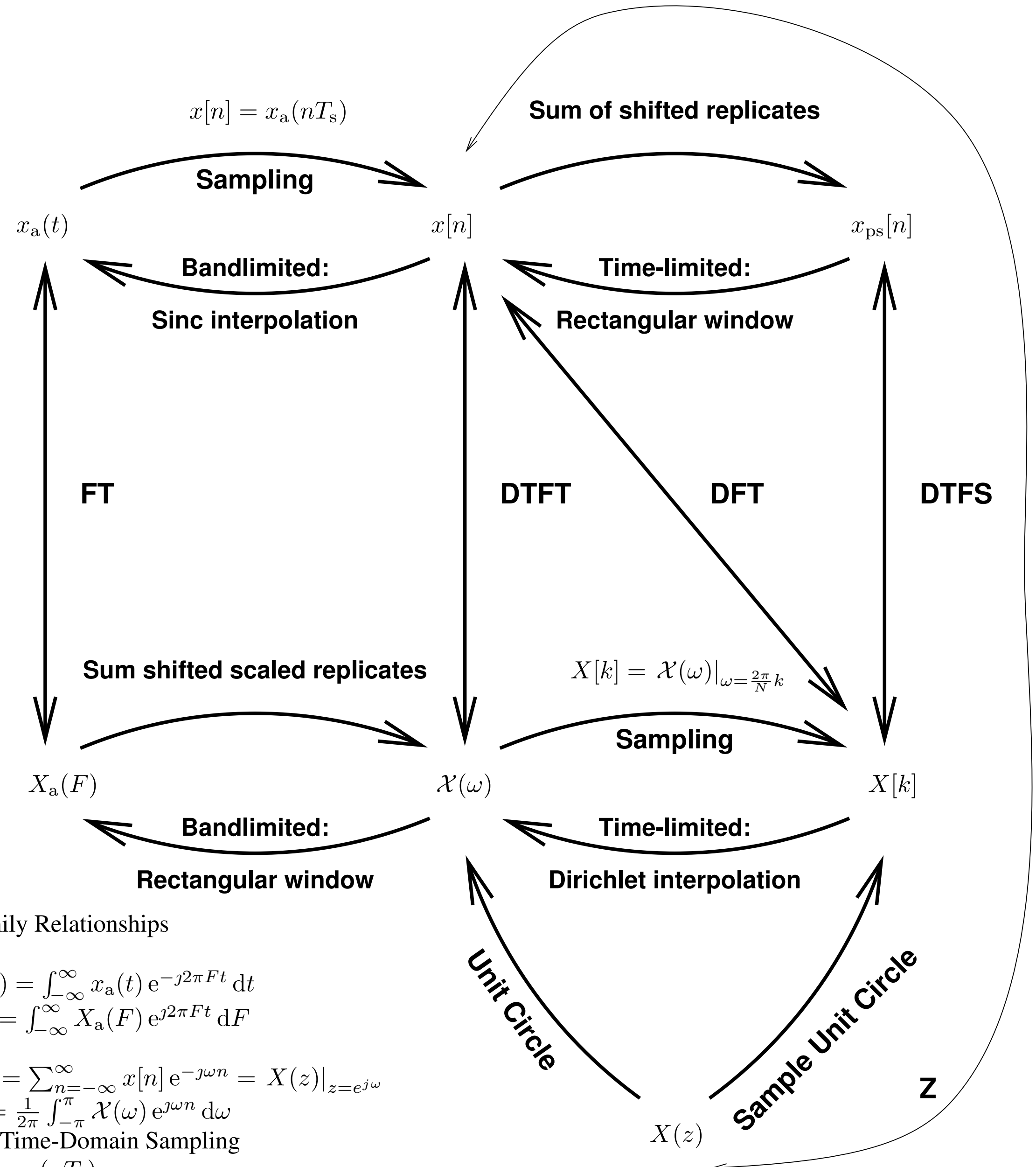
Chapter 5

The Discrete Fourier Transform

Contents



The FT Family Relationships

- FT
  - $X_{\mathrm{a}}(F) = \int_{-\infty}^{\infty} x_{\mathrm{a}}(t)\, \mathrm{e}^{-\jmath 2\pi F t} \,\mathrm{d}t$
  - $x_{\mathrm{a}}(t) = \int_{-\infty}^{\infty} X_{\mathrm{a}}(F)\, \mathrm{e}^{\jmath 2\pi F t} \,\mathrm{d}F$
- DTFT
  - $\mathcal{X}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, \mathrm{e}^{-\jmath\omega n} = X(z)|_{z=\mathrm{e}^{\jmath\omega}}$
  - $x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{X}(\omega)\, \mathrm{e}^{\jmath\omega n} \,\mathrm{d}\omega$
- Uniform Time-Domain Sampling
  - $x[n] = x_{\mathrm{a}}(nT_{\mathrm{s}})$
  - $\mathcal{X}(\omega) = \frac{1}{T_{\mathrm{s}}} \sum_{k=-\infty}^{\infty} X_{\mathrm{a}}\left(\frac{\omega/(2\pi)-k}{T_{\mathrm{s}}}\right)$ (sum of shifted scaled replicates of $X_{\mathrm{a}}(\cdot)$)
  - Recovering $x_{\mathrm{a}}(t)$ from $x[n]$ for bandlimited $x_{\mathrm{a}}(t)$, where $X_{\mathrm{a}}(F) = 0$ for $|F| \geq F_{\mathrm{s}}/2$
    - $X_{\mathrm{a}}(F) = T_{\mathrm{s}} \operatorname{rect}\left(\frac{F}{F_{\mathrm{s}}}\right) \mathcal{X}(2\pi F T_{\mathrm{s}})$ (rectangular window to pick out center replicate)
    - $x_{\mathrm{a}}(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t-nT_{\mathrm{s}}}{T_{\mathrm{s}}}\right)$, where $\operatorname{sinc}(x) = \sin(\pi x)/(\pi x)$. (sinc interpolation)

- DTFS
  - $c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_{\mathrm{ps}}[n]\, \mathrm{e}^{-\jmath\frac{2\pi}{N}kn} = \frac{1}{N}\, X(z)|_{z=\mathrm{e}^{\jmath\frac{2\pi}{N}k}}$
  - $x_{\mathrm{ps}}[n] = \sum_{k=0}^{N-1} c_k\, \mathrm{e}^{\jmath\frac{2\pi}{N}kn}$, $\omega_k = \frac{2\pi}{N}k$
- Uniform Frequency-Domain Sampling
  - $X[k] = \mathcal{X}(\omega)\Big|_{\omega=\frac{2\pi}{N}k}$, $k = 0,\ldots,N-1$
  - $X[k] = Nc_k$
  - $x_{\mathrm{ps}}[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, \mathrm{e}^{\jmath\frac{2\pi}{N}kn}$
  - $x_{\mathrm{ps}}[n] = \sum_{l=-\infty}^{\infty} x[n-lN]$ (sum of shifted replicates of $x[n]$)
  - Recovering $x[n]$ from $X[k]$ for time-limited $x[n]$, where $x[n] = 0$ except for $n = 0, 1, \ldots, L-1$ with $L \leq N$
    - $x[n] = x_{\mathrm{ps}}[n],\ n = 0,\ldots,L-1$, 0 otherwise. (discrete-time rectangular window)
    - $\mathcal{X}(\omega)$ related to $X[k]$ by Dirichlet interpolation: $\mathcal{X}(\omega) = \sum_{k=0}^{N-1} X[k]\, P(\omega - 2\pi k/N)$, where $P(\omega) = \frac{1}{N}\sum_{k=0}^{N-1} \mathrm{e}^{-\jmath\omega n}$.

Overview

Why *yet another* transform? After all, we now have FT tools for periodic and aperiodic signals in both CT and DT! What is left?

One of the most important properties of the DTFT is the convolution property: $y[n] = h[n] * x[n] \overset{\text{DTFT}}{\leftrightarrow} \mathcal{Y}(\omega) = \mathcal{H}(\omega)\, \mathcal{X}(\omega)$. This property is useful for analyzing linear systems (and for filter design), and also useful for "on paper" convolutions of two sequences $h[n]$ and $x[n]$, since if the sequences are simple ones whose DTFTs are known or are easily determined, we can simply multiply the two transforms and then "look up" the inverse transform to get the convolution.

What if we want to automate this procedure using a computer? Right away there is a problem since ω is a continuous variable that runs from $-\pi$ to π, so it looks like we need an (uncountably) infinite number of ω's which cannot be done on a computer.

For example, we cannot implement the ideal lowpass filter digitally.

This chapter exploit what happens if we do not use all the ω's, but rather just a finite set (which can be stored digitally). In general this will entail irrecoverable information loss. Fortunately, not always though! (Otherwise DSP would be a more academic subject.)

Any signal that is stored in a computer must be a **finite length sequence**, say $x[0], x[1], \ldots, x[L-1]$. Since there are only L signal time samples, it stands to reason that we should not need an infinite number of frequencies to adequately represent the signal. In fact, exactly $N \geq L$ frequencies should be enough information.

(We will see when we discuss zero-padding that for some purposes $N \approx 2L$ is an appropriate number of frequencies.)

Main points

- By the end of Chapter 5, we will know (among other things) how to use the DFT to convolve two generic sampled signals stored in a computer. By the end of Ch. 6, we will know that by using the FFT, this approach to convolution is generally *much* faster than using direct convolution, such as MATLAB's `conv` command.
- Using the DFT via the FFT lets us do a FT (of a finite length signal) to examine signal frequency content. (This is how digital spectrum analyzers work.)

Chapter 3 and 4 especially focussed on DT *systems*. Now we focus on DT signals for a while.

The **discrete Fourier transform** or **DFT** is the transform that deals with a finite discrete-time signal and a finite or discrete number of frequencies.

Which frequencies?

$$\boxed{\omega_k = \frac{2\pi}{N} k, \qquad k = 0, 1, \ldots, N-1.}$$

For a signal that is time-limited to $0, 1, \ldots, L-1$, the above $N \geq L$ frequencies contain all the information in the signal, *i.e.*, we can recover $x[n]$ from $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$.

However, it is also useful to see what happens if we throw away all but those N frequencies even for general aperiodic signals.

Discrete-time Fourier transform (DTFT) review

Recall that for a general aperiodic signal $x[n]$, the DTFT and its inverse is

$$\mathcal{X}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, \mathrm{e}^{-\jmath\omega n}, \qquad x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{X}(\omega)\, \mathrm{e}^{\jmath\omega n}\, \mathrm{d}\omega\,.$$

Discrete-time Fourier series (DTFS) review

Recall that for a N-periodic signal $x[n]$,

$$x[n] = \sum_{k=0}^{N-1} c_k\, \mathrm{e}^{\jmath \frac{2\pi}{N} kn} \quad \text{where} \quad c_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N} kn}\,.$$

Definition(s)

The N**-point DFT** of *any* signal $x[n]$ is defined as follows:

$$X[k] \triangleq \begin{cases} \sum_{n=0}^{N-1} x[n]\, e^{-\jmath \frac{2\pi}{N} kn}, & k = 0, \ldots, N-1 \\ ?? & \text{otherwise.} \end{cases}$$

Almost all books agree on the top part of this definition. (An exception is the 206 textbook (DSP First), which includes a $\frac{1}{N}$ out front to make the DFT match the DTFS.)
But there are several possible choices for the "??" part of this definition.

1. Treat $X[k]$ as an N-periodic function that is defined for all integer arguments $k \in \mathbb{Z}$.
   This is reasonable mathematically since
   $$X[n+N] = \sum_{n=0}^{N-1} x[n]\, e^{-\jmath \frac{2\pi}{N}(k+N)n} = \sum_{n=0}^{N-1} x[n]\, e^{-\jmath(\frac{2\pi}{N}kn + 2\pi kn)} = \sum_{n=0}^{N-1} x[n]\, e^{-\jmath \frac{2\pi}{N} kn} = X[k]\,.$$
2. Treat $X[k]$ as *undefined* for $k \notin \{0, \ldots, N-1\}$.
   This is reasonable from a practical perspective since in a computer we have subroutines that take an N-point signal $x[n]$ and return only the N values $X[0], \ldots, X[N-1]$, so trying to evaluate an expression like "$X[-k]$" will cause an error in a computer.
3. Treat $X[k]$ as being *zero* for $k \notin \{0, \ldots, N-1\}$.
   This is a variation on the previous option.

The book seems to waver somewhat between the first two conventions.

These lecture notes are based on the middle convention: that the N-point DFT is *undefined* except for $k \in \{0, \ldots, N-1\}$. This choice is made because it helps prevent computer programming errors.

Given $X[k]$ for $k \in \{0, \ldots, N-1\}$, the N**-point inverse DFT** is defined as follows:

$$\tilde{x}[n] = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{\jmath \frac{2\pi}{N} kn}, & n = 0, \ldots, N-1 \\ ??, & \text{otherwise.} \end{cases}$$

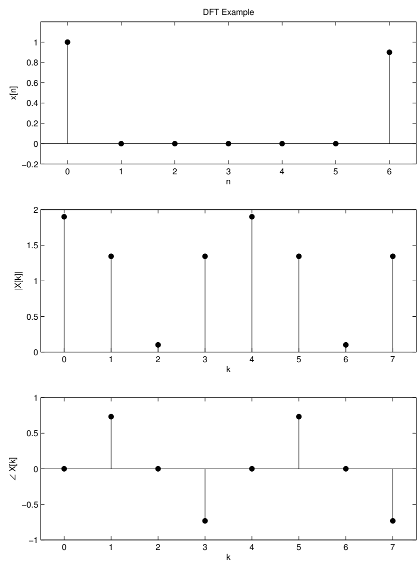
Here the natural choice for the "??" part depends on the type of signal is under consideration.

