

Lecture 16
Linear Filtering with the DFT

Reading: Sections 8.6 and 8.7 in Oppenheim, Schaffer & Buck (OSB).

Circular Convolution

$x[n]$ and $h[n]$ are two finite sequences of length N with DFTs denoted by $X[k]$ and $H[k]$, respectively. Let us form the product

$$W[k] = X[k]H[k],$$

and determine the sequence $w[n]$ of length N for which the DFT is $W[k]$.

First, extend $x[n]$ and $h[n]$ to periodic sequences with period N , $\tilde{x}[n]$ and $\tilde{h}[n]$, respectively. Then, the periodic convolution of $\tilde{x}[n]$ and $\tilde{h}[n]$ corresponds to multiplication of the corresponding periodic sequences of Fourier series coefficients (see OSB section 8.2.5).

$$\tilde{w}[n] = \sum_{m=0}^{N-1} \tilde{x}[m]\tilde{h}[n-m] \leftrightarrow \tilde{W}[k] = \tilde{X}[k]\tilde{H}[k]$$

The periodic sequence $\tilde{w}[n]$ for which the DFS coefficients are $\tilde{W}[k]$ corresponds to the periodic extension of the finite length sequence $w[n]$ with period N . We can recover $w[n]$ by extracting one period of $\tilde{w}[n]$:

$$\begin{aligned} w[n] &= \tilde{w}[n]\mathbf{R}_N[n] \\ &= \sum_{m=0}^{N-1} \tilde{x}[m]\tilde{h}[n-m] \\ &= \sum_{m=0}^{N-1} x[m]h[((n-m))_N] \end{aligned}$$

This operation is called circular convolution and denoted

$$w[n] = x[n] \circledast h[n].$$

Example:

Consider two constant sequences of length N , $x_1[n] = x_2[n]$, depicted in OSB Figure 8.15(a) and (b).

The N -point DFT of $x_1[n]$ is,

$$\begin{aligned} X_1[k] &= \sum_0^{N-1} x_1[n] W_N^{kn} = \sum_0^{N-1} W_N^{kn} = \frac{1 - W_N^{Nk}}{1 - W_N^k} = 0 & k \neq 0 \\ &= N & k = 0 \end{aligned}$$

where $W_N = e^{-j(2\pi/N)}$. Since $x_1[n] = x_2[n]$,

$$X_3[k] = X_1[k]X_2[k] = N^2\delta[k] = NX_1[k]$$

Using linearity, we can see that

$$\begin{aligned} x_3[n] = x_1[n] \circledast x_2[n] = Nx_1[n] &= N & 0 \leq n \leq N-1 \\ &= 0 & \text{otherwise.} \end{aligned}$$

The N -point circular convolution of $x_1[n]$ and $x_2[n]$ is depicted in OSB Figure 8.15(c).

Example:

Now, consider $x_1[n] = x_2[n]$ as $2L$ -point sequences by augmenting them with L zeros as shown in OSB Figure 8.16(a) and (b).

Performing a $2L$ -point circular convolution of the sequences, we get the sequence in OSB Figure 8.16(e), which is equal to the linear convolution of $x_1[n]$ and $x_2[n]$.

Circular Convolution as Linear Convolution with Aliasing

We know that convolution of two sequences corresponds to multiplication of the corresponding Fourier transforms:

$$y[n] = x[n] * h[n] \leftrightarrow Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

If we define a DFT

$$Y[k] = Y(e^{j(2\pi k/N)}), \quad 0 \leq k \leq N-1,$$

it follows that

$$Y[k] = X(e^{j(2\pi k/N)})H(e^{j(2\pi k/N)}) = X[k]H[k], \quad 0 \leq k \leq N-1.$$

From our definition of the circular convolution $w[n]$, $W[k] = X[k]H[k]$, so

$$W[k] = Y[k].$$

If $x[n]$ and $h[n]$ are sequences of length N , then $w[n]$ has length N , but $y[n]$ has the maximum length of $(2N-1)$. In order to calculate the N -point DFT of $y[n]$, we first form a periodic sequence of period N as follows:

$$\tilde{y}[n] = \sum_{r=-\infty}^{\infty} y[n - rN]$$

From the last lecture on the DFT, it follows that $Y[k]$ ($= W[k]$) is the DFT of one period of $\tilde{y}[n]$. Thus, the circular convolution of two finite-length sequences is equivalent to the linear convolution of the two sequences, followed by time aliasing.

$$\begin{aligned} w[n] = \tilde{y}[n]\mathbb{R}_N[n] &= \sum_{r=-\infty}^{\infty} y[n - rN] && 0 \leq n \leq N - 1 \\ &= 0 && \textit{otherwise} \end{aligned}$$

Example:

Consider two identical sequences $x_1[n]$ and $x_2[n]$ of length L in OSB Figure 8.18(a). The linear convolution of the two sequences is shown in OSB Figure 8.18(b). OSB Figure 8.18(c) and (d) show two of the shifted versions of (b). The L -point circular convolution of $x_1[n]$ and $x_2[n]$ is shown in OSB Figure 8.18(e), which can be formed by summing (b), (c), and (d) in the interval $0 \leq n \leq L - 1$.

