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

Yi-Wen Liu

Updated 12 Oct 2015

In previous lectures we covered convolution theorems; 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 of FFT is block-wise implementation of 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.

1 FIR filtering using the rectangular window

Suppose that we have an infinitely long signal $x[n]$ coming in, and we intend to convolve it with an FIR filter of length L. Assume that we will process $x[n]$ in a block-wise manner and the block length is N.

Then, the m^{th} block can be thought of as

$$x^{(m)}[n] = x[n + mN] \cdot w_R[n], \tag{1}$$

where $n = 0, 1, 2, ..., N-1$, and $w_R[n]$ denotes the rectangular window of length N,

$$w_R[n] = \begin{cases} 1, & 0 \le n < N, \\ 0, & \text{elsewhere.} \end{cases}$$

Note that we have the following identity:

$$\sum_{m=-\infty}^{\infty} w_R[n - mN] = 1. \tag{2}$$

This is an example of the *constant overlap-add* (COLA) condition and we will come back to it in a moment.

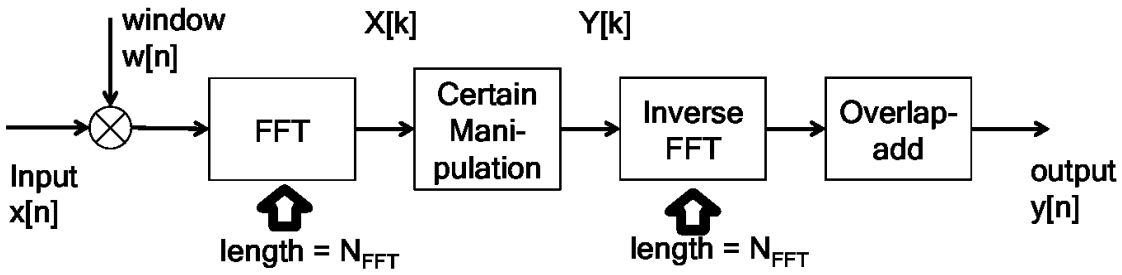

Figure 1: A general block diagram of OLA analysis and synthesis system

Now, to use FFT to convolve this block $x^{(m)}[n]$ with a finite impulse response $h[n], 0 \leq n < L$, we need to append zeros to the end of both $x^{(m)}[n]$ and $h[n]$ first. This operation is called *zero-padding*, and its purpose is to avoid *time-domain aliasing* due to cyclic convolution.

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?). For example, assume that the length of FFT is 1024. Denote the FFTs of $x^{(m)}[n]$ and $h[n]$ as $X^{(m)}[k]$ and $H[k]$ respectively, and $k = 0, 1, ..., 1023$. Then, the corresponding output block $y^{(m)}[n] = x^{(m)}[n] * h[n]$ can be given as follows,

$$y^{(m)}[n] = \frac{1}{1024} \sum_{k=0}^{1023} \left(X^{(m)}[k] H[k] \right) \exp\left(j\omega_k n \right), \tag{3}$$

where

$$\omega_k = k \frac{2\pi}{1024}.$$

Equation (3) means that the output $y^{(m)}[n]$ is the inverse DFT of $X^{(m)}[k]H[k]$ with length 1024. In this course, we will take it for granted that Eq. (3) can be accelerated using IFFT.

Finally, to put the blocks back together, we need to shift each block by multiples of N and then add; that is,

$$y[n] = \sum_{m=0}^{\infty} y^{(m)}(n - mN) = x[n] * h[n]. \tag{4}$$

Derivation of Eq. (4)?

2 The OLA synthesis

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

Remarks: Abrupt changes, or "discontinuities", are bad.

The constant overlap-add (COLA) criterion

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

$$\sum_{m=-\infty}^{\infty} w[n - mM] = C, \tag{5}$$

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

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

Exercise: for a Hann window of length 252, what are the possible M for COLA?

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. Following is a brief guideline.

- Determine a hopsize M that satisfies COLA for $w[n]$. Keep the hopsize identical in analysis and in synthesis.
- Use $w[n]$ in place of $w_R[n]$ in Eq. (1), so now the m^{th} block is defined as

$$x^{(m)}[n] = x[n + mM] \cdot w[n].$$

- Use Eq. (3) to synthesize the output in a blockwise manner.

[1]Note that every window satisfies Eq. (5) for $M = 1$. This is considered a trivial case and is not of interests in this class.

- Hop by M and sum up the output blocks; that is,

$$y[n] = \sum_{m=0}^{\infty} y^{(m)}[n - mM].$$

  Depending on the length of the filter, sometimes more than three blocks would overlap during synthesis.

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

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

$$X_m(\omega_k) = \sum_{n=-\infty}^{\infty} (w[n-m] \cdot x[n]) e^{-j\omega_k n}. \tag{6}$$

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

The OLA interpretation

The more straight-forward interpretation of Eq. (6) involves two steps:

- Apply a window to $x[n]$ near time m
- Apply DFT to each time-windowed 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.

To reconstruct the signal given its STFT, the window and the hopsize need to satisfy the COLA constraint in Eq. (5) during synthesis.

The FBS interpretation

To understand the filter-bank summation (FBS) interpretation, we need to rewrite Eq. (6) as follows,

$$\begin{aligned} X_m(\omega_k) &= \sum_{n=-\infty}^{\infty} \left(x[n]e^{-j\omega_k n}\right) w[n-m] && (7) \\ &= \{x_k * \mathrm{FLIP}(w)\}[m], && (8) \end{aligned}$$

where $x_k[n] = x[n]\exp(-j\omega_k n)$, and $\mathrm{FLIP}(w[n]) = w[-n]$.

Exercise: Let us draw a flow diagram for FBS.

Equation (8) 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.

Note that, in practice, most of the commonly used windows are symmetric; $w[-n] = w[n]$. Moreover, windows are smooth functions. So convolution with $w[-n]$ has a low-pass filtering effect for the modulated signals $x_k[n]$.

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 may cover this as a special topic near the end of this semester.

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.