- If $x[n]$ is a **finite length** signal, supported on $0, \ldots, L-1$, where $L \leq N$, then we interpret the inverse DFT as
  $$x[n] = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{\jmath \frac{2\pi}{N} kn}, & n = 0, \ldots, N-1 \\ 0, & \text{otherwise.} \end{cases}$$
  This definition is the most important one since our primary use of the DFT is for length L signals with $L \leq N$.
  In this case the "inverse" is named appropriately, since we really do recover $x[n]$ exactly from $\{X[k]\}_{k=0}^{N-1}$. The proof of this is essentially identical to the proof given for the self-consistency of the DTFS.
- If $x[n]$ is a N-periodic signal, then we really should use the DTFS instead of the DFT, but they are so incredibly similar that sometimes we will use the DFT, in which case we should interpret the inverse DFT as follows
  $$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{\jmath \frac{2\pi}{N} kn}\,.$$
  This is indeed a N-periodic expression.
- If $x[n]$ is a signal whose length *exceeds* N, *e.g.*, if $x[n]$ is a aperiodic infinitely long signal, then the inverse DFT is best expressed
  $$x_{\rm ps}[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{\jmath \frac{2\pi}{N} kn}\,,$$
  where $x_{\rm ps}[n] \neq x[n]$ in this case. (Nevertheless, it may be that $x_{\rm ps}[n] \approx x[n]$ for $n = 0, \ldots, N-1$ so the DFT can be useful even in this case.

Examples

Example. $x[n] = \delta[n] + 0.9\,\delta[n-6]$. What is L? $L = 7$. Let us use $N = 8$. (Powers of 2 are handy later for FFTs.)

$$X[k] = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\jmath \frac{2\pi}{N} kn} = \sum_{n=0}^{7} x[n]\,\mathrm{e}^{-\jmath \frac{2\pi}{8} kn} = \sum_{n=0}^{7} (\delta[n] + 0.9\,\delta[n-6])\, e^{-j2\pi kn/8} = 1 + 0.9\,\mathrm{e}^{-\jmath \frac{2\pi}{8} k6}.$$



Alternative approach to finding $X[k]$. First find $X(z)$, then sample around unit circle. $X(z) = 1 + 0.9z^{-6}$, so

$$X[k] = X(z)\Big|_{z=\mathrm{e}^{\jmath \frac{2\pi}{N} k}} = 1 + 0.9\,\mathrm{e}^{-\jmath \frac{2\pi}{8} k}.$$

Example. Find N-point inverse DFT of $\{X[k]\}_{k=0}^{N-1}$ where $X[k] = \begin{cases} 1, & k = k_0 \\ 0, & \text{otherwise} \end{cases} = \delta[k - k_0]$, for $k_0 \in \{0, \ldots, N-1\}$. ***Picture*** .

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\,\mathrm{e}^{\jmath \frac{2\pi}{N} kn} = \frac{1}{N}\,\mathrm{e}^{\jmath \frac{2\pi}{N} k_0 n}.$$

Thus we have the following important DFT pair.

$$\boxed{\text{If } k_0 \in \{0, \ldots, N-1\}\text{, then } \frac{1}{N}\,\mathrm{e}^{\jmath \frac{2\pi}{N} k_0 n} \overset{\mathrm{DFT}}{\underset{N}{\longleftrightarrow}} \delta[k - k_0].}$$

Example. Find N-point inverse DFT of $\{X[k]\}_{k=0}^{N-1}$ where

$$X[k] = \begin{cases} \mathrm{e}^{\jmath\phi}, & k = k_0 \\ \mathrm{e}^{-\jmath\phi}, & k = N - k_0 \\ 0, & \text{otherwise,} \end{cases} = \mathrm{e}^{\jmath\phi}\,\delta[k - k_0] + \mathrm{e}^{-\jmath\phi}\,\delta[k - (N - k_0])$$

for $k_0 \in \{1, \ldots, N/2 - 1, N/2 + 1, \ldots, N - 1\}$. ***Picture*** .

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\,\mathrm{e}^{\jmath \frac{2\pi}{N} kn} = \frac{1}{N}\,\mathrm{e}^{\jmath\phi}\,\mathrm{e}^{\jmath \frac{2\pi}{N} k_0 n} + \frac{1}{N}\,\mathrm{e}^{-\jmath\phi}\,\mathrm{e}^{\jmath \frac{2\pi}{N} (N-k_0) n} = \frac{2}{N} \cos\!\left(\frac{2\pi}{N} k_0 n + \phi\right).$$

Example. Find the 8-point DFT of the signal $x[n] = 6\cos^2\left(\frac{\pi}{4}n\right)$.

Expanding: $x[n] = 3 + 3\cos\left(\frac{\pi}{2}n\right) = 3 + \frac{3}{2}\,\mathrm{e}^{\jmath\frac{2\pi}{8}2n} + \frac{3}{2}\,\mathrm{e}^{-\jmath\frac{2\pi}{8}2n} = \frac{1}{8}\left[24 + 12\,\mathrm{e}^{\jmath\frac{2\pi}{8}2n} + 12\,\mathrm{e}^{\jmath\frac{2\pi}{8}(8-2)n}\right]$.
So by **coefficient matching**, we see that $X[k] = \{\underline{24}, 0, 12, 0, 0, 0, 12, 0\}$.

Example. Complex exponential signal with frequency that is an integer multiple of $\frac{2\pi}{N}$.
Suppose $x[n] = \mathrm{e}^{\jmath\frac{2\pi}{N}k_0 n} = \mathrm{e}^{\jmath\omega_0 n}$ for $n = 0, \ldots, N-1$, where $\omega_0 = \frac{2\pi}{N}k_0$ and k_0 is an integer.
Find the N-point DFT of $x[n]$.

$$X[k] = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} \mathrm{e}^{\jmath\frac{2\pi}{N}k_0 n}\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} \mathrm{e}^{-\jmath\frac{2\pi}{N}(k-k_0)n} = \begin{cases} N, & k = k_0 + lN, l \in \mathbb{Z} \\ 0, & \text{otherwise.} \end{cases}$$

Thus

$$\boxed{x[n] = \mathrm{e}^{\jmath\frac{2\pi}{N}k_0 n} \overset{\mathrm{DFT}}{\underset{N}{\longleftrightarrow}} X[k] = N\sum_{l=-\infty}^{\infty}\delta[k - k_0 - lN] = N\,\delta_N[k - k_0],}$$

where $\delta_N[n] \triangleq \sum_{l=-\infty}^{\infty}\delta[n - lN]$.

Example. Complex exponential signal with frequency that is *not* an integer multiple of $2\pi/N$.
Suppose $x[n] = \mathrm{e}^{\jmath\omega_0 n}$ for $n = 0, \ldots, N-1$, where $\omega_0 \neq \frac{2\pi}{N}k_0$ for *any* integer k.
Find the N-point DFT of $x[n]$.

$$X[k] = \sum_{n=0}^{N-1} \mathrm{e}^{\jmath\omega_0 n}\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1}\left(\mathrm{e}^{\jmath(\omega_0 - \frac{2\pi}{N}k)}\right)^n = \frac{1 - \left(\mathrm{e}^{\jmath(\omega_0 - \frac{2\pi}{N}k)}\right)^N}{1 - \mathrm{e}^{\jmath(\omega_0 - \frac{2\pi}{N}k)}} = \frac{1 - \mathrm{e}^{\jmath\omega_0 N}}{1 - \mathrm{e}^{\jmath(\omega_0 - \frac{2\pi}{N}k)}}.$$

Thus we have the following curious DFT pair.

$$\boxed{\text{If } \omega_0/\tfrac{2\pi}{N} \text{ is non-integer, then } \mathrm{e}^{\jmath\omega_0 n} \overset{\mathrm{DFT}}{\underset{N}{\longleftrightarrow}} X[k] = \frac{1 - \mathrm{e}^{\jmath\omega_0 N}}{1 - \mathrm{e}^{\jmath(\omega_0 - \frac{2\pi}{N}k)}}.}$$

What is going on in these examples?

Let $s[n] = \mathrm{e}^{\jmath\omega_0 n}$ be an eternal complex exponential signal, and define the following **rectangular window**

$$r_N[n] = \begin{cases} 1, & n = 0, \ldots, N-1 \\ 0, & \text{otherwise,} \end{cases}$$

which has the following DTFT:

$$\mathcal{R}(\omega) = \sum_{n=0}^{N-1}\mathrm{e}^{-\jmath\omega n} = \cdots = \mathrm{e}^{-\jmath\omega(N-1)/2}\,\mathcal{R}_r(\omega), \text{ where } \mathcal{R}_r(\omega) = \begin{cases} \dfrac{\sin(\omega N/2)}{\sin(\omega/2)}, & \omega \neq 0 \\ N, & \omega = 0 \end{cases} \approx N\,\mathrm{sinc}\left(N\frac{\omega}{2\pi}\right).$$

Then we have

$$x[n] = s[n]\,r_N[n] \Longrightarrow \mathcal{X}(\omega) = S(\omega) * R_N(\omega) = 2\pi\,\delta(\omega - \omega_0) * \mathcal{R}(\omega) = 2\pi\,\mathcal{R}(\omega - \omega_0),$$

where the above δ is a Dirac impulse. Thus

$$X[k] = \mathcal{X}(\omega)\Big|_{\omega = \frac{2\pi}{N}k} = 2\pi\,\mathcal{R}\left(\frac{2\pi}{N}k - \omega_0\right).$$

When $\omega_0 = \frac{2\pi}{N}k_0$, then the sinc is sampled only at the peak and the nulls, which gives the Dirac impulse form above.
Otherwise the sinc is sampled at many nonzero values, which gives the messy form above.

DTFT sampling preview

The DTFT formula is $\mathcal{X}(\omega) = \sum_{n=-\infty}^{\infty} x[n] \, \mathrm{e}^{-\jmath \omega n}$ whereas the DFT analysis formula is $X[k] = \sum_{n=0}^{N-1} x[n] \, \mathrm{e}^{-\jmath \frac{2\pi}{N} kn}$.

If $x[n]$ is a L**-point** signal, *i.e.*, it is nonzero only for $n = 0, 1, \ldots, L-1$, then the DTFT "simplifies" to $\mathcal{X}(\omega) = \sum_{n=0}^{L-1} x[n] \, \mathrm{e}^{-\jmath \omega n}$.

Comparing these two formulas leads to the following conclusion.

If $x[n]$ is a L-point signal with $L \leq N$, then the N-point DFT values are samples of the DTFT:

$$X[k] = \mathcal{X}(\omega) \Big|_{\omega = \frac{2\pi}{N} k} .$$

If we are given a DTFT $\mathcal{X}(\omega)$, and wish to find $x[n]$, then the "usual" approach would be to apply the inverse DTFT, *i.e.*, the **DTFT synthesis formula**: $x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{X}(\omega) \, \mathrm{e}^{\jmath \omega n} \, \mathrm{d}\omega$.
However, performing this integral can be inconvenient.
The above relationship between the DFT and the DTFT suggests the following easier approach.

- First sample the DTFT $\mathcal{X}(\omega)$ to get DFT values $X[k]$, $k = 0, \ldots, N-1$.
- Then take the inverse DFT of $X[k]$ (using the inverse FFT) to get (hopefully) the signal $x[n]$.

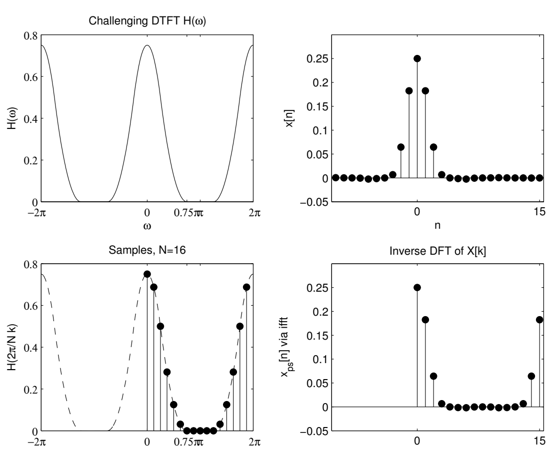
Does this approach always work? No!
Why not? Because the DFT/DTFT relationship holds only if $x[n]$ is an L-point signal with $L \leq N$.

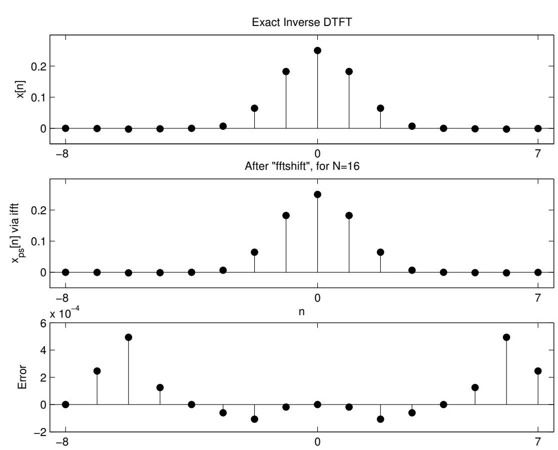
Example. Find the signal $x[n]$ that has the following spectrum, with $\omega_0 = \pi/2$.

$$\mathcal{X}(\omega) = \begin{cases} \frac{3}{4} - \left|\frac{\omega}{\omega_0}\right|^2, & \left|\frac{\omega}{\omega_0}\right| \leq \frac{1}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \left|\frac{\omega}{\omega_0}\right|\right)^2, & \frac{1}{2} < \left|\frac{\omega}{\omega_0}\right| \leq \frac{3}{2} \\ 0, & \frac{3}{2} < \left|\frac{\omega}{\omega_0}\right| \leq \frac{\pi}{\omega_0} \\ \text{periodic} & \text{otherwise.} \end{cases}$$

It seems it would be painful to find $x[n]$ via the inverse DTFT integral!

