Examples

Example 4.3: Real-Valued Transforms

The "classic" definitions of the various discrete-time transforms we have just studied owe their elegance, simplicity and symmetry to the use of complex exponentials as the chosen basis functions. Nonetheless, and in particular for beginners who still need time to "interiorize" their structure, these transforms present an intuitive challenge in two major respects:

- the fact that the transform of a purely real signal is a complex signal
- the appearance of negative frequencies.

Although both facts "explain themselves away" as soon as one tinkers with the formulas a bit, it is probably instructive to explicitly derive alternative formulations for the Fourier transform of a discrete-time signal that do not involve complex exponentials. We will see that, although perfectly equivalent to the classic transforms, these alternative representations are more awkward to manipulate and more difficult to implement algorithmically.

A Real-Valued DFT. Consider an N-tap real signal $x[n]$ and its DFT. The symmetry relations (4.80) and (4.81) allow us to state the following:

- the real part of the DFT, being symmetric, is uniquely specified by half of the DFT values
- the imaginary part of the zero-th DFT coefficient is zero
- the imaginary part of the DFT, being antisymmetric, is uniquely specified half of the DFT values, minus one.

Let's therefore define a new transform that captures the information of the

DFT of a real signal. Call its coefficients $R[n], n = 0, \dots, N-1$; we have

$$
\begin{aligned}
R[0] &= X[0] &&= \sum_{n=0}^{N-1} x[n] \\
R[1] &= \mathrm{Re}\{X[1]\} &&= \sum_{n=0}^{N-1} x[n]\cos((2\pi/N)n) \\
&\dots \\
R[M] &= \mathrm{Re}\{X[M]\} &&= \sum_{n=0}^{N-1} x[n]\cos((2\pi/N)Mn) \\
R[M+1] &= \mathrm{Im}\{X[1]\} &&= \sum_{n=0}^{N-1} x[n]\sin((2\pi/N)n) \\
&\dots \\
R[N-1] &= \mathrm{Im}\{X[N-M]\} &&= \sum_{n=0}^{N-1} x[n]\sin((2\pi/N)(N-M)n)
\end{aligned}
\tag{4.1}
$$

where M is meant to indicate more or less "half the number of coefficients in the DFT" – we'll consider its exact value in just a moment. In the meantime, let's notice that we can split the above list into three types of coefficients:

(a) the sum of all values of $x[n]$ (i.e., the first coefficient),

(b) $C = M$ "cosine" coefficients,

(c) $S = N - M - 1$ "sine" coefficients.

Now the first inconveniences begin: unfortunately the value of M changes according to whether N is even or odd. We have

$$
\begin{aligned}
M &= (N-1)/2 \qquad &&\text{for } N \text{ odd} \\
M &= N/2 &&\text{for } N \text{ even}
\end{aligned}
\tag{4.2}
$$

so that we need to differentiate between the two cases in an algorithmic implementation of the transform. It is easy to derive that

$$
\begin{aligned}
C &= \lfloor N/2 \rfloor \\
S &= \lfloor (N-1)/2 \rfloor
\end{aligned}
$$

so that, in particular, for even-length signals, the number of cosine coefficients will exceed by one that of the sine coefficients.

The new transform is therefore defined by can be encoded by the following $N \times N$ matrix:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \cos(\omega_N) & \cos(2\omega_N) & \dots & \cos((N-1)\omega_N) \\ 1 & \cos(2\omega_N) & \cos(4\omega_N) & \dots & \cos(2(N-1)\omega_N) \\ & & & \dots & \\ 1 & \cos(C\omega_N) & \cos(2C\omega_N) & \dots & \cos((N-1)C\omega_N) \\ 0 & \sin(\omega_N) & \sin(2\omega_N) & \dots & \sin((N-1)\omega_N) \\ & & & \dots & \\ 0 & \sin(S\omega_N) & \sin(2S\omega_N) & \dots & \sin((N-1)S\omega_N) \end{bmatrix} \tag{4.3}$$

