

L6: Short-time Fourier analysis and synthesis

Overview

Analysis: Fourier-transform view

Analysis: filtering view

Synthesis: filter bank summation (FBS) method

Synthesis: overlap-add (OLA) method

STFT magnitude

This lecture is based on chapter 7 of [Quatieri, 2002]

Overview

Recap from previous lectures

– Discrete time Fourier transform (DTFT)

- Taking the expression of the Fourier transform $X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$, the DTFT can be derived by numerical integration

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

– where $x[n] = x(nT_S)$ and $\hat{\omega} = 2\pi F/F_S$

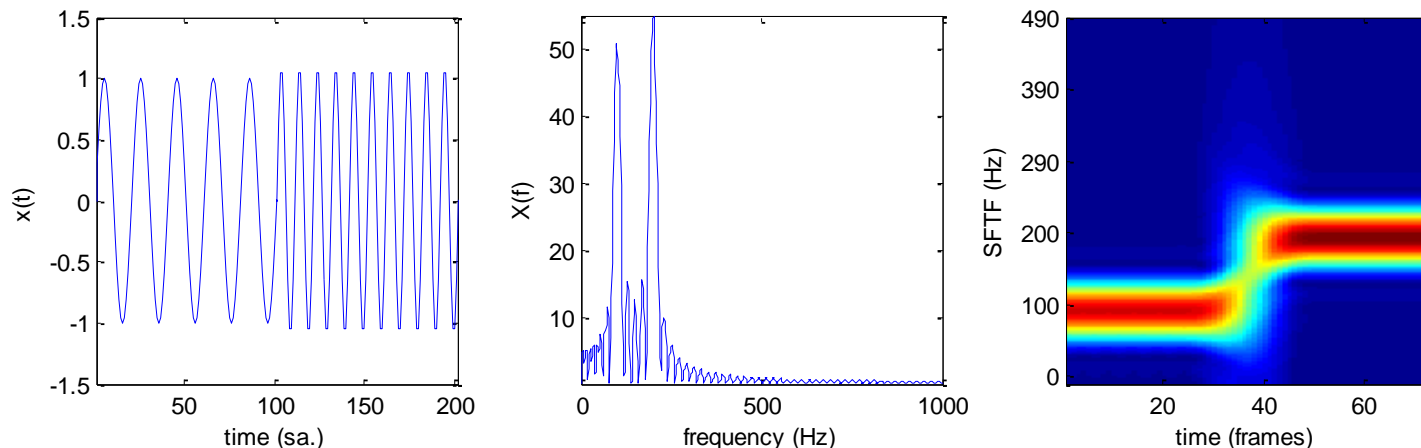
– Discrete Fourier transform (DFT)

- The DFT is obtained by “sampling” the DTFT at N discrete frequencies $\omega_k = 2\pi F_S/N$, which yields the transform

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

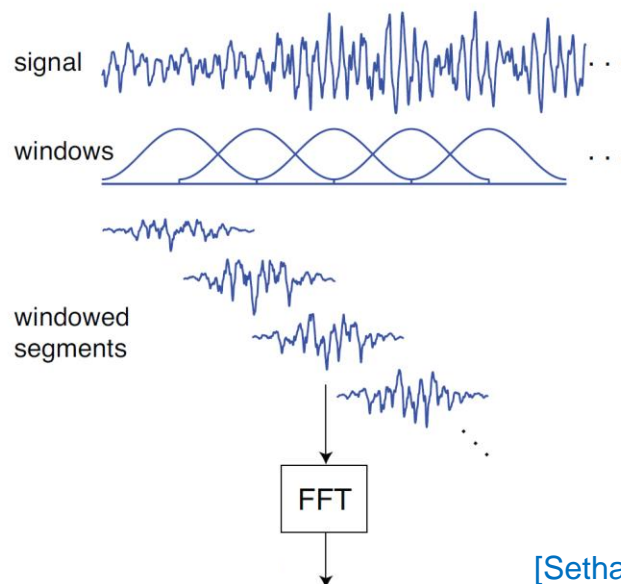
Why is another Fourier transform needed?

- The spectral content of speech changes over time (non stationary)
 - As an example, formants change as a function of the spoken phonemes
 - Applying the DFT over a long window does not reveal transitions in spectral content
- To avoid this issue, we apply the DFT over short periods of time
 - For short enough windows, speech can be considered to be stationary
 - Remember, though, that there is a time-frequency tradeoff here



The short-time Fourier transform in a nutshell

- Define analysis window (e.g., 30ms narrowband, 5 ms wideband)
- Define the amount of overlap between windows (e.g., 30%)
- Define a windowing function (e.g., Hann, Gaussian)
- Generate windowed segments (multiply signal by windowing function)
- Apply the FFT to each windowed segment