Simpler (approximate!) approach: sample $\mathcal{X}(\omega)$, then compute inverse DFT (using FFT).





In fact in this case there is an analytical solution: $x[n] = \frac{1}{4} \operatorname{sinc}^3\left(\frac{1}{4} n\right)$, so we can compare $x[n]$ and $x_{\text{ps}}[n]$.

The modulo function

If $m = m_0 + lN$ with $m_0 \in \{0, 1, \ldots, N-1\}$ and $l \in \mathbb{Z}$ then $m \bmod N = m_0$.
You can also think of $m \bmod N$ as the **remainder** when dividing m by N.

<u>Example</u>. $1 \bmod 4 = 1$; $7 \bmod 4 = 3$; $-1 \bmod 4 = 3$; $-8 \bmod 4 = 0$.

For *any* signal $x[n]$, be it time-limited or not, we define the **N-point periodic superposition** of $x[n]$ as follows:

$$x_{\text{ps}}[n] \triangleq \sum_{l=-\infty}^{\infty} x[n - lN] \, .$$

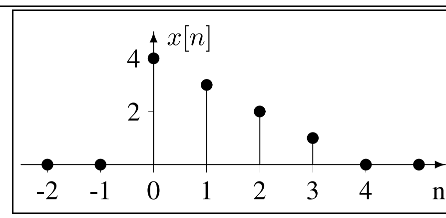
Note that *all* the values of $x[n]$ affect $x_{\text{ps}}[n]$.
The book uses the terms "**periodic extension**" (p396) and "**periodic repetition**" (p395) to describe $x_{\text{ps}}[n]$, but these terms are less descriptive. The term *superposition* always implies a summation in signal processing.

For *any* signal $x[n]$, be it time-limited or not, we define the **N-point circular extension** of a signal $x[n]$ as follows:

$$x((n))_N \triangleq x[n \bmod N] \, .$$

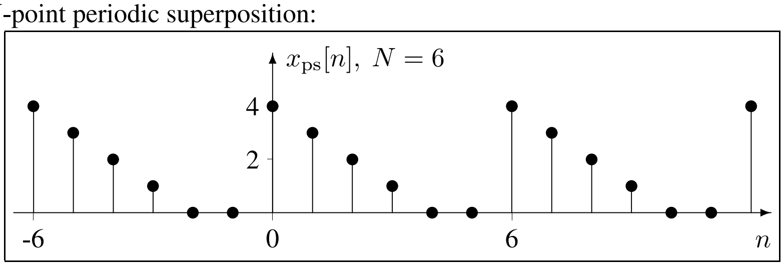
Note that *only* the values of $x[n]$ for $n \in \{0, \ldots, N-1\}$ affect $x[n \bmod N]$. So this is truly an "extension."

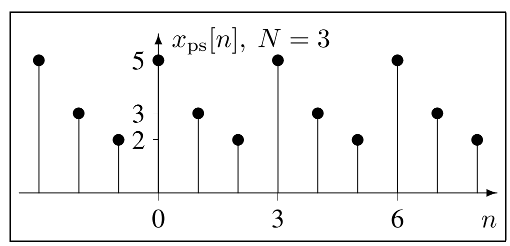
- Both $x_{\text{ps}}[n]$ and $x[n \bmod N]$ are N-periodic signals. They are both defined for all values of $n \in \mathbb{Z}$.
- In general, $x_{\text{ps}}[n]$ and $x[n \bmod N]$ are different signals.
- If $x[n]$ is a **time-limited signal** over $0, \ldots, L-1$, also called a **finite-length sequence**, with $L \leq N$, then $x_{\text{ps}}[n] = x[n \bmod N]$, and they consist of of shifted replicates of $x[n]$. Otherwise $x_{\text{ps}}[n]$ and $x[n \bmod N]$ differ!



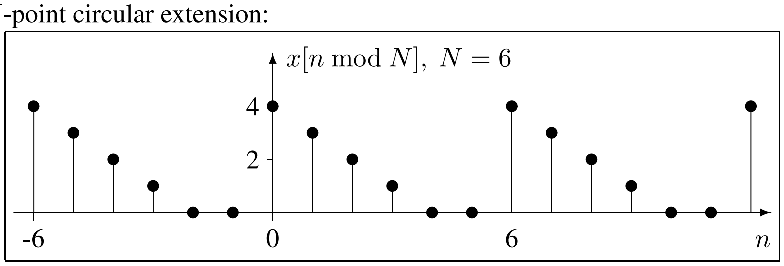
<u>Example</u>. $x[n] = \{\underline{4}, 3, 2, 1\} = \begin{cases} 4 - n, & 0 \leq n \leq 3 \\ 0, & \text{otherwise,} \end{cases}$ so $L = 4$.

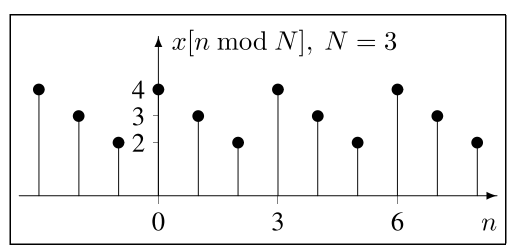
N-point periodic superposition:





N-point circular extension:





Why is this important? Because the following two processes are equivalent only if $L \leq N$.

$$x[n] \to \boxed{\text{DTFT}} \to \mathcal{X}(\omega) \to \boxed{\text{sample}} \to \left\{ \mathcal{X}\left(\tfrac{2\pi}{N} k\right) \right\}_{k=0}^{N-1} \to \boxed{N\text{-point inv DFT}} \to x_{\text{ps}}[n]$$

$$x[n] \to \boxed{N\text{-point DFT}} \to \boxed{N\text{-point inv DFT}} \to x[n] \text{ or } x[n \bmod N]$$

5.1

Frequency domain sampling: Properties and applications

Recall that the DTFT $\mathcal{X}(\omega)$ is periodic, and that *this periodicity arose because of the time-domain sampling*.

5.1.1 **Frequency-domain sampling and reconstruction of DT signals**

Now suppose we **sample the DTFT** at the N particular frequencies given above (think: to store digitally):

$$\mathcal{X}\left(\frac{2\pi}{N}k\right) = \mathcal{X}(\omega)\Big|_{\omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1.$$

Picture of $[0, 2\pi)$ ***with sample locations and values, and around unit circle too.***

Questions arise:

- I. How is $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$ related to DTFS (both involve unit circle) and to DFT $\{X[k]\}_{k=0}^{N-1}$.
- II. When (if ever) can we recover $x[n]$ from $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$ or from DFT $\{X[k]\}_{k=0}^{N-1}$?
  Probably not always since loss of information; after all inverse DTFT requires an integral over ω.
- III. How can we recover $x[n]$ from $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$ or from DFT $\{X[k]\}_{k=0}^{N-1}$?

Before doing math, think about duality. Earlier we considered sampling a CT signal, which led to replication in frequency domain, and recovery possible if signal bandlimited (finite support in replicated domain). Now we are sampling in the frequency domain.

We explore this by manipulating and re-interpreting the formula for $\mathcal{X}\left(\frac{2\pi}{N}k\right)$.

First break up the infinite summation into the intervals $\ldots(-2N..-N-1)\ (-N..-1)\ (0..N-1)\ (N..2N-1)\ (2N..3N-1)\ldots$

$$\begin{aligned}
\mathcal{X}\left(\frac{2\pi}{N}k\right) &= \mathcal{X}(\omega)\Big|_{\omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} \\
&= \sum_{l'=-\infty}^{\infty} \sum_{n'=l'N}^{l'N+N-1} x[n']\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn'} \quad \text{(split up sum)} \\
&= \sum_{l'=-\infty}^{\infty} \sum_{n=0}^{N-1} x[n+l'N]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}k(n+l'N)} \quad (n = n' - l'N') \\
&= \sum_{l'=-\infty}^{\infty} \sum_{n=0}^{N-1} x[n+l'N]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} \quad \text{(simplify exponent, let } l = -l'\text{)} \\
&= \sum_{n=0}^{N-1} \left(\sum_{l=-\infty}^{\infty} x[n-lN]\right) \mathrm{e}^{-\jmath\frac{2\pi}{N}kn} \quad \text{(exchange sums)}
\end{aligned}$$

where $n = n' - l'N$ and $l = -l'$. Thus

$$\boxed{\mathcal{X}\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1} x_{\mathrm{ps}}[n]\,\mathrm{e}^{-\jmath\frac{2\pi}{N}kn} \quad \text{where} \quad x_{\mathrm{ps}}[n] \triangleq \sum_{l=-\infty}^{\infty} x[n-lN]\,.}$$

The above relationship holds for *any* aperiodic DT signal whose DTFT exists. It is true $\forall N \in \mathbb{N}$ and $\forall k \in \mathbb{Z}$.
I call $x_{\mathrm{ps}}[n]$ the N-point periodic superposition of $x[n]$.

The signal $x_{\mathrm{ps}}[n]$ is N-periodic, *i.e.*, clearly $x_{\mathrm{ps}}[n+N] = x_{\mathrm{ps}}[n]$.

In summary: $x_{\mathrm{ps}}[n] \overset{\mathrm{DFT}}{\underset{N}{\longleftrightarrow}} X_{ps}[k] = \mathcal{X}\left(\frac{2\pi}{N}k\right)$

What tool do we use to represent periodic discrete-time signals?

Since $x_{\mathrm{ps}}[n]$ is N-periodic, we know from Ch. 4 that we can expand it using a discrete-time Fourier series (DTFS):

$$x_{\mathrm{ps}}[n] = \sum_{k=0}^{N-1} c_k \, \mathrm{e}^{\jmath \frac{2\pi}{N} kn} \quad \text{where} \quad c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_{\mathrm{ps}}[n] \, \mathrm{e}^{-\jmath \frac{2\pi}{N} kn} .$$

Result I. Comparing the DTFS coefficients with the DTFT samples $\mathcal{X}\left(\frac{2\pi}{N} k\right)$, we see that

$$\boxed{c_k = \frac{1}{N} \mathcal{X}\left(\frac{2\pi}{N} k\right).}$$

Thus, we can recover the periodic signal $x_{\mathrm{ps}}[n]$ from the DTFT samples $\left\{\mathcal{X}\left(\frac{2\pi}{N} k\right)\right\}_{k=0}^{N-1}$ via the DTFS **synthesis** equation:

$$x_{\mathrm{ps}}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{X}\left(\frac{2\pi}{N} k\right) \mathrm{e}^{\jmath \frac{2\pi}{N} kn} .$$

However, recovery of $x_{\mathrm{ps}}[n]$ from $\left\{\mathcal{X}\left(\frac{2\pi}{N} k\right)\right\}_{k=0}^{N-1}$ does not alone ensure that we can recover the *original* signal $x[n]$ from the DTFT samples $\left\{\mathcal{X}\left(\frac{2\pi}{N} k\right)\right\}_{k=0}^{N-1}$.

To address this, we must further study how $x_{\mathrm{ps}}[n]$ relates to $x[n]$.

Recall from Ch. 4 that time-domain sampling yields a DT frequency spectrum that is a sum of shifted replicates of the CT frequency spectrum of $x_{\mathrm{a}}(t)$.

Similarly, here we are considering frequency-domain sampling, and the result is that the periodic signal $x_{\mathrm{ps}}[n]$ consists of a **sum of shifted replicates** of the discrete-time signal $x[n]$, as described by the relationship $x_{\mathrm{ps}}[n] = \sum_{l=-\infty}^{\infty} x[n - lN]$.

Example. Suppose $x[n] = a^n\, u[n]$ where $|a| < 1$, for which $\mathcal{X}(\omega) = \dfrac{1}{1 - a\,\mathrm{e}^{-\jmath\omega}}$. This is *not* a time-limited signal.

Now suppose we take N samples of this DTFT at $\mathcal{X}\left(\frac{2\pi}{N}k\right) = \mathcal{X}(\omega)\Big|_{\omega=\frac{2\pi}{N}k}$ (equally spaced around the unit circle).

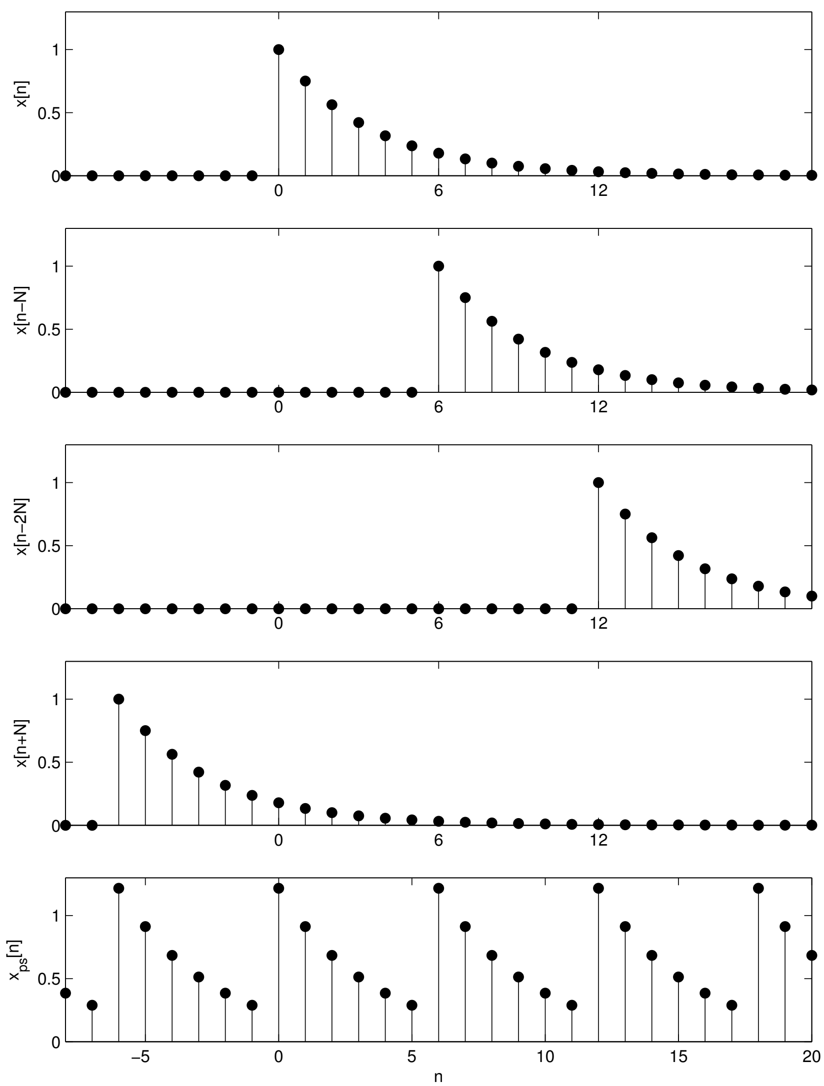
Then we synthesize $x_{\mathrm{ps}}[n]$ from those samples via inverse DFT. What will $x_{\mathrm{ps}}[n]$ look like?

