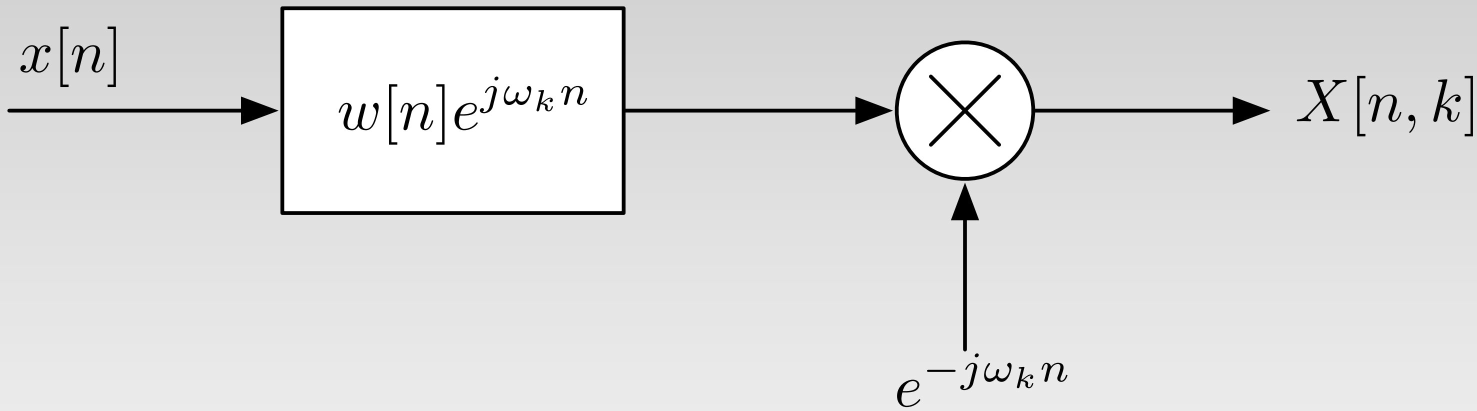
- **Pre-multiplying and post-multiplying by $e^{-\omega_k n}$ produces**

$$X[n,k] = e^{-}j\omega_k n \sum_{n=-\infty}^{\infty} (w[n-m]e^{j\omega_k(n-m)})x[m]$$

- **Which can be expressed as the bandpass implementation of the SFFT:**



The bandpass implementation of the STFT



- The bandpass implementation can be thought of as passing the signal through a (single-channel) bandpass filter and then shifting the output down to "baseband"
- All three implementations are mathematically equivalent representations of the STFT
- The signal at the output of the BPF has the same magnitude as $X[n,k]$ but different phase

Some additional comments on implementations

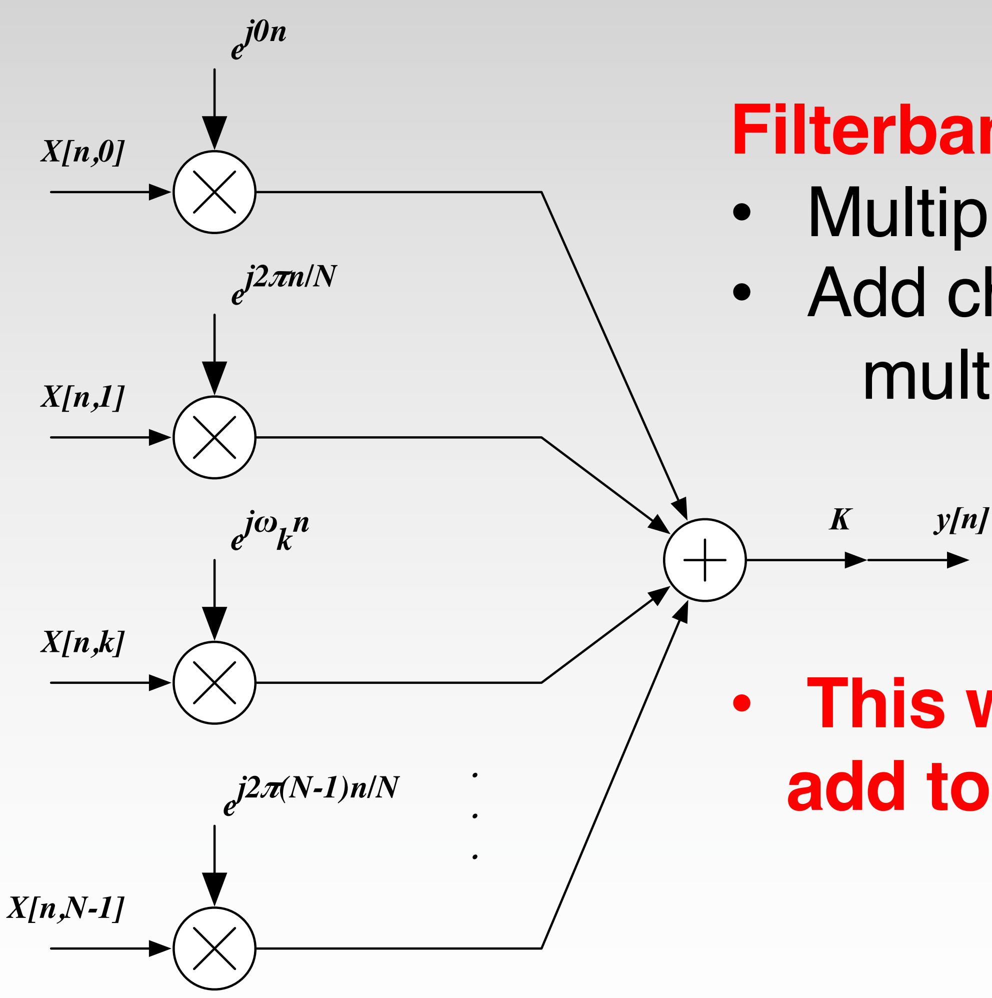
- In the Fourier transform implementation will develop the STFT on a column-by-column (or time frame by time frame) basis
- In the LP and BP implementations we work on a row-by-row (or frequency-by-frequency) basis