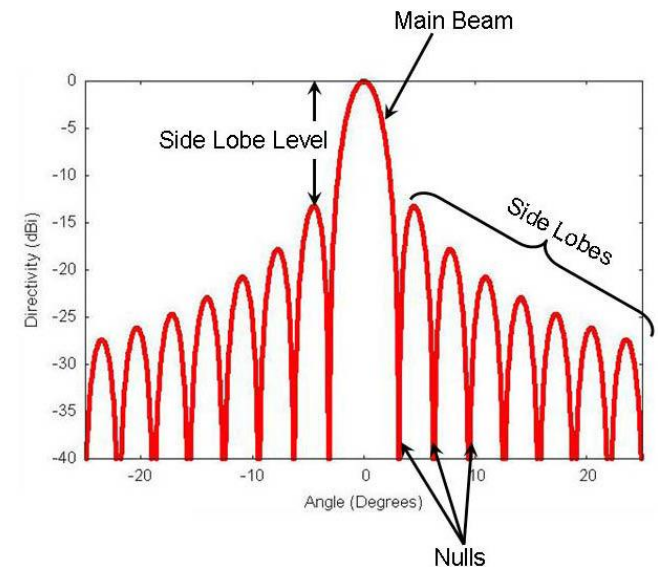
[Sethares, 2007]

STFT: Fourier analysis view

Windowing function

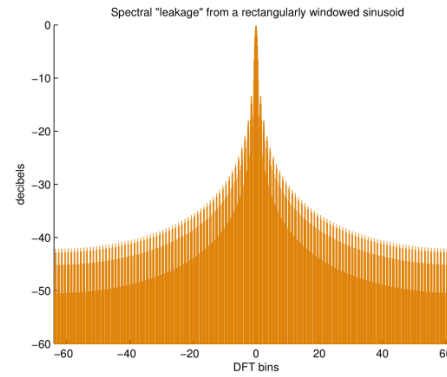
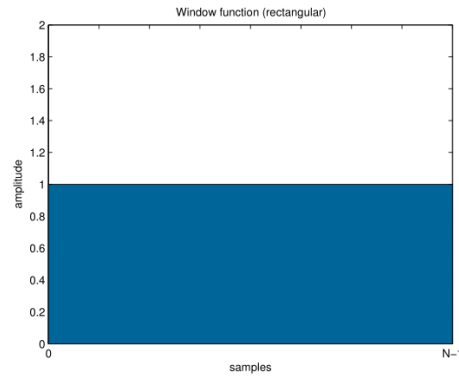
- To “localize” the speech signal in time, we define a windowing function $w[n, \tau]$, which is generally tapered at its ends to avoid unnatural discontinuities in the speech segment
- Any window affects the spectral estimate computed on it
 - The window is selected to trade off the width of its main lobe and attenuation of its side lobes
- The most common are the Hann and Hamming windows (raised cosines)

$$w[n, \tau] = 0.54 - 0.4 \cos \left[\frac{2\pi(n - \tau)}{N_w - 1} \right]$$
$$w[n, \tau] = 0.5 \left(1 - \cos \left(\frac{2\pi(n - \tau)}{N - 1} \right) \right)$$

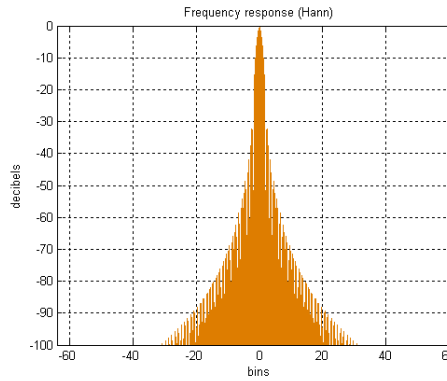
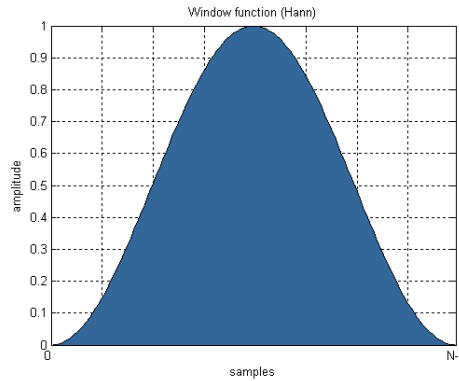


http://en.wikipedia.org/wiki/Window_function

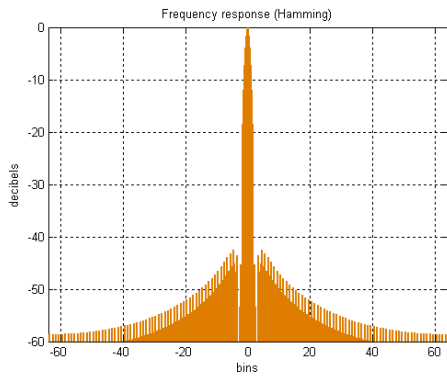
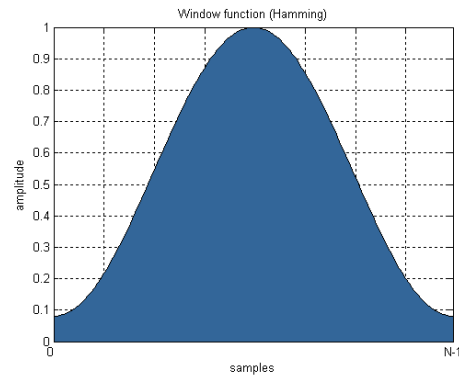
Rectangular