where of course $\omega_N = 2\pi/N$. All we need to do in order to verify the validity of the transform is show that $\mathbf{R}$ does indeed represent a change of basis in $\mathbb{R}^N$ or, in other words, that $\mathbf{R}\mathbf{R}^T$ is diagonal. That can be done in a variety of ways, all of which are extremely tedious since they involve trigonometric summations[1]. As a quick check, one can perform the multiplication in Matlab and be satisfied with a numerical answer. Although the matrix defines an orthogonal transform, it is not orthonormal and it turns out that the normalizing factors depend both on the coefficient's index *and* on whether N is even or odd. The details are left as an easy, if unrewarding, exercise.

Now that we have a bona-fide transform, let's try to understand the meaning of each coefficient – we know that the best way to do so is to consider the reconstruction formula. Remember that in the DFT, the physical interpretation is that of a bank of N complex exponential generators at all frequencies multiple of $2\pi/N$; each generator is scaled in amplitude by the corresponding DFT coefficient's magnitude and set to start with an offset equal to the phase of the same coefficient. Here, we have a potentially unequal set of real sinusoidal generators at only *positive* multiples of $2\pi/N$.; with possible exception of the last frequency, these are *quadrature* generators, i.e., each frequency is output in phase (the cosine component) and in quadrature (the sine component). The relative amplitude of the corresponding sine

[1] Probably the easiest way is to transform the sines and cosines in $\mathbf{R}$ into... complex exponentials!

and cosine coefficients determines the "shape" of the waveform output by the generator at frequency $(2\pi/N)k$, for $k = 0, \dots, C$. The sum of all these waveforms at positive frequencies gives back the original signal.

Although no information has been lost, with respect to the DFT the ease of use of the real-valued transform is diminished in these respects:

- we need to differentiate between even and odd lengths
- the energy associated to each frequency is split between two different coefficients'; e.g., the energy for $\omega = (2\pi/n)k$, for $k > 0$, is $R[k]^2 + R[C+k]^2$. As a consequence, visual inspection of the coefficients' magnitude (seen as a sequence in $\mathbb{R}^N$) is much less effective.
- the structure of the analysis and synthesis formulas is no longer symmetric and therefore its algorithmic implementations are more difficult to optimize.

The Discrete Cosine Transform. Another approach leading to a real-valued transform begins with the observation that, even if a signal $x[n]$ is smooth, the N-periodic signal $\tilde{x}[n]$ implicitly defined by its DFT may exhibit large jumps at the periodization boundaries. Indeed, even if the difference between successive samples is not too big, since $\tilde{x}[N-1]$ and $\tilde{x}[N] = \tilde{x}[0]$ are values far apart, the jump at the boundary may be large, as shown in Figure 4.3. Now, given a length-N signal $x[n]$, we can always build a symmetric length-$2N$ signal as

$$x_2[n] = \begin{cases} x[n] & \text{for } 0 \le n \le N-1 \\ x[2N-n-1] & \text{for } N \le n \le 2N-1 \end{cases}$$

i.e., we stick the mirror image of the signal after the signal itself. The result is even-length and symmetric around $N - 1/2$ and its $2N$-periodization will be smooth at the periodization boundaries as well, as shown in Figure 4.3.

Since $x_2[n]$ is defined by just N values, it stands to reason that its DFT will

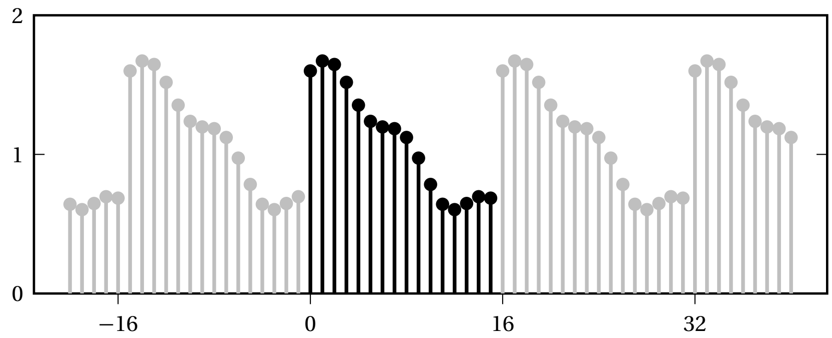

Figure 4.1: Boundary "discontinuities" caused by straight periodization ($N = 16$).