Since the length of the linear convolution is $(2L-1)$, the result of the $2L$ -point circular convolution in OSB Figure 8.18(f) is identical to the result of linear convolution.

Now, consider two finite-duration sequences $x_1[n]$ and $x_2[n]$, with $x_1[n]$ of length L , and $x_2[n]$ of length $P < L$ as illustrated in OSB Figure 8.19. Let $x_3[n]$ be the linear convolution of $x_1[n]$ and $x_2[n]$.

To determine the L -point circular convolution $x_{3_p}[n]$, we use the time-aliasing interpretation:

$$\begin{aligned} x_{3_p}[n] = x_1[n] \textcircled{L} x_2[n] &= \sum_{r=-\infty}^{\infty} x_3[n - rL], && 0 \leq n \leq L - 1, \\ &= 0, && \textit{otherwise}. \end{aligned}$$

OSB Figure 8.20 shows the terms for (a) $r = 0$, (b) $r = -1$, and (c) $r = 1$. When the terms are summed to calculate the circular convolution in OSB Figure 8.20(d), the last $(P-1)$ points

of $x_3[n + L]$ will be added to the first $(P - 1)$ points of $x_3[n]$.

We can alternatively view the process of forming the circular convolution $x_{3_p}[n]$ as wrapping the linear convolution $x_3[n]$ around a cylinder of circumference L . As shown in OSB Figure 8.21, the first $(P - 1)$ points are corrupted by time aliasing, and the points from $n = P - 1$ to $n = L - 1$ are identical to the corresponding points of the linear convolution.

As shown in OSB Figure 8.21, it is clear that time aliasing in the circular convolution can be avoided if N , the length of the DFT, is larger than or equal to $(L + P - 1)$.

Block Convolution

In lecture 19, we will learn highly efficient algorithms for computing the DFT. Because of these algorithms, it is computationally efficient to implement a linear convolution of two sequences by computing the DFTs, multiplying them, and computing the IDFT. Since multiplying the DFTs corresponds to circular convolution of the corresponding sequences, we must avoid time aliasing to recover linear convolution from the result of the IDFT.

Consider an input data $x[n]$ of length N , and an FIR filter $h[n]$ of length P . The linear convolution of the two sequences has length $(N + P - 1)$. To avoid time aliasing, the DFT length must be at least $(N + P - 1)$. However, in many applications, such as filtering a speech waveform, the length of the input data is of indefinite duration as depicted in OSB Figure 8.22. Computing the DFT of the entire input signal in this case can be impractical, and will cause a long delay since we need all samples of the input before filtering. The solution is to use block convolution, in which the input signal is segmented into sections of length L . Then, we can use the DFT to convolve each section with the FIR, and get the desired linear convolution by fitting the filtered sections.

Overlap-add Method

First, segment the input signal into sections of L , and convolve each section with the FIR of length P . The linear convolution of one section of the input and the FIR will result in a sequence $y[n]$ of length $(L + P - 1)$. Therefore, we can use the DFT of length $(L + P - 1)$ to compute the convolution without time aliasing. As shown in OSB Figure 8.23, the nonzero points in the filtered sections will overlap by $(P - 1)$ points, and these overlap points should be added together to construct the output. This procedure is called overlap-add method.

Overlap-save Method

In the overlap-add method, after computing each section, we need to store $(P - 1)$ values of $y[n]$ and wait for the next data segment to add overlapped points. In cases this is not

desirable, we can use an alternative method, overlap-save method.

In OSB Figure 8.21, we saw that in a circular convolution not all points are corrupted by time aliasing. The first $(P - 1)$ points of each segment are time aliased, but we have $L - (P - 1) = (L - P + 1)$ points that are equal to the linear convolution. Therefore, as shown in OSB Figure 8.24, the portion of each output section in the region $0 \leq n \leq P - 2$ is discarded, and the remaining samples are saved to construct the final filtered output.