The Overlap-Add (OLA) Systems for Audio Analysis and Synthesis

Yi-Wen Liu

11 March 2013

1 Motivation

In previous lectures we covered the convolution theorem; convolution in the time domain is equivalent to multiplication in the frequency domain. Thanks to the fast Fourier transform (FFT), convolution can be done more efficiently in the frequency domain than in the time domain. An immediate application would be FIR (finite impulse-response) filtering. There are a few reasons to process audio signals in a block-wise manner, as shown in Fig. 1:

- The signal may never end but the memory space is limited.
- For real-time applications, we need to produce output as the input is coming in.
- (*Adaptivity*) as the input changes, we may want to change the way it is filtered.

In the case of FIR filtering, each block of length N is convolved with the filter of length L via FFT. Be aware that zeros need to be appended to the end of both the block and the filter before taking the FFT. This operation is called *zero-padding*, and its purpose is to ensure that cyclic convolution does not cause the filtered signal to wrap around itself in the time domain. Usually, we zero-pad to the next lowest power of 2 such that the FFT length 2^K is longer than $N + L - 1$ (Why?).

2 The OLA synthesis: putting things back together

2.1 Rectangular windowing

When we do FIR filtering using the OLA system, we start with blocks of length N but obtain output blocks of length $N + L - 1$. So, be careful to overlap the output blocks correctly: the hope size should still be N and adjacent blocks overlap by $L - 1$ samples.

In certain applications, we need to vary the filter while a signal is still coming in. For example, imagine the scenario when a user could be sliding a bar on its computing device (computer, smart phone, iPad, etc) to vary the cut-off frequency, gain, or other characteristics of the filter. Rectangular window is known to be less appropriate for *time-varying* applications because it creates "hard boundaries" on both sides. Next, we

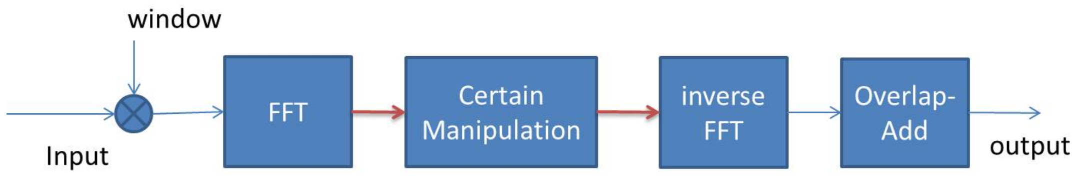

Figure 1: A general block diagram of OLA analysis and synthesis system

introduce a more general class of windows that enable us to do time-varying filtering *without glitches*.

2.2 The constant overlap-add (COLA) criterion

The Hann window (see Appendix for definition) is a representative window that reduces audible glitches for time-varying signal processing. It belongs to a class of windows $w(n)$ that satisfy the constant overlap-add (COLA) criterion:

$$\sum_{k=-\infty}^{\infty} w[n-kM] = 1, \tag{1}$$

where M is a *hopsize* that enables the moving sum (i.e., "overlap-add") of the window to become constant.

For any given window length, the Hann window and the Blackman-Harris family of windows satisfy the COLA constraint for certain values of M (See Appendix for the definition of these windows).[1]

To apply the Hann window in the OLA system, conduct the analysis part as in Fig. 1. After filtering and inverse DFT, the length of the output block may increase. It is important to align adjacent blocks correctly. General rules are:

- Keep the hopsize identical in analysis and in synthesis.
- Sum up wherever nearby blocks overlap. Sometimes more than three blocks would overlap during synthesis.

[1]Note that every window satisfies Eq. (1) for $M = 1$. This is considered a trivial case and is not of interests in this class.

3 Dual interpretations of short-time Fourier transform: OLA vs. FBS

Recall that the STFT is defined as follows [1],

$$X_m(\omega_k) = \sum_{n=-\infty}^{\infty} (w[n-m] \cdot x[n]) e^{-j\omega_k n}. \tag{2}$$

Below, we shall look at two different interpretations of the STFT.