From the above equation, for $n = 0, \ldots, N-1$:

$$x_{\mathrm{ps}}[n] = \sum_{l=-\infty}^{\infty} x[n - lN] = \sum_{l=-\infty}^{\infty} a^{n-lN}\, u[n - lN] = a^n \sum_{l=-\infty}^{\infty} a^{-lN}\, u[n - lN] = a^n \sum_{l=-\infty}^{\lfloor n/N \rfloor} a^{-lN} = a^n \sum_{l=-\infty}^{0} a^{-lN} = \frac{a^n}{1 - a^N}.$$

More generally, for any $n \in \mathbb{Z}$: $x_{\mathrm{ps}}[n] = \frac{a^{n \bmod N}}{1-a^N}$. Does $x_{\mathrm{ps}}[n] = x[n]$? No, but as $N \to \infty$, $x_{\mathrm{ps}}[n] \to x[n]$ for any fixed $n \geq 0$.

The problem is **time-domain aliasing**; the time-domain replicates **overlap** before summing.



This time aliasing arises from our attempt to form a signal from a finite number of values of ω.

For this aperiodic non time-limited signal, no finite set of ω_k's will allow perfect reconstruction of $x[n]$.

Time-limited signals

There is an important "special case" where the time-domain replicates do not overlap. That case is **time-limited** signals.

By convention, we say $x[n]$ is **time-limited** with duration L if $x[n]$ is nonzero only in the interval $0, 1, \ldots, L-1$.

Result II.

If $x[n]$ is time-limited to duration L, then if $N \geq L$, then there is no overlap in the replicates when forming $x_{\mathrm{ps}}[n]$ from $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$.

Result III. Thus, if $x[n]$ is time-limited with duration $L \leq N$, (called **adequate frequency sampling**) then we can recover $x[n]$ from $x_{\mathrm{ps}}[n]$ by simply extracting the appropriate values:

$$x[n] = \begin{cases} x_{\mathrm{ps}}[n], & 0 \leq n \leq L-1 \\ 0, & \text{otherwise.} \end{cases}$$

From the recovered $x[n]$ we could use the DTFT formula to find $\mathcal{X}(\omega)$ for any ω, meaning we have gone from the samples $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$ back to the continuous $\mathcal{X}(\omega)$ (in the time-limited case). It is also possible to recover $\mathcal{X}(\omega)$ directly from the samples using the **Dirichlet interpolation** formula - see text.

Review FT family tree.

5.1.2

The discrete Fourier transform (DFT)

For the rest of this chapter, our primary focus will be **time-limited** signals $x[n]$, which are nonzero only for $n = 0, \ldots, L-1$.

For such signals, the DTFT "simplifies" to

$$\mathcal{X}(\omega) = \sum_{n=0}^{L-1} x[n]\, \mathrm{e}^{-\jmath\omega n}.$$

We pick any $N \geq L$ and sample the DTFT uniformly over $[0, 2\pi)$

$$X[k] = \mathcal{X}(\omega)\Big|_{\omega=\frac{2\pi}{N}k} = \sum_{n=0}^{L-1} x[n]\, \mathrm{e}^{-\jmath\frac{2\pi}{N}kn}$$

or for convenience

$$X[k] = \sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath\frac{2\pi}{N}kn}, \quad k = 0, \ldots, N-1,$$

where the upper limit is changed to N with no effect since $x[n] = 0$ for $n = L, \ldots, N-1$.
The above expression is called the **discrete Fourier transform** or (**DFT**).

The DFT is always defined, since it is a finite sum!

Since $x[n]$ is time-limited, we can recover it from $x_{\mathrm{ps}}[n]$, which in turn can be recovered from the DTFS synthesis equation.

Thus we have the **inverse DFT** or (**IDFT**):

$$x[n] = \begin{cases} \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, \mathrm{e}^{\jmath\frac{2\pi}{N}kn}, & n = 0, \ldots, N-1 \\ 0, & \text{otherwise.} \end{cases}$$

The above pair of equations can be implemented easily on a computer since there is just a finite sum. Fortunately, we can implement the sums cleverly using the **fast-Fourier transform** (**FFT**), as discussed in Ch. 6.

Although the ranges of the indices only run from 0 to $N-1$, the expressions are defined for any integer k and n.

5.1.3 The DFT as a linear transformation ***skim***

Define

$$\boxed{W_N \triangleq e^{-\jmath \frac{2\pi}{N}}}$$

then we can rewrite the DFT and IDFT formulas as:

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn}, \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, W_N^{-kn},}$$

which we can also write in matrix-vector form $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{x}$ and $\boldsymbol{x} = \frac{1}{N}\boldsymbol{W}'\boldsymbol{X}$ (see text).

Graduate students should study the matrix-vector form, since it is very useful for theoretical analysis of many SP methods.

5.1.4 Relationship of the DFT to other transforms

- DTFS Suppose we take *any* signal $x[n]$ and form its N-point circular extension, *i.e.*, $x[n \bmod N]$. Since $x[n \bmod N]$ is N-periodic, it has a DTFS representation with some coefficients c_k. Those coefficients are related to the N-point DFT of $x[n]$ as follows:
$$\boxed{c_k = \frac{1}{N} X[k]\,.}$$
- DTFS
Suppose we take any signal $x[n]$ and form its N-point periodic superposition, *i.e.*, $x_{\mathrm{ps}}[n] = \sum_{l=-\infty}^{\infty} x[n - lN]$. Since $x_{\mathrm{ps}}[n]$ is N-periodic, it has a DTFS representation with some coefficients c_k. Those coefficients are related to the DTFT of $x[n]$ as follows:
$$\boxed{x[k] = \frac{1}{N} \mathcal{X}\left(\frac{2\pi}{N} k\right).}$$
If in addition $x[n]$ has length $L \leq N$, then the DTFS coefficients are related to the N-point DFT of $x[n]$ as follows: $c_k = \frac{1}{N} X[k]$.
- DTFT
If $x[n]$ is an L-point signal with $L \leq N$ then $\boxed{X[k] = \mathcal{X}(\omega)\Big|_{\omega = \frac{2\pi}{N} k}.}$
- z-transform
If $x[n]$ is an L-point signal with $L \leq N$ then $\boxed{X[k] = X(z)\Big|_{z = e^{\jmath \frac{2\pi}{N} k}}.}$
- If $x[n]$ is an L-point signal with $L \leq N$, then we can express $X(z)$ in terms of $X[k]$ (see text):
$$X(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X[k]}{1 - e^{\jmath \frac{2\pi}{N} k} z^{-1}}.$$
- If $x[n]$ is an L-point signal with $L \leq N$, then we can express $\mathcal{X}(\omega)$ in terms of $X[k]$. (Left to reader.)
- Fourier series of continuous time periodic signal
see text. DFT values $X[k]$ related to aliased version of the CT Fourier coefficients.

The DFT, IDFT - computational perspective

In the above description we have introduced the DFT from the point of view of sampling the DTFT. In this section we take a more computational perspective, which may be simpler.

Given a DT signal $x[n]$, we define the N-point DFT of $x[n]$ to be

$$\boxed{X[k] \triangleq \sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N} kn}, \quad k = 0, \ldots, N-1.}$$

Since this is a finite sum, it is well defined for *any* signal $x[n]$, be it time-limited, periodic, or aperiodic.
The expression for $X[k]$ is certainly mathematically well defined for *any* integer k, and on paper it is sometimes useful to think about all integers k. However, the DFT is periodic with period N:

$$X[k+N] = X[k], \quad \forall k.$$

Thus, in a computer we will only bother to compute $X[k]$ for $k = 0, 1, \ldots, N-1$.
Input: N time values $\{x[n]\}_{n=0}^{N-1}$, output: N DFT values $\{X[k]\}_{k=0}^{N-1}$.

Given $\{X[k]\}_{k=0}^{N-1}$, the N-point inverse DFT is given by

$$\boxed{x_i[n] \triangleq \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, \mathrm{e}^{\jmath \frac{2\pi}{N} kn}, \quad n = 0, \ldots, N-1.}$$

This expression is also mathematically well defined for *any* integer n, but on paper it is also sometimes useful to think about all integers n. However, the IDFT is also periodic with period N:

$$x_i[n+N] = x_i[n], \quad \forall n.$$

Thus, again, in a computer we would only bother to compute this sum for $n = 0, \ldots, N-1$. Again an N-point input and N-point output.

Now we discuss the properties of the DFT and the IDFT for various types of DT signals $x[n]$.

- 1. For *any* DT signal $x[n]$, we have
$$x[n] = x_i[n], \quad \text{for } n = 0, \ldots, N-1,$$
so the inverse DFT is aptly named. If you take the DFT of *any* DT signal, be it time-limited, not time-limited, periodic, or aperiodic, and then compute the IDFT, the IDFT $x_i[n]$ will agree with the original signal $x[n]$ over the interval $n = 0, \ldots, N-1$.
- 2. If $x[n]$ is a **periodic** signal with period N, then it has a DTFS representation with some coefficients $\{c_k\}_{k=0}^{N-1}$ where
$$X[k] = N c_k.$$
In this case $x[n] = x_i[n]$ for *all* n.
- 3. If $x[n]$ is **time-limited** to $0, \ldots, L-1$ where $L \leq N$, then in addition to Property 1, we have
$$X[k] = \mathcal{X}(\omega)\Big|_{\omega = \frac{2\pi}{N} k}.$$
If $x[n]$ is not time-limited, then there is still a relationship between $X[k]$ and $\mathcal{X}(\omega)$, but it is fairly complicated and not easily interpreted. Most of the time we do want to be able to directly interpret the DFT values $X[k]$'s as frequency components, so we focus on time-limited signals.
In the time-limited case, we can "recover" the original signal from $x_i[n]$ by simply taking
$$x[n] = \begin{cases} x_i[n], & n = 0, \ldots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

Since we will focus mainly on time-limited signals in the context of the DFT, for simplicity we will replace the $x_i[n]$ in the IDFT with just $x[n]$. But one should keep in mind that the IDFT formula really gives a periodic extension of $\{x[n]\}_{n=0}^{N-1}$, and to "truly recover" $x[n]$ using the IDFT you would set to zero all the values other than $n = 0, 1, \ldots, N-1$.

5.2 Properties of the DFT, IDFT

$$X[k] = \sum_{n=0}^{N-1} x[n] \, e^{-\jmath \frac{2\pi}{N} kn}, \quad k = 0, \ldots, N-1. \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \, e^{\jmath \frac{2\pi}{N} kn}, \quad n = 0, \ldots, N-1.$$

5.2.1 Periodicity, linearity, and symmetry properties

Periodicity

$$X[k+N] = X[k] \ \forall k$$

The above formula is easy to see by plugging in to the DFT definition and simplifying the complex exponential.

$$x_i[n+N] = x_i[n] \ \forall n$$

This formula may seem surprising, since supposedly we are dealing with time-limited signals! Frequency-domain sampling leads to periodic replication in the time domain. To "recover" the original time-limited signal from the above formulas, we would just use $n = 0, \ldots, N-1$ and zero the rest. In a computer we essentially only care about 0 to $N-1$ anyway, so the signal periodicity is mainly a theoretical property, except as it impacts convolution below.

So mathematically we are dealing with N-periodic sequences, even though practically we are thinking of it as finite-length.

Another way of thinking about it: suppose you give me two finite-length sequences and ask me to convolve them. All you really care is that I give you back the correct answer (and quickly); it does not matter to you as the "client" that the convolution algorithm that I use may be based on ideas involving periodic signals, as long I as return to you the correct answer.

Remember, one of our goals here is to develop a (much) faster convolution method than the brute-force `conv` approach. It so happens that periodic signals play an important role in the derivation and analysis, but by using the final results appropriately, we will also end up with a very practical method that works even for non-periodic time-limited signals.

Linearity

$$a\,x[n] + b\,y[n] \ \underset{N}{\overset{\text{DFT}}{\longleftrightarrow}} \ a\,X[k] + b\,Y[k]$$

Post for properties

Most of the properties of the DTFT have analogous relationships for the DFT. However, our previous definitions of signal properties and operations like symmetries, time-reversal, time-shifts, etc. do not quite work directly for time-limited signals over $0, \ldots, N-1$. Time-limited signals are never symmetric in the sense used previously. And ordinary time-reversal or time-shifting would move some or many of the sample values outside of the interval $0, \ldots, N-1$. What we need is a new concept of termsymmetry, **time-reversal**, and **time-shift** in the context of time-limited signals. The "appropriate" definitions (meaning the ones that work with the DFT) are defined in terms of the periodic extension of a signal.