Hann



Hamming



http://en.wikipedia.org/wiki/Window_function

Discrete-time Short-time Fourier transform

- The Fourier transform of the windowed speech waveform is defined as

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

- where the sequence $f_n[m] = x[m]w[n-m]$ is a short-time section of the speech signal $x[m]$ at time n

Discrete STFT

- By analogy with the DTFT/DFT, the discrete STFT is defined as

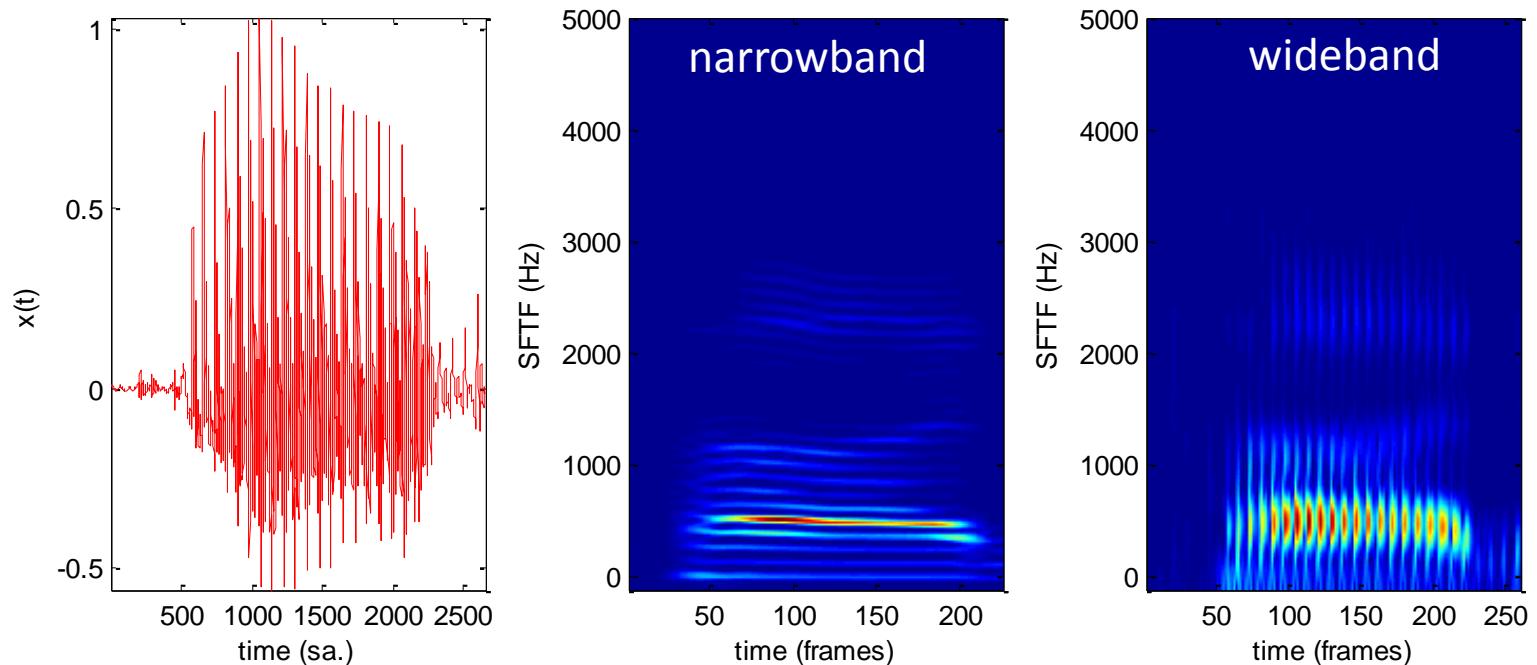
$$X(n, k) = X(n, \omega) \Big|_{\omega = \frac{2\pi}{N}k}$$

- The spectrogram we saw in previous lectures is a graphical display of the magnitude of the discrete STFT, generally in log scale

$$S(n, k) = \log|X(n, k)|^2$$

- This can be thought of as a 2D plot of the relative energy content in frequency at different time locations

- For a long window $w[n]$, the result is the narrowband spectrogram, which exhibits the harmonic structure in the form of horizontal striations
- For a short window $w[n]$, the result is the wideband spectrogram, which exhibits periodic temporal structure in the form of vertical striations