3.1 The OLA interpretation

The OLA interpretation of Eq. (2) has been introduced in previous sections. To summarize, it involves two steps: (i) Apply a window to $x[n]$ around time m; (ii) Apply DFT to each windowed block of signal. Therefore, the STFT is viewed as *a series of time-varying spectra*. Each spectrum correspond to a block of signal in time, and blocks can overlap. For perfect reconstruction, the window and the hopsize need to satisfy the COLA constraint in Eq. (1).

3.2 The FBS interpretation

To understand the filter-bank summation (FBS) interpretation, we need to rewrite Eq. (2) as follows,

$$\begin{aligned} X_m(\omega_k) &= \sum_{n=-\infty}^{\infty} \left(x[n] e^{-j\omega_k n} \right) w[n-m] && (3) \\ &= \{x_k * \mathrm{FLIP}(w)\}[m], && (4) \end{aligned}$$

where $x_k[n] = x[n] \exp(-j\omega_k n)$, and $\mathrm{FLIP}(w[n]) = w[-n]$. Equation (4) leads to a new way of looking at STFT which involves two steps:

- Modulate $x[n]$ so as to shift its spectrum by $-\omega_k$.
- Apply an FIR filter whose impulse response is specified by $w[-n]$.

Therefore, $X_m(\omega_k)$ can be thought of as the output signal at time m for the k-th filter in the filterbank. The synthesis filter bank involves re-modulation to carrier frequencies (i.e., multiply by $e^{j\omega_k n}$) and summation over all channels.

The fact that we can hop the window by M and gets perfect reconstruction in time implies that we can down sample the filtered response by a factor of M and still get perfect reconstruction. Refer to [3] for further reading on FBS. The FBS architecture is commonly used in audio signal processing for medical applications, including hearing aids and cochlear implants. We will cover this as a special topic near the end of this semester.

4 Exercises

1. If the length of a filter is L taps in time, how much multiplication does it take to convolve a signal of length N with this filter in the time domain? How about in the frequency domain using FFT of length N?
2. Let's write the Hann window of length $N+1$ as $w[n] = \cos^2(\pi n/N)$ for $|n| \leq N/2$. By means of sketching, find out a hopsize $M > 1$ such that $w(n)$ satisfies Eq. (1).
3. Draw a block diagram for FBS analysis and synthesis.

A Commonly used windows

Last week, we introduced windows as a multiplier function in the short-time Fourier transform (STFT). In this section, let us look at a few commonly used windows and calculate their DTFTs.

A.1 The rectangular window

A rectangular window of length N can be defined as

$$w_R[n] = 1, n = 0, 1, 2, ..., N-1,$$

and $w_R[n] = 0,$ elsewhere. Its DTFT is

$$W_R(\omega) = e^{-j\omega(N-1)/2} \cdot \frac{\sin\left(N\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}. \tag{5}$$

Derivation:

Note the linear phase term $e^{-j\omega(N-1)/2}$. As we will see in more examples, this is a property for all window functions that look symmetric under horizontal flipping. (Why?)

A.1.1 zero-phase windows

We have defined the rectangular window from time $n = 0$ to $N-1$. **When N is an odd number**, it is easier to shift the window in time so it centers around time zero. Now we have a modified definition for the rectangular window:

$$w_R[n] = 1,\ n = -\frac{N-1}{2}, ..., 0, ...\frac{N-1}{2},$$

and

$$w_R[n] = 0,\ \text{elsewhere.}$$

When an even-symmetric window is centered around time zero, its DTFT becomes purely real. Therefore, such a window is called a *zero-phase* window for its phase is 0 at every frequency (or strictly speaking, the phase is either 0 or π). Now, the DTFT of $w_R[n]$ is simply

$$W_R(\omega) = \frac{\sin\left(N\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}. \tag{6}$$

The linear-phase term has disappeared. From here on, we will use the zero-phase definition for the rectangular window in favor of mathematical simplicity.