(Do convolution last, after familiar with circular properties.)

Symmetries

The next set of properties of the DFT describes what happens when the signal $x[n]$ has certain symmetries. Hence we first need to appropriately define "symmetries" in the context of periodic sequences.

A signal $x[n]$ is called N**-point circularly even** iff its N-point circular extension, $x[n \bmod N]$, is even.

Equivalently, a signal $x[n]$ is N-point circularly even iff $x[n \bmod N] = x[-n \bmod N]$.

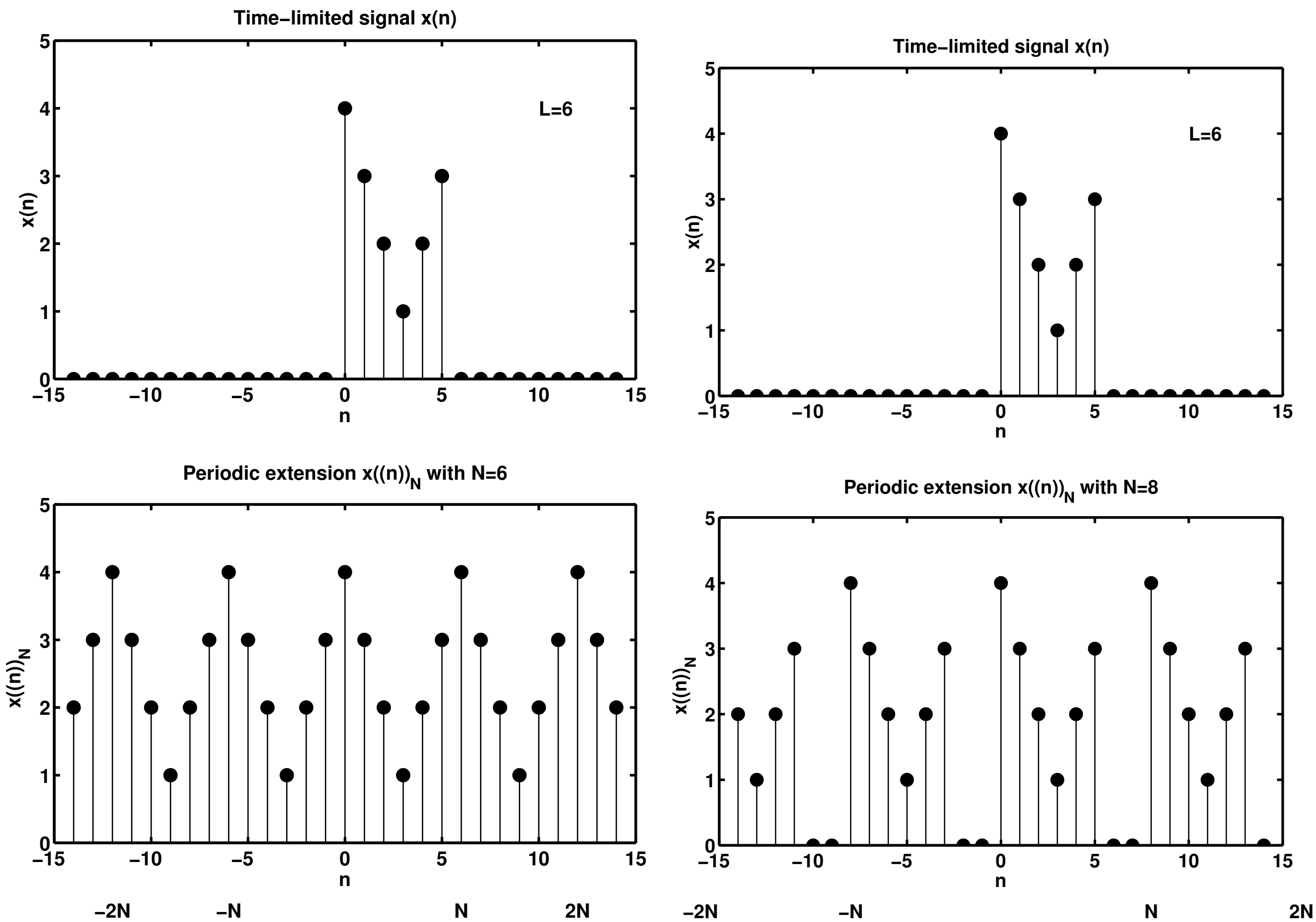
Example. $x[n] = \{\underline{4}, 3, 2, 1\}$.

Is this signal 4-point circularly even? No, since $x[n \bmod N]$ is not even.

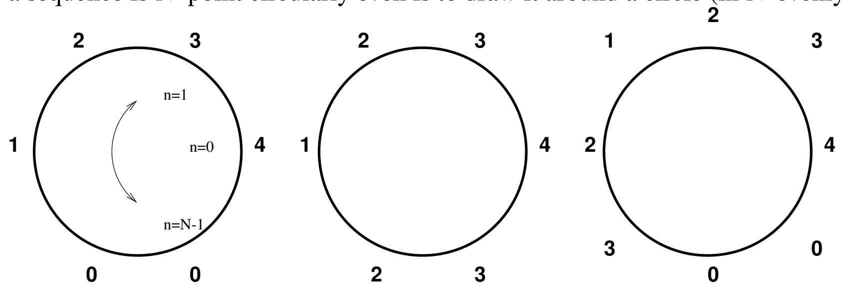
Example. $x[n] = \{\underline{4}, 3, 2, 1, 2, 3\}$

Is this signal 6-point circularly even? Yes, as the following figure illustrates.

Is it 8-point circularly even? No, for $N = 8$ it is not circularly even, since $x[n \bmod N]$ is not even for $N = 8$.



A simple test for whether a sequence is N-point circularly even is to draw it around a circle (in N evenly spaced points).



If the sequence is the same whether you read it out CW or CCW, then it is circularly even.

By considering the points around the circle, we conclude the following.

An signal is N**-point circularly even** iff $x[N-n] = x[n],\ n = 1, \ldots, N-1$, where $x[0]$ is arbitrary.
An signal is N**-point circularly odd** iff $x[N-n] = -x[n],\ n = 1, \ldots, N-1$, with $x[0] = 0$.

Example. A 6-point circularly odd sequence: $x[n] = \{\underline{0}, -3, 2, 0, -2, 3\}$.

Decomposing an N-point sequence into circularly even and circularly odd components:

$$x[n] = x_{\mathrm{ce}}[n] + x_{\mathrm{co}}[n]$$

$$x_{\mathrm{ce}}[n] = \begin{cases} \frac{1}{2}\left(x[n] + x[N-n]\right), & n = 1, \ldots, N-1 \\ x[0], & n = 0 \end{cases} \qquad x_{\mathrm{co}}[n] = \begin{cases} \frac{1}{2}\left(x[n] - x[N-n]\right), & n = 1, \ldots, N-1 \\ 0, & n = 0 \end{cases}$$

Example. $x[n] = \{\underline{4}, 3, 2, 1\}$ with $N = 4 \Longrightarrow x_{\mathrm{ce}}[n] = \{\underline{4}, 2, 2, 2\}$, $x_{\mathrm{co}}[n] = \{\underline{0}, 1, 0, -1\}$ ***circle pictures***

Now we are finally ready to return to the properties of the DFT.

Symmetry properties

- If $x[n]$ is real, then (*cf.* **Hermitian symmetry** of DTFT) its DFT has **circular Hermitian symmetry**:

$$X[k] = X^*[-k \bmod N].$$

Proof. It follows by the following periodicity argument, since $-k \bmod N = -k + lN$ for some $l \in \mathbb{Z}$:

$$X^*[-k \bmod N] = \left[\sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N}(-k \bmod N)n}\right]^* = \left[\sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N}(-k)n}\right]^* = \sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N}kn} = X[k].$$

- If $x[n]$ is circularly even, then $X[k]$ is circularly even.
Proof (using $n' = N - n$)

$$X[k] = \sum_{n=0}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N}kn} = x[0] + \sum_{n=1}^{N-1} x[n]\, \mathrm{e}^{-\jmath \frac{2\pi}{N}kn} = x[0] + \sum_{n'=1}^{N-1} x[N-n']\, \mathrm{e}^{-\jmath \frac{2\pi}{N}k(N-n')}$$

$$= \sum_{n'=0}^{N-1} x[n']\, \mathrm{e}^{-\jmath \frac{2\pi}{N}(-k)n'} = X[-k \bmod N].$$

- Combining the above:

If $x[n]$ is real and circularly even, then $X[k]$ is also real and circularly even.

- There are many such relationships, as summarized in the following diagram.

$$\begin{array}{ccccccccc} x[n] & = & x_R^{\mathrm{ce}}[n] & + & \jmath x_I^{\mathrm{ce}}[n] & + & x_R^{\mathrm{co}}[n] & + & \jmath x_I^{\mathrm{co}}[n] \\ & & \updownarrow & & \updownarrow & & & \times & \\ X[k] & = & X_R^{\mathrm{ce}}[k] & + & \jmath X_I^{\mathrm{ce}}[k] & + & X_R^{\mathrm{co}}[k] & + & \jmath X_I^{\mathrm{co}}[k] \end{array}$$

Note: *(5.2.31)* looks different than *(4.3.37)* only because the order of terms differs.

5.2.3 Additional DFT properties

Circular time-reversal

Again we must be careful, since ordinary time-reversal a time-limited sequence would yield a sequence that is not limited to 0 to $N-1$. Instead, we first take the N-point circular extension of the signal, time-reverse that, and then pick out the values from 0 to $N-1$. This is called **circular time-reversal**. It is equivalent to writing the sequence CCW around a circle, and then reading the values CW.

Example. If $x[n] = \{\underline{10}, 11, 12, 13, 14\}$ and $N = 6$, then $x[-n \bmod N] = \{\underline{10}, 0, 14, 13, 12, 11\}$. ***Picture of circle***

We will use $x[-n \bmod N]$ to denote circular time-reversal. Like the textbook, we may sometimes use the convenient shorthand "$x[-n \bmod N] = x[N-n]$." But note that this shorthand is imprecise since $x[-n \bmod N]\Big|_{n=0} = x[0]$ rather than $x[N]$. Specifically:

$$x[-n \bmod N] = \begin{cases} x[0], & n = 0 \\ x[N-n], & n = 1, \ldots, N-1 \\ \text{(periodic)}, & \text{otherwise.} \end{cases}$$

The DFT property is (*cf.* DTFT property: $x[-n] \overset{\text{DTFT}}{\leftrightarrow} \mathcal{X}(-\omega)$):

$$\boxed{x[-n \bmod N] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X[-k \bmod N]\,.}$$

Circular time-shift

Again we must be careful, since time-shifting a time-limited sequence would yield a sequence that is not limited to 0 to $N-1$. Instead, we first take the periodic extension of the signal, time-shift that, and then pick out the values from 0 to $N-1$. This is called a N**-point circular time-shift**. It is equivalent to writing the sequence CCW around a circle, and then reading the values CCW starting from point $-l$.

Example. If $x[n] = \{\underline{10}, 11, 12, 13, 14\}$ and $N = 6$ and $l = 2$, then $x[n - l \bmod N] = \{\underline{14}, 0, 10, 11, 12, 13\}$. ***Picture of circle***

$$\boxed{x[n - n_0 \bmod N] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} e^{-\jmath \frac{2\pi}{N} k n_0} X[k]}$$

Circular frequency-shift / complex modulation

$$\boxed{x[n]\, e^{\jmath \frac{2\pi}{N} k_0 n} \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X[k - k_0 \bmod N]}$$

Complex conjugate

$$\boxed{x^*[n] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X^*[-k \bmod N]} \qquad \boxed{x^*[-n \bmod N] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X^*[k]}$$

Parseval's theorem

$$\boxed{\sum_{n=0}^{N-1} x[n]\, y^*[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, Y^*[k]} \qquad \boxed{\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2}$$

Expresses signal time-domain energy in terms of "average energy in frequency components."

Duality

Exercise.

Recall that one of the most important DTFT properties is that if $\mathcal{Y}(\omega) = \mathcal{H}(\omega)\,\mathcal{X}(\omega)$ then $y[n] = x[n] * h[n]$, so convolution in the time domain becomes simply multiplication in the transform domain. What about for the DFT?

5.2.2

Multiplication of two DFTs and circular convolution (This is one of the most important topics in this chapter.)

Suppose $h[n] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} H[k]$ and $x[n] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X[k]$, where $h[n]$ is time-limited with $M \leq N$ and $x[n]$ is time-limited with $L \leq N$. (The N's *must* be same!)

(For now think of $x[n]$ and $h[n]$ as two generic signals; they need not be an "input" and a "filter.")

Suppose we multiply the two sets of DFT coefficients:

$$S[k] \stackrel{\triangle}{=} H[k]\,X[k], \quad k = 0, \ldots, N-1,$$

and then take the N-point inverse DFT of $\{S[k]\}_{k=0}^{N-1}$ to get $\{s[n]\}_{k=0}^{N-1}$. How does $s[n]$ relate to $h[n]$ and $x[n]$?

The resulting $s[n]$ is *not* the ordinary convolution of $h[n]$ and $x[n]$, but rather the N**-point circular convolution** of $h[n]$ and $x[n]$, denoted $s[n] = h[n] \; Ⓝ \; x[n]$.

(The derivation is given on the next page.)

We can interpret the N-point circular convolution expression

$$s[n] = x[n] \; Ⓝ \; h[n] = \sum_{m=0}^{N-1} x[m]\, h[n - m \bmod N]$$

in words as follows.