STFT: filtering view

The STFT can also be interpreted as a filtering operation

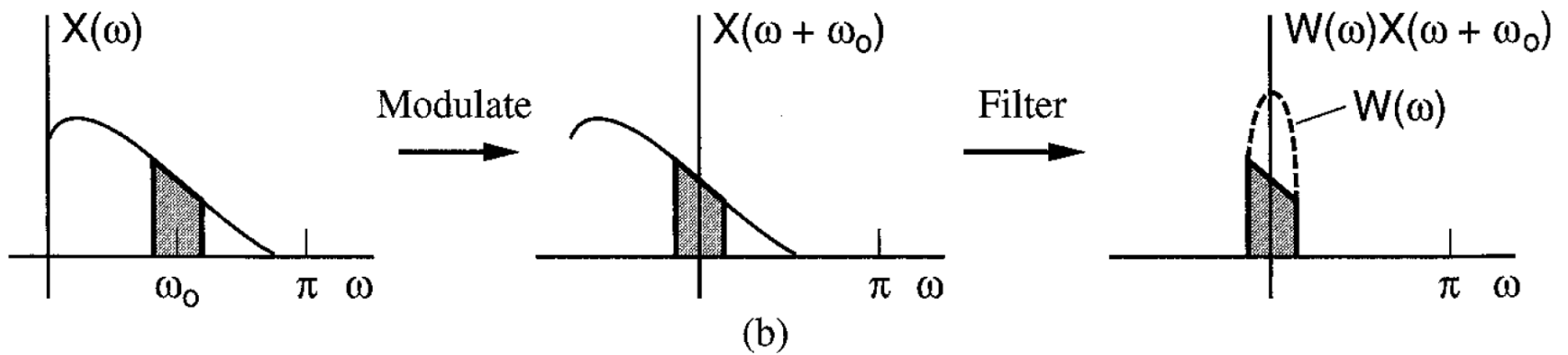
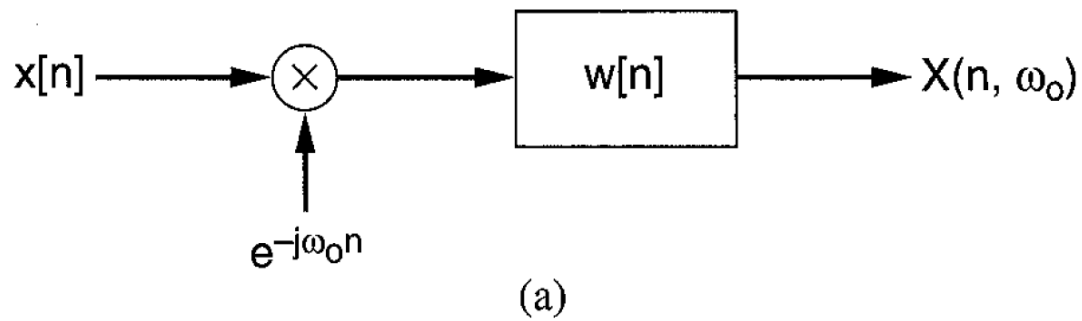
- In this case, the analysis window $w[n]$ plays the role of the filter impulse response
- To illustrate this view, we fix the value of ω at ω_0 , and rewrite

$$X(n, \omega_0) = \sum_{m=-\infty}^{\infty} (x[m]e^{-j\omega_0 m})w[n - m]$$

- which can be interpreted as the convolution of the signal $(x[n]e^{-j\omega_0 n})$ with the sequence $w[n]$:

$$X(n, \omega_0) = (x[n]e^{-j\omega_0 n}) * w[n]$$

- and the product $x[n]e^{-j\omega_0 n}$ can be interpreted as the modulation of $x[n]$ up to frequency ω_0 (i.e., per the frequency shift property of the FT)

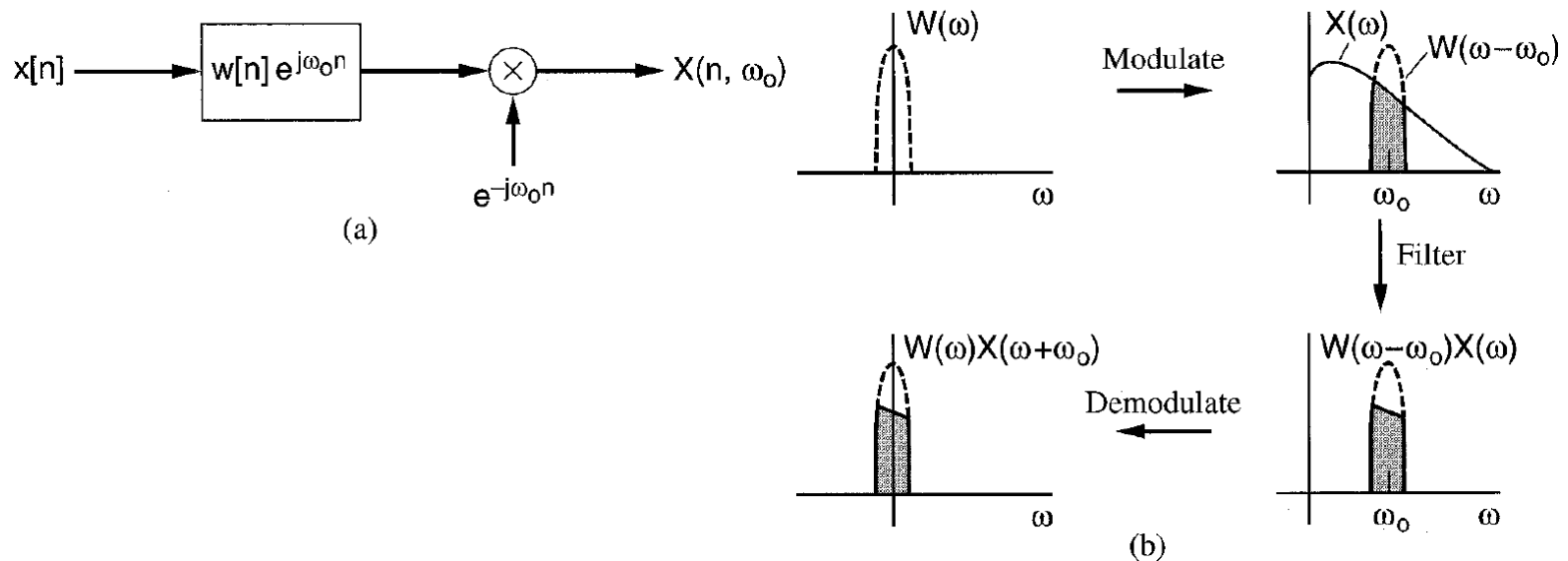


[Quatieri, 2002]