A.2 The Hann or raised-cosine window

In this section, we shall follow the MATLAB convention, which defines a Hann window of length N (usually an odd number) as

$$w[n] = \frac{1+\cos(\Omega_N n)}{2} \equiv \cos^2\left(\frac{\Omega_N}{2}n\right),\ n = -\frac{N-1}{2}, ..., 0, ...\frac{N-1}{2},$$

where $\Omega_N = 2\pi/(N-1)$ is 2π times the fundamental frequency of the raised cosine function. It is straightforward to check that $w[n]$ satisfies the COLA criteria (1) for the hopsize $M = (N-1)/2$, or any integer M that divides $N-1$ (!!).

Note that

$$w[n] = w_R[n] \cdot \frac{1+\cos(\Omega_N n)}{2}, \tag{7}$$

where $w_R[n]$ is a rectangular window of length $N-1$. Equation (7) can be written as

$$\begin{aligned} w[n] &= w_R[n] \cdot \frac{1+\cos(\Omega_N n)}{2} \\ &= w_R[n] \cdot \left(\frac{1}{2} + \frac{1}{4}e^{j\Omega_N n} + \frac{1}{4}e^{-j\Omega_N n}\right). \end{aligned}$$

Using the shift theorem and linearity, we obtain the DTFT of the Hann window:

$$W(\omega) = \frac{1}{2}W_R(\omega) + \frac{1}{4}W_R(\omega - \Omega_N) + \frac{1}{4}W_R(\omega + \Omega_N). \tag{8}$$

Exercise: Use Eqs. (6) and (8) to sketch the DTFT of the Hann window for $N = 17$. Note that the period of the raised-cosine function is $8+8 = 16 = 17-1$. In the frequency domain, what's the width of the mainlobe?

A.3 The Blackman-Harris family of windows

The Blackman-Harris family of windows are defined by considering a more general summation of shifted `sinc` functions in the frequency domain. In the time domain, a Blackman-Harris window $w[n]$ can be defined as follows,

$$w[n] = w_R[n] \sum_{k=0}^{K-1} a_k \cos(k\Omega_N n),$$

where the length (i.e., number of nonzero taps) of $w_R[n]$ is $N-1$, and $\Omega_N = 2\pi/(N-1)$. It is straightforward to show that $w[n]$'s DTFT is indeed a linear combination of shifted `sinc` function (Eq. 6).

In particular, choosing $K = 2$, and $(a_0, a_1, a_2) = (0.42, 0.5, 0.08)$ leads to what is called *the Blackman window* (MATLAB function `blackman()`). The Blackman window has two good properties: first, its sidelobe is at -58 dB relative to its mainlobe; to compare, the sidelobe is -31 dB for the Hann window. Secondly, its roll-off rate is about 18 dB/octave, same as that of the Hann window, but three times better than that of the rectangular window. These properties make the Blackman window useful for *spectral estimation* purposes (to be covered later in the semester). However, these good properties are achieved at the cost of increasing the width of the mainlobe.

Exercise: What's the mainlobe width of the Blackman window transform?

A.4 Notes on other types of windows

Over the years, many other types of windows have become popular each for different reasons. For instances,

- The Chebyshev window has equally suppressed sidelobes
- The Hann-Poisson window does not have sidelobes at all (!)
- The Kaiser family of windows have optimal mainlobe-to-sidelobes energy ratio

We can also see that some windows satisfy the *constant over-lap-add* (COLA) criterion but most other windows do not. This property alone makes them appealing for FFT-based audio analysis and synthesis. More interested readers can refer to MATLAB's help files to choose an appropriate window for different purposes:

```
WINDOW Window function gateway.
   WINDOW(@WNAME,N) returns an N-point window of type specified
by the function handle @WNAME in a column vector.

@blackman       - Blackman window.
@blackmanharris - Minimum 4-term Blackman-Harris window.
@hann           - Hann window.
@rectwin        - Rectangular window.
```

...

You are also encouraged to refer to Harris' pioneering paper [2] for further reading.

References

[1] Jont B. Allen and L. R. Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, Nov. 1977.

[2] F. J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, Jan 1978.

[3] M. R. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-24(3):243–248, June 1976.