- Take *one* of the two sequences, *e.g.*, $h[n]$, and form its N-point circular extension $h[n \bmod N]$.
- Perform ordinary convolution of that extended signal $h[n \bmod N]$ with the time-limited signal $x[n]$. We need only bother to compute the results for $n = 0, \ldots, N-1$.

Example. Circular convolution (using DFT): $x[n] = \{\underline{2}, 0, 3, -1\}$, $h[n] = \{\underline{10}, 20, 30, 40\}$. Find $s[n] = h[n] \; ④ \; x[n]$, *i.e.*, $N = 4$.

Circular convolution, the fast way (in frequency domain):

```
s = ifft(fft([2 0 3 -1]) .* fft([10 20 30 40]))
```

yields

$$s[n] = h[n] \; ④ \; x[n] = \{\underline{90}, 130, 50, 130\}.$$

Note that since $s[n]$ is the IDFT of $S[k] = H[k]\,X[k]$, it is periodic with period $N = 4$.

Recipe for finding $h[n] \; ④ \; x[n]$ manually in time domain:

- Draw one sequence $x[n]$ CCW around a circle.
- Fold (time reverse) the other sequence $h[n]$ by drawing it CW around a circle. See illustration on subsequent page.
- $s[0]$ is the sum of the element-by-element product of the two sequences around the circle. For the above example, $s[0] = x[0]\,h[0] + x[1]\,h[3] + x[2]\,h[2] + x[3]\,h[1] = 2 \cdot 10 + 0 \cdot 40 + 3 \cdot 30 + (-1) \cdot 20 = 90$.
- Shift the folded sequence by rotating CCW around the circle, then multiply and sum. For the above example: $s[1] = x[0]\,h[3] + x[1]\,h[2] + x[2]\,h[1] + x[3]\,h[0] = 2 \cdot 40 + 0 \cdot 30 + 3 \cdot 20 + (-1) \cdot 10 = 130$.
- Repeat N times. Thankfully we can do this with FFTs painlessly.

Illustrated on subsequent page.

If asked to do it manually, one can either use the above time domain recipe, or manually compute $X[k]$ and $H[k]$ and multiply, and then manually compute inverse DFT.

Circular Convolution Derivation

Input information:

$$H[k] = \sum_{n=0}^{N-1} h[n]\, e^{-\jmath \frac{2\pi}{N} kn} \qquad X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-\jmath \frac{2\pi}{N} kn}$$

$$S[k] = H[k]\, X[k],\ k = 0, \ldots, N-1 \qquad s[n] = \frac{1}{N} \sum_{k=0}^{N-1} S[k]\, e^{\jmath \frac{2\pi}{N} kn}.$$

Goal: relate $s[n]$ to $h[n]$ and $x[n]$.

Recall that we previously showed (4.2.2) the following useful equality:

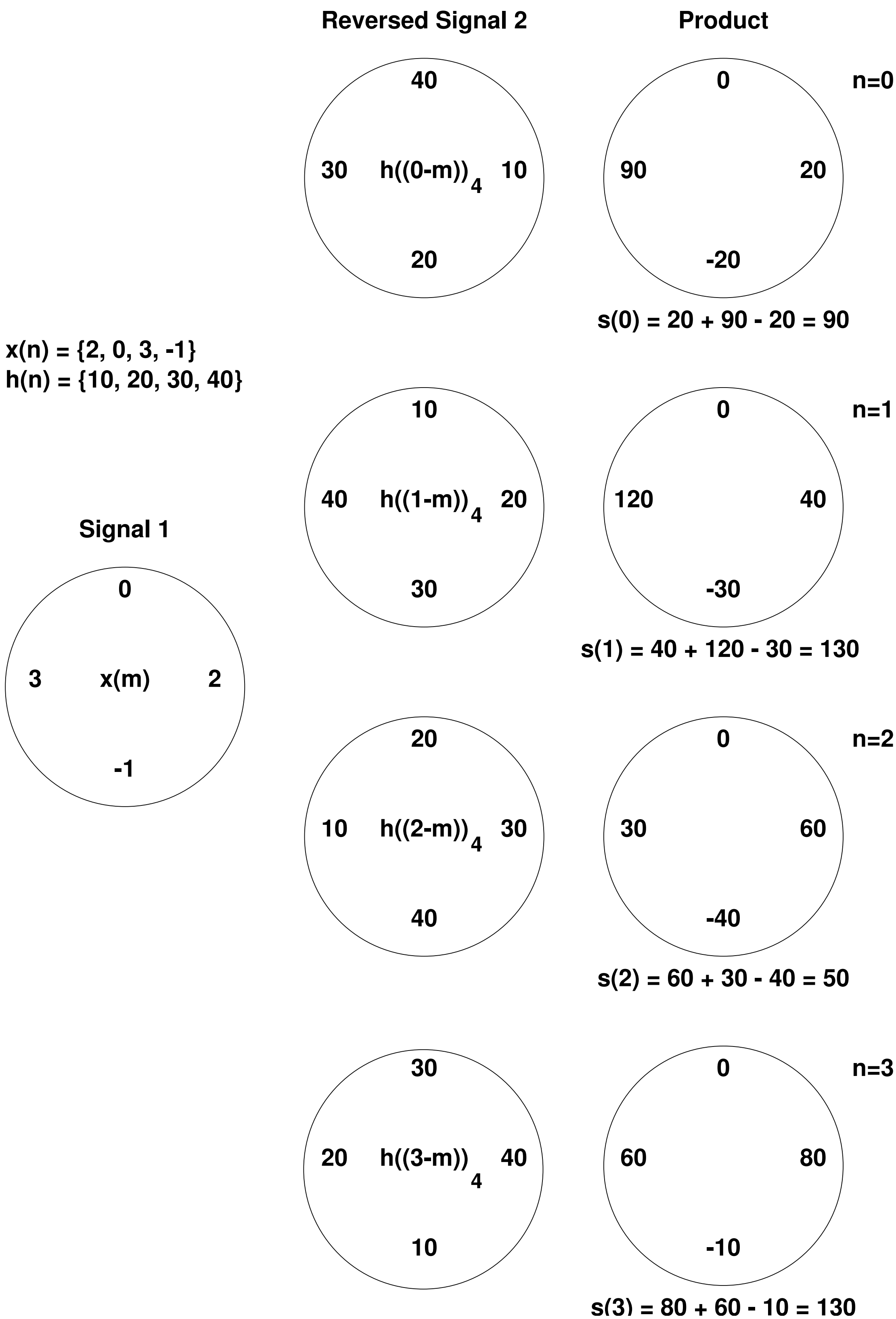
$$\frac{1}{N} \sum_{k=0}^{N-1} e^{\jmath \frac{2\pi}{N} km} = \begin{cases} 1, & m = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases} = \sum_{l=-\infty}^{\infty} \delta[m - lN] = \delta[n \bmod N].$$

Another useful fact: $h[n' \bmod N] = \sum_{m'=0}^{N-1} h[m'] \sum_{l=-\infty}^{\infty} \delta[n' - m' - lN]$.

$$\begin{aligned}
s[n] &= \frac{1}{N} \sum_{k=0}^{N-1} S[k]\, e^{\jmath \frac{2\pi}{N} kn} = \frac{1}{N} \sum_{k=0}^{N-1} H[k]\, X[k]\, e^{\jmath \frac{2\pi}{N} kn} \ \text{(IDFT and } S[k] \text{ def'n)} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{m'=0}^{N-1} h[m']\, e^{-\jmath \frac{2\pi}{N} km'} \right] \left[\sum_{m=0}^{N-1} x[m]\, e^{-\jmath \frac{2\pi}{N} km} \right] e^{\jmath \frac{2\pi}{N} kn} \ \text{(DFT def'n)} \\
&= \sum_{m=0}^{N-1} x[m] \sum_{m'=0}^{N-1} h[m'] \left[\frac{1}{N} \sum_{k=0}^{N-1} e^{\jmath \frac{2\pi}{N} k(n-m-m')} \right] \ \text{(exch. sums)} \\
&= \sum_{m=0}^{N-1} x[m] \sum_{m'=0}^{N-1} h[m'] \left[\sum_{l=-\infty}^{\infty} \delta[n - m - m' - lN] \right] \ \text{(exp. sum prop.)} \\
&= \sum_{m=0}^{N-1} x[m]\, h[n - m \bmod N] \ \text{(modulo fact)} \\
&\triangleq h[n] \ Ⓝ \ x[n] = x[n] \ Ⓝ \ h[n] \quad \text{def'n of } Ⓝ
\end{aligned}$$

In summary, if $s[n] = \text{IDFT}_N(\text{DFT}_N(x[n])\ .*\ \text{DFT}_N(h[n]))$, then

$$\boxed{s[n] = h[n] \ Ⓝ \ x[n] \ \underset{N}{\overset{\text{DFT}}{\longleftrightarrow}} \ S[k] = H[k]\, X[k].}$$


Reversed Signal 2
Product
40
30
h((0-m))4
10
20
0
90
20
-20
n=0
s(0) = 20 + 90 - 20 = 90
x(n) = {2, 0, 3, -1}
h(n) = {10, 20, 30, 40}
10
40
h((1-m))4
20
30
0
120
40
-30
n=1
s(1) = 40 + 120 - 30 = 130
Signal 1
0
3
x(m)
2
-1
20
10
h((2-m))4
30
40
0
30
60
-40
n=2
s(2) = 60 + 30 - 40 = 50
30
20
h((3-m))4
40
10
0
60
80
-10
n=3
s(3) = 80 + 60 - 10 = 130

Remember we set out to find a faster way to do convolution. Did it work? Ordinary "linear convolution" for the above signals can be computed using: `y = conv([2 0 3 -1], [10 20 30 40])` which yields

$$y[n] = h[n] * x[n] = \{\underline{20}, 40, 90, 130, 70, 90, -40\}\,.$$

The result has $L + M - 1 = 4 + 4 - 1 = 7$ nonzero values. All other values are zero.

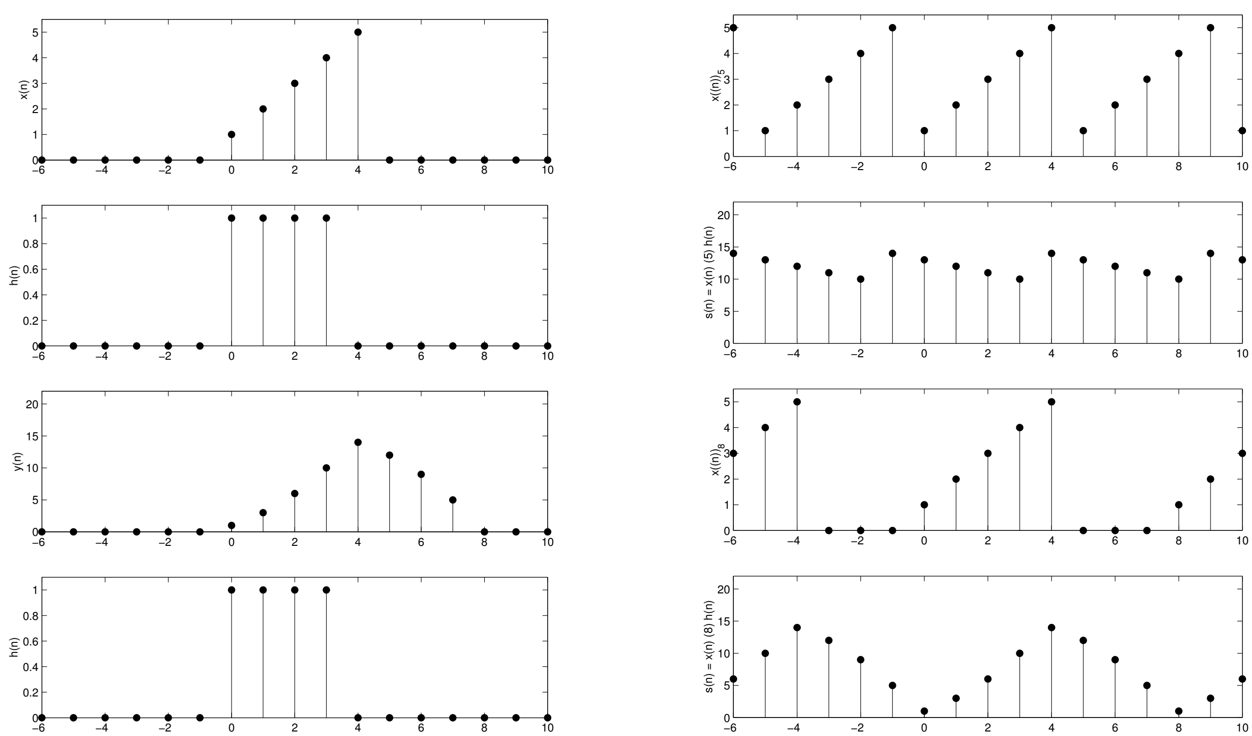
Note that $y[n] \neq s[n]$, *i.e.*, 4-point circular convolution of $x[n]$ and $h[n]$, which can be computed rapidly by FFT's, did not result in the linear convolution of $x[n]$ and $h[n]$. Fortunately this can be "fixed" using **zero-padding**.