– Alternatively, we can rearrange as (without proof)

$$X(n, \omega_0) = e^{-j\omega_0 n} (x[n] * w[n] e^{j\omega_0 n})$$

- In this case, the sequence $x[n]$ is first passed through the same filter (with a linear phase factor $e^{j\omega_0 n}$), and the filter output is demodulated by $e^{-j\omega_0 n}$

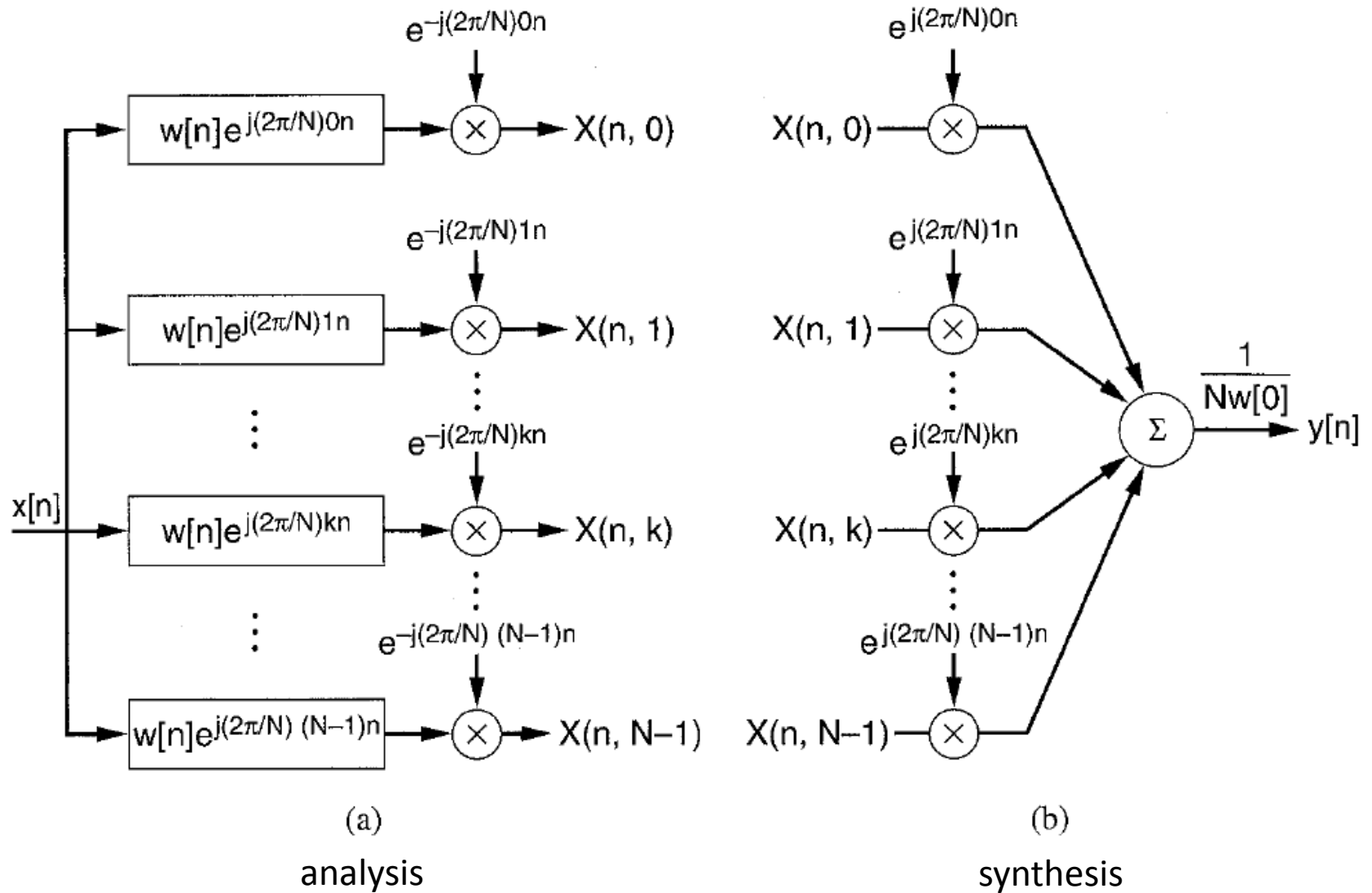


[Quatieri, 2002]

- This later rearrangement allows us to interpret the discrete STFT as the output of a filter bank

$$X(n, k) = e^{-j\frac{2\pi}{N}kn} \left(x[n] * w[n] e^{-j\frac{2\pi}{N}kn} \right)$$

- Note that each filter is acting as a bandpass filter centered around its selected frequency
- Thus, the discrete STFT can be viewed as a collection of sequences, each corresponding to the frequency components of $x[n]$ falling within a particular frequency band
 - This filtering view is shown in the next slide, both from the analysis side and from the synthesis (reconstruction) side



[Quatieri, 2002]

Examples

[ex6p1.m](#)

Generate STFT using Matlab functions

[ex6p2.m](#)

Generate filterbank outputs using the filtering view of the STFT

[ex6p3.m](#)

Time-frequency resolution tradeoff (Quatieri fig 7.8)

Short-time synthesis

Under what conditions is the STFT invertible?