- Because the STFT is lowpass in nature, it can be downsampled. The downsampling ratio depends on the size and shape of the window.

Reconstructing the time function

- **Two major methods used:**
  - Filterbank summation (FBS), based on LP and BP implementations
  - Overlap-add (OLA), based on the Fourier transform implementation

Reconstructing the time function using FBS



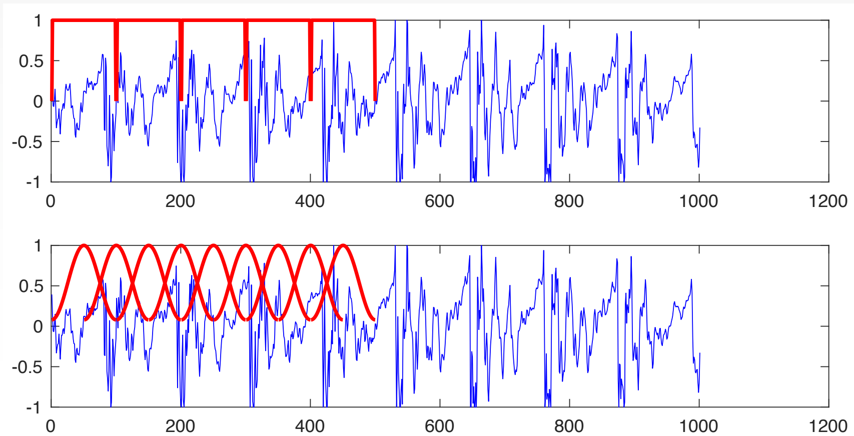
Filterbank summation:

- Multiply each channel by $e^{\omega_k n}$
- Add channels together and multiply by a constant

- **This will work if all filters add to a constant in frequency**

The overlap-add (OLA) method of reconstruction

- **Procedure:**
  - Compute the IDTFT for each column of the STFT
  - Add the IDTFTs together in the locations of the original window locations
- **The OLA resynthesis approach will work if all of the windows add up to a constant. Two (of many) solutions:**
  - Abutting rectangular windows
  - Hamming windows spaced by 50% of their length



How many numbers do we need to keep?

- **The answer depends on the method used for analysis and synthesis.**
- **For the Fourier transform STFT analysis with OLA resynthesis:**
  - Need at least **N** samples in frequency for windows of length **N** (as is always true for DFTs)
  - The analysis frames can be separated by N samples for rectangular windows or N/2 samples for Hamming windows
  - This means that the total number of STFT coefficients per second needed will be $NF_s/N = F_s$ for rectangular windows or $NF_s/(N/2)$ for Hamming windows
- **Hence, the STFT requires the same or double the number of numbers in the original waveform. (And these numbers are complex!) We accept this for the benefits that STFTs provide**

Summary

- Short-time Fourier transforms enable us to analyze how frequency components evolve over time. The most straightforward approach is to window the time function and compute the DFT
- The duration of the window mediates temporal versus spectral resolution
- The original waveform can be resynthesized from the STFT representation
- The number of numbers needed for the representation is somewhat greater, but that is a small price to pay for the ability to analyze and manipulate the input.

[redacted]