How are $s[n]$ and $y[n]$ related? $s[n]$ is a time-domain aliased version of $y[n]$; a sum of 4-point shifted replicates of $y[n]$:

$$s[n] = \sum_{l=-\infty}^{\infty} y[n - lN] = \sum_{l=-\infty}^{\infty} (x * h)[n - lN].$$

If $N \geq L$ and $N \geq M$ (which is true in above example), then $x[n]$ and $h[n]$ are sampled adequately, so $H[k] = \mathcal{H}\left(\frac{2\pi}{N}k\right)$ and $X[k] = \mathcal{X}\left(\frac{2\pi}{N}k\right)$. So $S[k] = \mathcal{H}(\omega)\,\mathcal{X}(\omega)|_{\omega=\frac{2\pi}{N}k} = \mathcal{Y}(\omega)|_{\omega=\frac{2\pi}{N}k}$, and hence $s[n]$ will be the N-point periodic superposition of the inverse DTFT of $\mathcal{Y}(\omega) = \mathcal{H}(\omega)\,\mathcal{X}(\omega)$, *i.e.*, the N-point periodic superposition of $y[n] = h[n] * x[n]$.

```
-4              0              4              n
...
 20 40 90 130   70 90 -40                                  y[n+4]
                20 40  90 130  70 90 -40                   y[n]
                               20 40  90 130  70 90 -40    y[n-4]
                                              ...
          ... 90 130 50 130  ...                           s[n] (sum of above)
```



Exercise. Review the properties of linear convolution and generalize each property to the case of N-point circular convolution.

Example. $x[n] \; \textcircled{N} \; \delta[n - n_0] = \begin{cases} x[n - n_0 \bmod N], & n_0 \in \{0, \ldots, N-1\} \\ 0, & \text{otherwise.} \end{cases}$

Related DFT properties

If $s[n] = x[n] \circledN h[n]$ then

$$\sum_{n=0}^{N-1} s[n] = S[0] = H[0]\,X[0] = \left[\sum_{n=0}^{N-1} x[n]\right]\left[\sum_{n=0}^{N-1} h[n]\right],$$

which is a similar property to that for ordinary linear convolution, except here the sums are finite.

Circular correlation

$$\tilde{r}_{xy}(l) = x[l] \circledN y^*[-l \bmod N] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X[k]\,Y^*[k]$$

Circular autocorrelation

$$\tilde{r}_{xx}(l) = x[l] \circledN x^*[-l \bmod N] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} |X[k]|^2$$

Time-domain multiplication

$$x[n]\,y[n] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} \frac{1}{N} X[k] \circledN Y[k]$$

Properties of circular correlation ***Read***

- $\tilde{r}_{xy}(l) = x[n] \circledN y^*[-n \bmod N]\Big|_{n=l}$
- $\tilde{r}_{xy}(l) = \tilde{r}^*_{yx}(-l)$ (conjugate symmetry)

  Proof: $\tilde{r}_{yx}(l) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} Y[k]\,X^*[k] = (X[k]\,Y^*[k])^* \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} \tilde{r}^*_{xy}(-l)$ by complex conjugate property above.

  Thus $\tilde{r}_{yx}(l) = \tilde{r}^*_{xy}(-l)$.
- $\tilde{r}_{xy}(l)$ is periodic with period N
- $|\tilde{r}_{xy}(l)| \leq \sqrt{E_x E_y}$ where $E_x = \sum_{n=0}^{N-1} |x[n]|^2$ is the signal energy "in one period."

Proof for real signals:

$$\begin{aligned}
0 &\leq \sum_{n=0}^{N-1}\left[\frac{x[n]}{\sqrt{E_x}} \pm \frac{y[n-l \bmod N]}{\sqrt{E_y}}\right]^2 \\
&= \frac{1}{E_x}\sum_{n=0}^{N-1}[x[n]]^2 + \frac{1}{E_y}\sum_{n=0}^{N-1}[y[n-l \bmod N]]^2 \pm \frac{2}{\sqrt{E_x E_y}}\left[\sum_{n=0}^{N-1} x[n]\,y[n-l \bmod N]\right] \\
&= 2 \pm 2\frac{r_{xy}[l]}{\sqrt{E_x E_y}}.
\end{aligned}$$

Thus $-\sqrt{E_x E_y} \leq r_{xy}[l] \leq \sqrt{E_x E_y}$.

Summary

We have now covered all the most important transforms:
continuous time: Laplace, Fourier, Fourier Series,
discrete time: Z, DTFT, DTFS, DFT/FFT

The first six are for pencil and paper analysis/intuition/understanding. The DFT/FFT is for doing.

Example: time delay estimation, such as the radar example on HW.

If $y[n] = x[n-n_0] \approx x[n-n_0 \bmod N]$ then DFT gives $Y[k] = X[k]\,e^{-\jmath\frac{2\pi}{N}kn_0}$. Multiplying: $X[k]\,Y^*[k] = |X[k]|^2\,e^{\jmath\frac{2\pi}{N}kn_0}$, taking phase gives $\frac{2\pi}{N}kn_0$ which (after unwrapping) is a line in k, so find slope and $\hat{n}_0 = \text{slope}/\frac{2\pi}{N}$. Multiply by T_s to estimate delay (related to distance to target).

Note periodic ambiguity (tracking).

5.3 Linear filtering methods based on the DFT

5.3.1 Use of the DFT in linear filtering

DFT-based filtering *always* performs a circular convolution. This is a property of the math; usually it is not our desired operation. But by zero-padding adequately, circular convolution and linear convolution can yield the same result.

Goal: find $y[n]$ when:
$x[n] \to \boxed{h[n]} \to y[n] = h[n] * x[n]$
$x[n]$ time limited to $0, \ldots, L-1$
$h[n]$ time limited to $0, \ldots, M-1$ (FIR)

Brute force approach (about LM multiplies required): $y[n] = \sum_{k=0}^{L-1} x[k]\, h[n-k]$.
What range of n? $y[n]$ is time limited to $n = 0, \ldots, (L+M-1)-1$.

Since $y[n]$ is a $L+M-1$ point signal, we can reconstruct $y[n]$ from $N \geq L+M-1$ samples of its DTFT, *i.e.*,

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} Y[k]\, e^{\jmath \frac{2\pi}{N} kn}, \text{ where } Y[k] = \mathcal{Y}(\omega) \Big|_{\omega = \frac{2\pi}{N} k}.$$

But since $\mathcal{Y}(\omega) = \mathcal{H}(\omega)\, \mathcal{X}(\omega)$, then

$$Y[k] = \mathcal{Y}(\omega) \Big|_{\omega = \frac{2\pi}{N} k} = \mathcal{H}(\omega)\, \mathcal{X}(\omega) \Big|_{\omega = \frac{2\pi}{N} k}.$$

Does

$$X[k] = \mathcal{X}(\omega) \Big|_{\omega = \frac{2\pi}{N} k} \text{ and } H[k] = \mathcal{H}(\omega) \Big|_{\omega = \frac{2\pi}{N} k},$$

where $X[k]$ is the N-point DFT of $x[n]$? Yes, because $x[n]$ is time-limited to $L \leq N$, and $h[n]$ is time-limited to $M \leq N$. Thus

$$Y[k] = H[k]\, X[k],$$

and we have the following simple recipe for performing linear convolution using DFTs.

Convolution via DFT
- zero pad $x[n]$ to $N \geq L+M-1$ elements
- zero pad $h[n]$ to $N \geq L+M-1$ elements
- compute N-point DFT $X[k]$ of $x[n]$
- compute N-point DFT $H[k]$ of $h[n]$
- multiply: $Y[k] = X[k]\, H[k]$, $k = 0, \ldots, N-1$
- compute inverse DFT of $Y[k]$ to get $y[n]$

In MATLAB:

```
x = [2 0 3 -1]
h = [10 20 30 40]
X = fft(x,8) % tells to compute an 8-point DFT
H = fft(x,8)
Y = H .* X
y = ifft(y)
```

Brief way:
```
y = ifft(fft([2 0 3 -1], 8) .* fft([10 20 30 40], 8))
```
This returns 8 values. What is the last value? 0 because $L+M-1 = 7$

Key point: with zero-padding the general time-domain aliasing "phenomena" is still present, but the signal values that are time-aliased are all zero, so they cause no detrimental effect! We have thus "tricked" a circular convolution into performing something that gives the same answer as linear convolution.

5.3.2

Filtering of long data sequences

We have mentioned that signals stored in a computer must have finite length. But "finite" can still be very large.

Example: music CD: $L = 44.1\text{kHz} \cdot 70\,\text{min} \cdot 60 \frac{\text{sec}}{\text{min}} \cdot 2\text{channels} \approx 400 \cdot 10^6$ samples!

How can we digitally filter such signals if RAM smaller than signal data? Goal: linear convolution.

Options for filtering in MATLAB. Each has a role. Which option is best depends on L and on form of system description ($h[n]$ or $H(z)$ or $\mathcal{H}(\omega)$).

1. Direct convolution $y[n] = \sum_{k=0}^{n} x[n-k]\, h[k]$

```
y = conv(x,h)
```

- Given FIR $h[n]$ (length M)
- $x[n]$ can be arbitrarily large - read sequentially from storage.
- $\approx LM$ multiplies
- appropriate if $M \ll \log_2 L$

2. Recursion based on constant-coefficient linear difference equation

$$y[n] = \sum_{k=1}^{N} a_k\, y[n-k] + \sum_{k=0}^{M} b_k\, x[n-k],$$

```
y = filter(b,a,x)
```

- Given $H(z)$ rational (or pole-zero etc.)
- Usually N, M small relative to L
- $\approx L(N+M)$ multiplies

Fairly simple recursion to implement.

3. Zero-padded DFT/FFT from impulse response $y[n] = x[n] \circledN h[n]$ where $N \geq L + M - 1$.

```
y = ifft(fft(x,N) .* fft(h,N))
```

- Given FIR $h[n]$ length M
- $O(N \log_2 N)$ multiplies
- Appropriate when $M \approx L \approx N/2$, *i.e.*, signal and impulse response have similar lengths. (Often not the case, consider a CD with $400 \cdot 10^6$ samples; rarely would we need to use a filter so long. Even an IIR filter could be truncated to *far* fewer samples.

Caution: check imaginary part of `ifft` if both $x[n]$ and $h[n]$ are real!

4. Overlap-add method for very long sequences but short filters.
Recipe:
- FFT signal a block at a time
- multiply $X[k]$'s by $H[k]$'s with $h[n]$ zero padded if necessary
- stitch together results from each block.

see text for details.

```
y = fftfilt(h,x)
```

- Given $h[n]$ or $H[k]$
- $O(\frac{L}{N} N \log_2 N) = O(L \log_2 N)$ multiplies
- appropriate when $h[n]$ much shorter than $x[n]$

Is there an overlap-save method in MATLAB?

5. Zero-padded DFT/FFT from frequency response $\mathcal{H}(\omega)$, using the fact that $\mathcal{Y}(\omega) = \mathcal{H}(\omega)\,\mathcal{X}(\omega)$

```
Hv = ?
y = ifft(fft(x,N) .* Hv)
```

- Given $\mathcal{H}(\omega)$ but finding $h[n]$ is painful
- Exact only if $h[n]$ is FIR and $N \geq L + M - 1$, but for sufficiently large N, often a reasonable approximation. How large of N? Need $N \geq L + M_0 - 1$, where M_0 is approximate length of "significant nonzero" part of $h[n]$.
- $O(N \log_2 N)$ multiplies

What should `Hv` be? It should be $\mathcal{H}(\omega_k),\ k = 0, \ldots, N-1$, where $\omega_k = \frac{2\pi}{N}k$.

Example. Suppose we have a signal $x[n]$ and we wish to apply the following filter:

$$\mathcal{H}(\omega) = \begin{cases} e^{-\jmath\omega 16}, & |\omega| \leq \pi/4 \\ 0, & \pi/4 < |\omega| \leq \pi. \end{cases}$$

On paper "no problem:" take DTFT of $x[n]$, multiply by $\mathcal{H}(\omega)$, take inverse DTFT, gives exact $y[n]$ if no math errors.

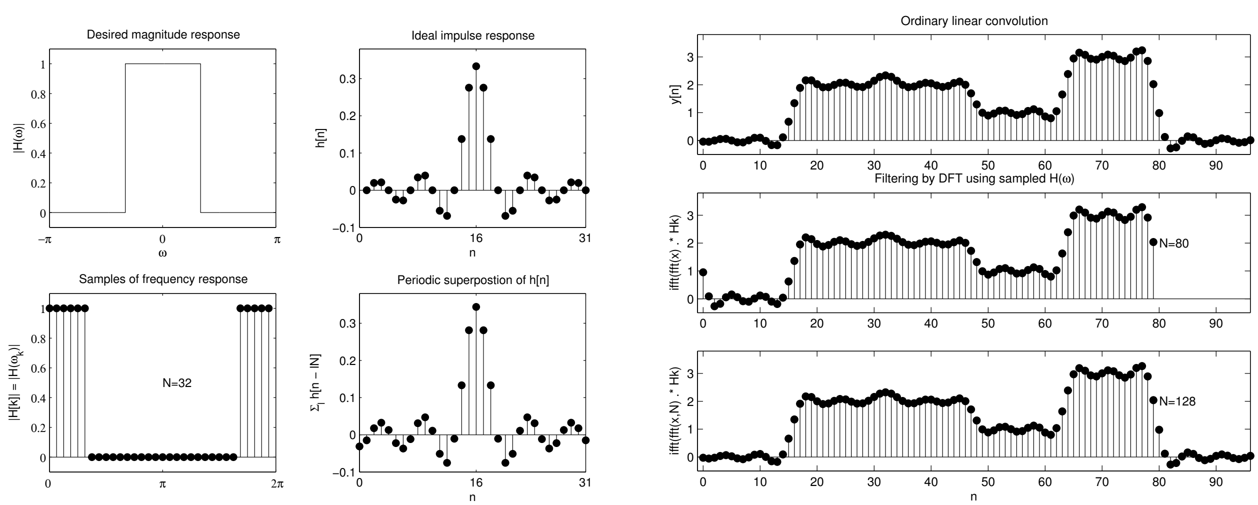
Suppose $N = 16$, what does `abs(Hv)` look like?

```
om = (0:15)*2*pi/15
Hv = (om <= pi/4) || (om >= 3*pi/4) \% ignoring phase here
```

which is [1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1]

Because $|\mathcal{H}(\omega)| = \begin{cases} \ldots, & \ldots \\ 1, & |\omega| \leq \pi/4 \\ 0, & \pi/4 < |\omega| \leq \pi \\ 0, & \pi < |\omega| < 7\pi/4 \\ 1, & 7\pi/4 \leq |\omega| \leq 9\pi/4 \\ \ldots, & \ldots. \end{cases}$

What is the catch? $y[n]$ is not exact. Why not? IDFT of $H[k]$ has length N, so $N + L - 1 > N$ so $y[n]$ *always* undersampled, no matter what N is, so always time-domain aliasing in $y[n]$. But if N is large enough relative to the *effective* length of $y[n]$, then time-domain aliasing can be made minimal. Again need $N > L + M_0 - 1$, where M_0 is the approximate length of $h[n]$.