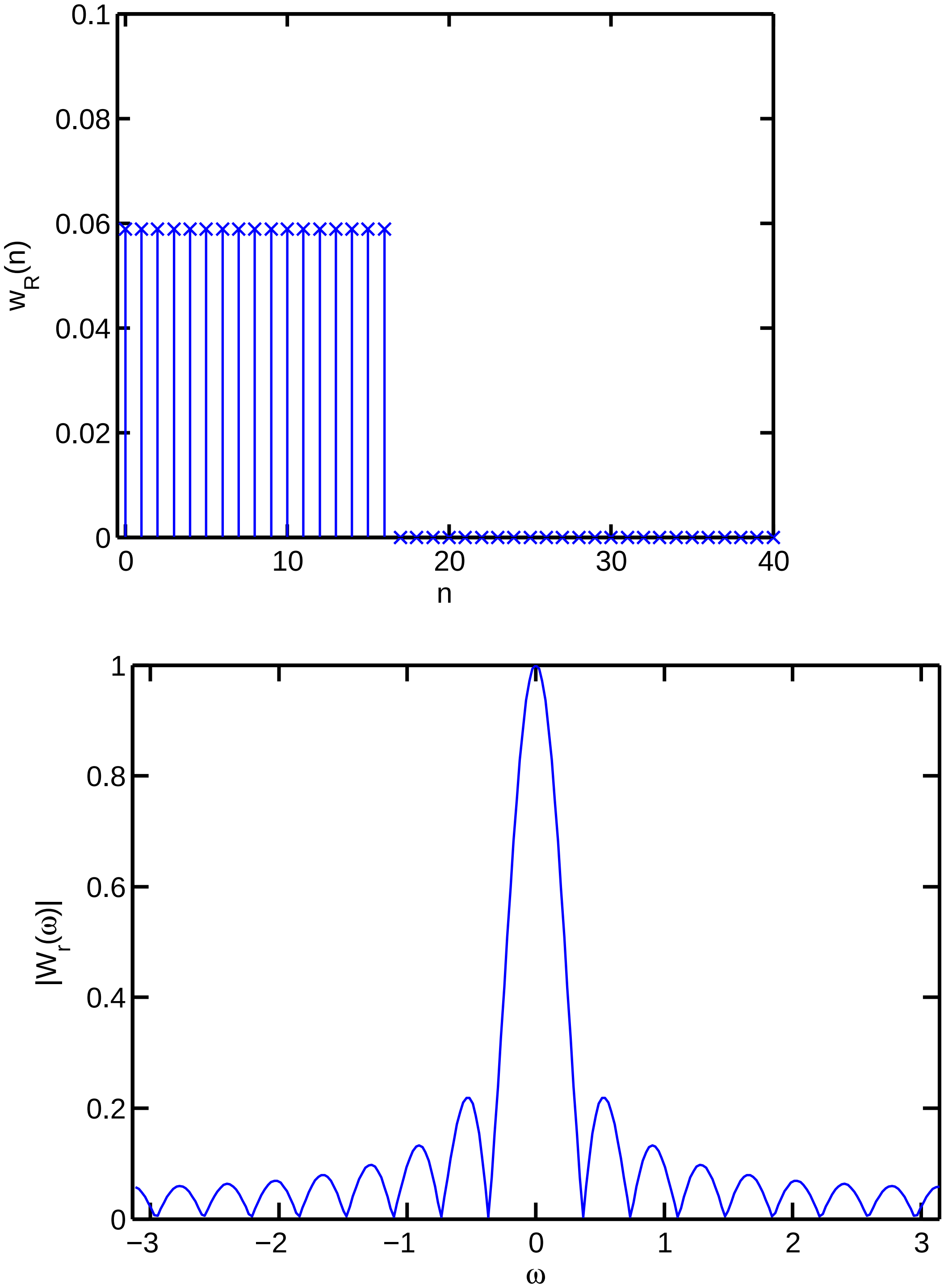

Figure 2: *Top*: the rectangular window of length 17. *Bottom*: its magnitude spectrum.