- The discrete-time STFT $X(n, \omega)$ is generally invertible

- Recall that

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

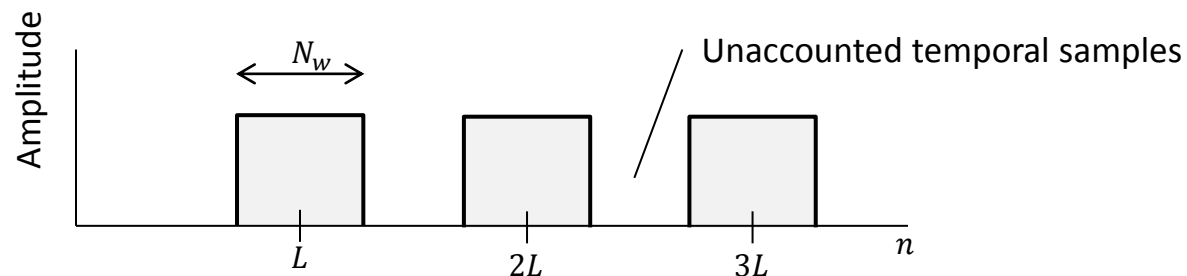
$$\text{with } f_n[m] = x[m]w[n - m]$$

- Evaluating $f_n[m]$ at $m = n$ we obtain $f_n[n] = x[n]w[0]$
- So assuming that $w[0] \neq 0$, we can estimate $x[n]$ as

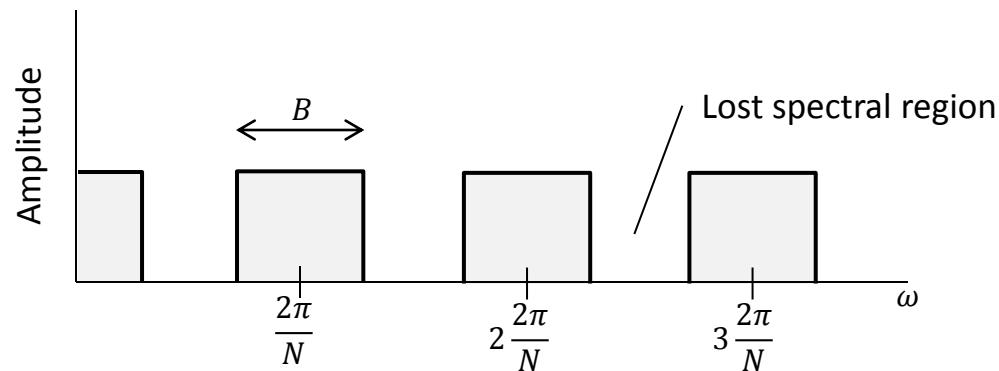
$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n, \omega) e^{j\omega n} d\omega$$

- This is known as a *synthesis equation* for the DT STFT

- Redundancy of the discrete-time STFT
 - There are many synthesis equations that map $X(n, \omega)$ uniquely to $x[n]$
 - Therefore, the STFT is very redundant if we move the analysis window one sample at a time ($n = 1, 2, 3 \dots$)
 - For this reason, the STFT is generally computed by decimating over time, that is, at integer multiples ($n = L, 2L, 3L \dots$)
- For large L , however, the DT STFT may become non-invertible
 - As an example, assume that $w[n]$ is nonzero over its length N_w
 - In this case, when $L > N_w$, there are some samples of $x[n]$ that are not included in the computation of $X(n, \omega)$
 - Thus, these samples can have arbitrary values yet yield the same $X(kL, \omega)$
 - Since $X(kL, \omega)$ is not uniquely defined, it is not invertible



- Likewise, the discrete STFT $x(n, k)$ is not always invertible
 - Consider the case where $w[n]$ is band-limited with bandwidth B
 - If the sampling interval $2\pi/N$ is greater than B , some of the frequency components in $x[n]$ do not pass through any of the filters of the STFT
 - Thus, those frequency components can have any arbitrary values yet produce the same discrete STFT
 - In consequence, depending on the frequency sampling resolution, the discrete STFT may become non invertible