be uniquely determined by only N values as well. Indeed,

$$\begin{aligned}
X_2[k] &= \sum_{n=0}^{2N-1} x_2[n] e^{-j\frac{\pi}{N}nk} \\
&= \sum_{n=0}^{N-1} x[n](e^{-j\frac{\pi}{N}nk} + e^{-j\frac{\pi}{N}(2N-n-1)k}) \\
&= \sum_{n=0}^{N-1} x[n](e^{-j\frac{\pi}{N}nk} + e^{j\frac{\pi}{N}(n+1)k}) \\
&= e^{j\frac{\pi}{N}\frac{k}{2}} \sum_{n=0}^{N-1} x[n](e^{-j\frac{\pi}{N}(n+\frac{1}{2})k} + e^{j\frac{\pi}{N}(n+\frac{1}{2})k}) \\
&= 2e^{j\frac{\pi}{2}Nk} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right) \\
&= 2e^{j\frac{\pi}{2N}k} C[k]
\end{aligned}$$

and it is easy to verify that $C[N] = 0$ and that $C[k] = C[2N-k]$ for $k = 1, \ldots, N-1$. Thus, the DFT of the mirrored, $2N$-periodized signal is uniquely specified (via a scalar factor $2e^{j\frac{\pi}{2N}k}$) by just N coefficients $C[k]$, which are called the Discrete Cosine Transform (DCT) of the length-N signal $x[n]$. The

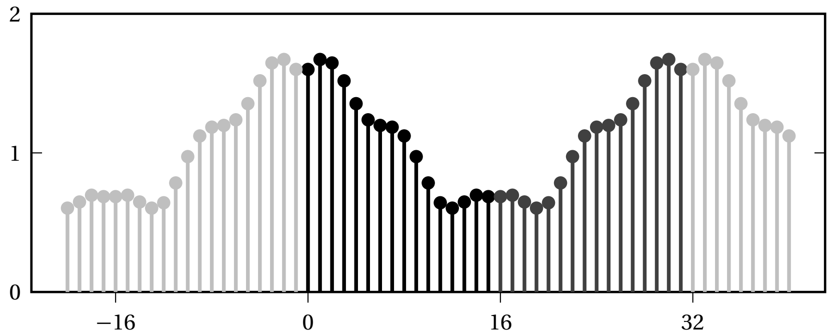

Figure 4.2: Mirroring eliminates discontinuities.

DCT is a full-fledged transform in $\mathbb{R}^N$ and its analysis and synthesis formulas are:

$$
\begin{aligned}
C[k] &= \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right) \\
x[n] &= \frac{1}{N}C[0] + \frac{2}{N}\sum_{k=1}^{N-1} C[k] \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)
\end{aligned}
$$

The DCT therefore defines another Fourier-like real-valued transform for real-valued sequences; unfortunately, the somewhat more involved underlying periodization and the resulting asymmetry between the analysis and synthesis formulas make the DCT difficult to use in theoretical derivations. On the practical front, however, the smoothness achieved by the mirroring at the signal's boundaries results in a set of DCT coefficients with a much better *energy compaction* property. In simple terms, this means that with respect to the DFT, no representational power is wasted to encode the artificial discontinuity created by the DFT periodization and that therefore fewer DCT coefficients are needed to approximate the original signal. This approximation power is particularly important in compression application and that's why the DCT is the transform of choice in many image encoding schemes.