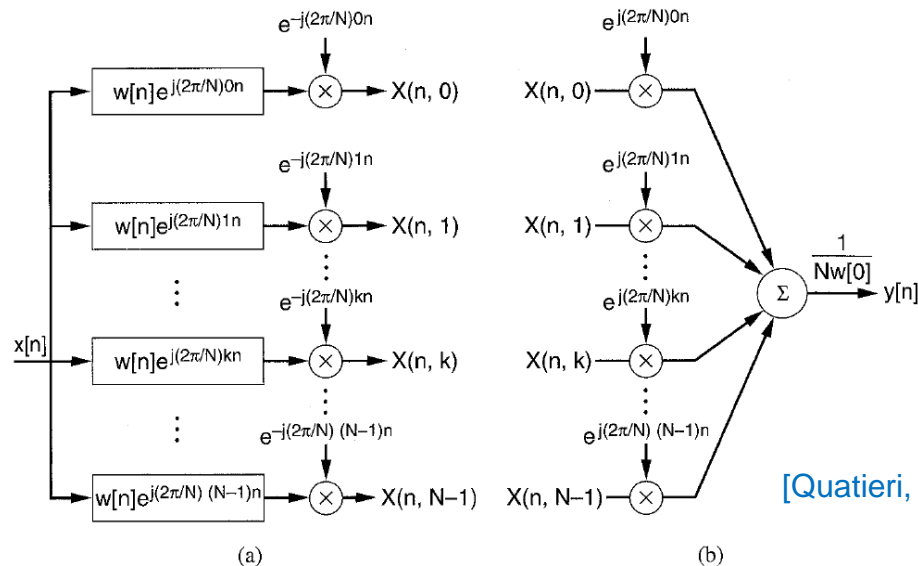
[Quatieri, 2002]

Synthesis: filter bank summation

FBS is based on the filtering interpretation of the STFT

- As we saw earlier, according to this interpretation the discrete STFT is considered to be the set of outputs from a bank of filters
- In the FSB method, the output of each filter is modulated with a complex exponential, and these outputs are summed to recover the original signal

$$y(n) = \frac{1}{Nw[0]} \sum_{m=-\infty}^{\infty} X(n, k) e^{j\frac{2\pi}{N}nk}$$

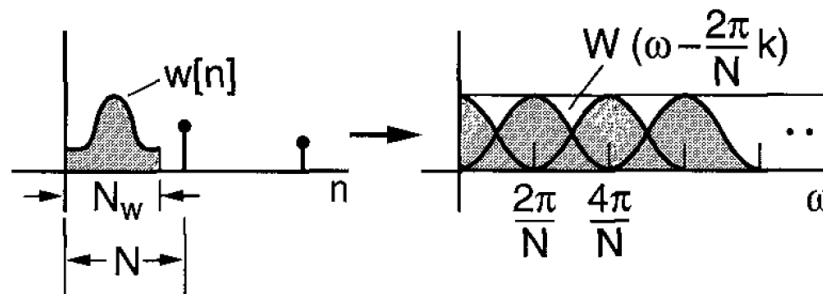


– Under which conditions does FBS yield exact synthesis?

- It can be shown that $y[n] = x[n]$ if either
 1. The length of $w[n]$ is less than or equal to the no. of filters ($N_w \geq N$), or
 2. For $N_w > N$:

$$\sum_{k=0}^{N-1} W\left(\omega - \frac{2\pi}{N}k\right) = Nw[0]$$

- The latter is known as the *BFS constraint*, and states that the frequency response of the analysis filters should sum to a constant across the entire bandwidth



[Quatieri, 2002]

Synthesis: Overlap-add

OLA is based on the Fourier transform view of the STFT

- In the OLA method, we take the inverse DFT for each fixed time in the discrete STFT
- In principle, we could then divide by the analysis window
 - This method is not used, however, as small perturbations in the STFT can become amplified in the estimated signal $y[n]$
- Instead, we perform an OLA operation between the sections
 - This works provided that $w[n]$ is designed such that the OLA effectively eliminates the analysis windows from the synthesized sequence
 - The intuition is that the redundancy within overlapping segments and the averaging of the redundant samples averages out the effect of windowing
- Thus, the OLA method can be expressed as

$$y[n] = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} \left[\sum_{k=0}^{N-1} X(p, k) e^{j\frac{2\pi}{N}kn} \right]$$

- where the term inside the square brackets is the IDFT

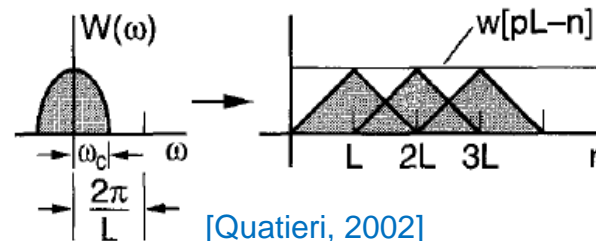
– Under which conditions does OLA yield exact synthesis?

- It can be shown that if the discrete STFT has been decimated by a factor L , the condition $y[n] = x[n]$ is met when

$$\sum_{p=-\infty}^{\infty} w[pL - n] = \frac{W(0)}{L}$$

- which holds when either
 1. The analysis window has finite bandwidth with maximum frequency ω_c less than $2\pi/L$, or
 2. The sum of all the analysis windows (obtained by sliding $w[n]$ with L -point increments) adds up to a constant
- In this case, $x[n]$ can then be resynthesized as

$$x[n] = \frac{L}{W(0)} \sum_{p=-\infty}^{\infty} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(pL, k) e^{j\frac{2\pi}{N}kn} \right]$$



[Quatieri, 2002]

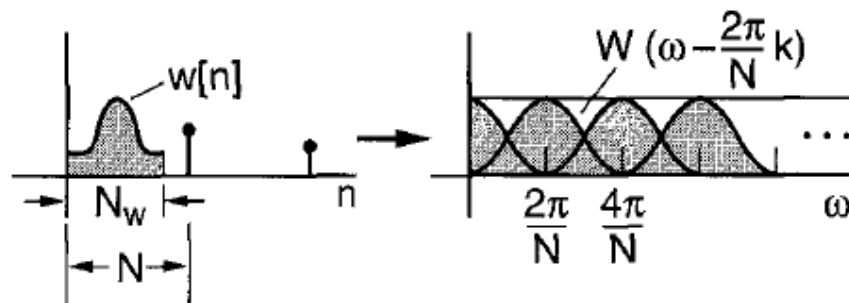
FBS Method

$$y[n] = \left[\frac{1}{Nw[0]} \right] \underbrace{\sum_{k=0}^{N-1} X(n, k) e^{j \frac{2\pi}{N} kn}}_{\text{Adding Frequency Components For Each } n}$$

Adding Frequency
Components For Each n

FBS Constraint: $\sum_{k=0}^{N-1} W(\omega - \frac{2\pi}{N}k) = Nw[0]$

For $N_w < N \rightarrow y[n] = x[n]$



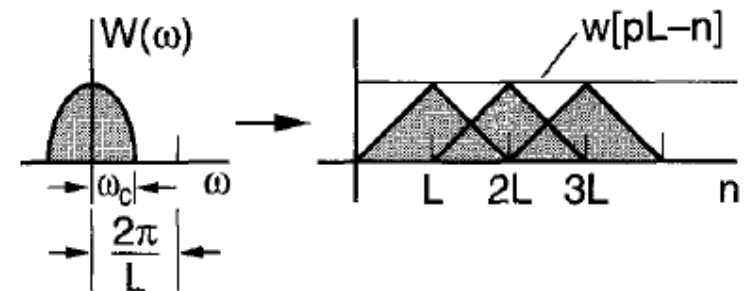
OLA Method

$$y[n] = \left[\frac{L}{W(0)} \right] \underbrace{\sum_{p=-\infty}^{\infty} x[n] w[pL-n]}_{\text{Adding Time Components For Each } n}$$

Adding Time Components
For Each n

OLA Constraint: $\sum_{p=-\infty}^{\infty} w[pL-n] = \frac{W(0)}{L}$

For $\omega_c < \frac{2\pi}{L} \rightarrow y[n] = x[n]$



[Quatieri, 2002]

STFT magnitude

The spectrogram (STFT magnitude) is widely used in speech

- For one, evidence suggests that the human ear extracts information strictly from a spectrogram representation of the speech signal
- Likewise, trained researchers can visually “read” spectrograms, which further indicates that the spectrogram retains most of the information in the speech signal (at least at the phonetic level)
- Hence, one may question whether the original signal $x[n]$ can be recovered from $|X(n, \omega)|$, that is, by ignoring phase information

Inversion of the STFTM

- Several methods may be used to estimate $x[n]$ from the STFTM
- Here we focus on a fairly intuitive least-squares approximation

Least-squares estimation from the STFT magnitude

- In this approach, we seek to estimate a sequence $x_e[n]$ whose STFT magnitude $|X_e(n, \omega)|$ is “closest” (in a least-squared-error sense) to the known STFT magnitude $|X(n, \omega)|$
- The iteration takes place as follows
 - An arbitrary sequence (usually white noise) is selected as the first estimate $x_e^1[n]$
 - We then compute the STFT of $x_e^1[n]$ and modify it by replacing its magnitude by that of $|X(n, \omega)|$

$$X^1(m, \omega) = |X(m, \omega)| \frac{X_e^i(m, \omega)}{|X_e^i(m, \omega)|}$$

- From this, we obtain a new signal estimate as

$$x_e^i[n] = \frac{\sum_{m=-\infty}^{\infty} w[m-n] g_m^{i-1}[n]}{\sum_{m=-\infty}^{\infty} w^2[m-n]}$$

where $g_m^{i-1}[n]$ is the inverse DFT of $X^{i-1}(m, \omega)$

- And the process continues iteratively until convergence or a stopping criterion is met

- It can be shown that this process reduces the distance between $|X_e(n, \omega)|$ and $|X(n, \omega)|$ at each iteration
- Thus, the process converges to a local minimum, though not necessarily a global minimum

– All steps in the iteration can be summarized as (Quatieri, 2002; p. 342)

$$x_e^{i+1}[n] = \frac{\sum_{m=-\infty}^{\infty} w[m-n] \frac{1}{2\pi} \int_{-\pi}^{\pi} X^i(m, \omega) e^{j\omega n} d\omega}{\sum_{m=-\infty}^{\infty} w^2[m-n]}$$

$$\text{where } X^i(m, \omega) = |X(m, \omega)| \frac{X_e^i(m, \omega)}{|X_e^i(m, \omega)|}$$

Example

ex6p4.m

Estimate a signal from its STFT magnitude