5.4

Frequency analysis of signals using the DFT

Suppose you are working with an analog device (such as a microphone, or a pressure sensor in an automobile, etc.) that produces a signal $x_{\mathrm{a}}(t)$, and you would like to examine its frequency spectrum $X_{\mathrm{a}}(F)$.

Three options: analytical, analog, digital

Analytical.
If $x_{\mathrm{a}}(t)$ has a simple analytical form, such as $x_{\mathrm{a}}(t) = \mathrm{e}^{-t^2}$, then one can use integration or tables and FT properties to find $X_{\mathrm{a}}(F)$.

Analog spectrum analyzer

$$x_{\mathrm{a}}(t) \to \underset{\underset{\cos(F_0 t)}{\uparrow}}{\otimes} \to \boxed{\text{lowpass filter}} \to \boxed{\text{integrator: } \int |\cdot|^2 \,\mathrm{d}t} \to \boxed{\text{energy in band}}$$

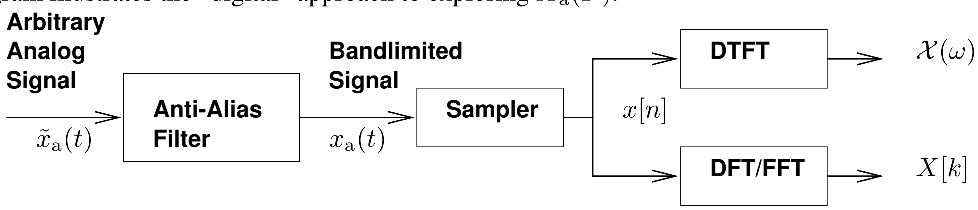
Or bank of bandpass filters. ***picture of overlapping responses*** $H_k(F)$

For a filter bank, resolution is limited by the number of "channels." (Sinusoids within the passband produce same response.)

For mixer approach, spectral resolution limited by lowpass bandwidth.
Narrow lowpass means higher resolution, but then longer transient response so longer delay before integrating, so slower readout.

Digital spectrum analyzer

The following diagram illustrates the "digital" approach to exploring $X_{\mathrm{a}}(F)$.



Consider first the hypothetical top path, *i.e.*, imagine that we could really compute $\mathcal{X}(\omega)$.

From sampling theorem (4.2.85),

$$\mathcal{X}(\omega) = \frac{1}{T_{\mathrm{s}}} \sum_{k=-\infty}^{\infty} X_{\mathrm{a}}\left(\frac{\omega/(2\pi) - k}{T_{\mathrm{s}}}\right).$$

For an ideal anti-alias filter, there is no overlap of the replicates in the above sum, so

$$X_{\mathrm{a}}(F) = T_{\mathrm{s}}\, \mathcal{X}\left(2\pi \frac{F}{F_{\mathrm{s}}}\right) \operatorname{rect}\left(\frac{F}{F_{\mathrm{s}}}\right).$$

Thus if we could compute $\mathcal{X}(\omega)$, then we could display $\mathcal{X}(\omega)$ with an appropriate horizontal axis as $X_{\mathrm{a}}(F)$!

In practice we cannot compute $\mathcal{X}(\omega)$ for all ω.

However, let us suppose that $x[n]$ is time-limited to $0, \ldots, L-1$, and then suppose we compute a N-point DFT of $x[n]$ for $N \geq L$.
Since $x[n]$ is time-limited:

$$X[k] = \mathcal{X}\left(\frac{2\pi}{N} k\right), \qquad k = 0, \ldots, N-1.$$

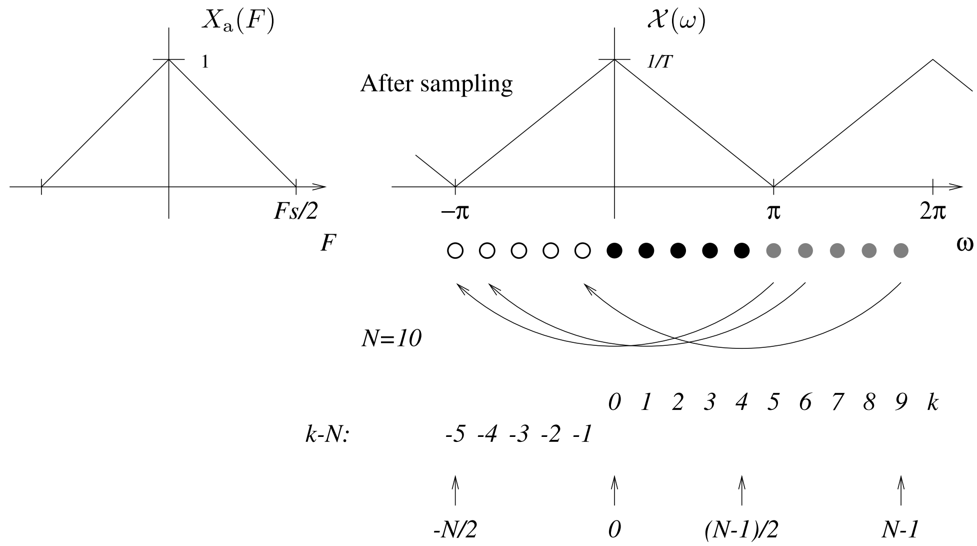
However, we have seen above that $\mathcal{X}(\omega)$ is related to $X_{\mathrm{a}}(F)$ for $|F| \leq F_{\mathrm{s}}/2$. So $X[k]$ is also related to $X_{\mathrm{a}}(F)$.

k related to ω_k by $\omega_k = \frac{2\pi}{N} k$
ω related to F by $\omega = 2\pi F/F_{\mathrm{s}}$ for $|F| \leq F_{\mathrm{s}}/2$, or $F = \frac{\omega}{2\pi} F_{\mathrm{s}}$ for $\omega \in [-\pi, \pi]$

Combining:

$$F_k = \begin{cases} \frac{k}{N} F_{\mathrm{s}}, & k = 0, \ldots, N/2 - 1 \\ \frac{k-N}{N} F_{\mathrm{s}}, & k = N/2, \ldots, N-1 \end{cases}$$

`fftshift`



Main point: display vs F_k most natural.

Spectral resolution

In principle one could choose to plot as many frequency values (N) as one would like of the DTFT, so the spectral resolution could be made arbitrarily fine. However, large N means more storage, longer sampling time, longer computation, etc.

In practice, we store L samples of $x[n]$, and then compute an N-point FFT of $x[n]$ to get $\left\{\mathcal{X}\left(\frac{2\pi}{N}k\right)\right\}_{k=0}^{N-1}$. How good is our frequency resolution as a function of N?

Analysis method. Realize that $x[n]$ is *not* exactly time limited since it is bandlimited by assumption after passing through the ideal anti-alias filter. (A signal cannot be both bandlimited and time limited.)

Consider $y[n] = x[n]\, w[n]$ where $w[n] = \begin{cases} 1, & n = 0, \ldots, L-1 \\ 0, & \text{otherwise,} \end{cases}$. How do $\mathcal{Y}(\omega)$ and $\mathcal{X}(\omega)$ relate? $\mathcal{Y}(\omega) = \mathcal{X}(\omega) * \mathcal{W}(\omega)$ where $\mathcal{W}(\omega) = \frac{\sin(\omega L/2)}{\sin(\omega/2)}\, e^{-\jmath\omega(L-1)/2}$ (periodic version of a sinc function). So $\mathcal{Y}(\omega)$ is a "blurred out" version of $\mathcal{X}(\omega)$, smeared by the FT of the rectangular window $x[n]$. This is called **spectral leakage**.

The width of $\mathcal{W}(\omega)$ is approximately $2\pi/L \triangleq \Delta\omega = \omega_k - \omega_{k-1}$ for a L-point DFT.

We would like to express this width in terms of the analog signal spectrum.

$\omega = 2\pi u/F_s$ so $\Delta F = F_s \Delta \frac{\omega}{2\pi} = \frac{F_s}{L} = \frac{1}{LT_s}$.

$$\boxed{\Delta F = \frac{1}{LT_s}}$$

so spectral width is the inverse of the total sampling time! longer time means finer spectral resolution.

Example: audio, suppose we would like 22Hz resolution. Then $22 = 1/(LT_s)$ so $L = F_s/22\text{Hz} = 44\text{kHz}/22\text{Hz} = 2000$ samples needed.

Interpolation / upsampling revisited

Suppose we are given the (stored) samples $x[n]$ of a band-limited analog signal $x_\mathrm{a}(t)$. Suppose we wish we could have $M-1$ samples in between each sample, *i.e.*, we would like to find a signal $y[n]$ such that

$$y[n] = \begin{cases} x[n/M], & n = 0, \pm M, \pm 2M, \cdots \\ \text{“intermediate values”}, & \text{otherwise.} \end{cases}$$

Time-domain approach

We could recover $x_\mathrm{a}(t)$ from $x[n]$ using the sinc-interpolation formula:

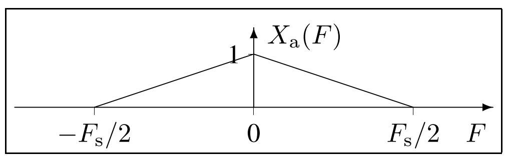
$$x_\mathrm{a}(t) = \sum_{l=-\infty}^{\infty} x[l] \operatorname{sinc}\left(\frac{t - lT_\mathrm{s}}{T_\mathrm{s}}\right),$$

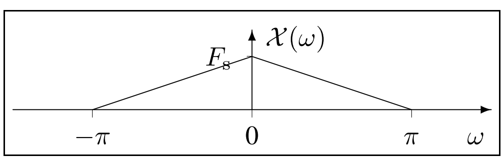
and then evaluate $x[n]$ at intermediate points:

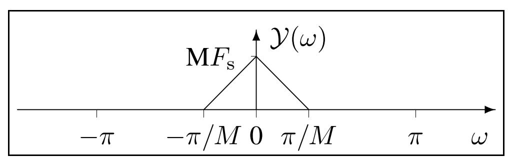
$$y[n] = x_\mathrm{a}\left(\frac{n}{M} T_\mathrm{s}\right) = \sum_{l=-\infty}^{\infty} x[l] \operatorname{sinc}(n/M - l)\,.$$

This is the “ultimate” upsampling method, but it requires excessive amounts of computation.

How do the spectra of $x[n]$ and $y[n]$ relate to $x_\mathrm{a}(t)$?







FFT-based approach

A more efficient way is to take an N-point DFT, and zero-pad in the frequency domain! Why zero pad? Because of the pictures of $\mathcal{X}(\omega)$ and $\mathcal{Y}(\omega)$ above!

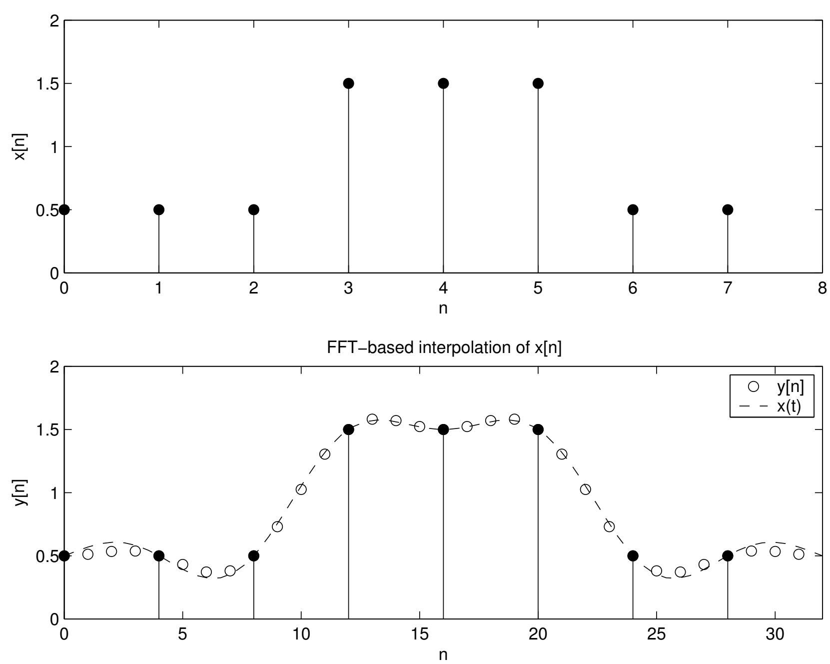
$$Y[k] = \left\{ X[0]\, X[1], \ldots, X[N/2-1], \frac{1}{2}\, X[N/2], [(M-1)N - 1 \text{ zeros}], \frac{1}{2}\, X[N/2], X[N/2+1], \ldots, X[N-1] \right\}.$$

Take MN-point IDFT of $Y[k]$ and scale by M. Yields “sinc-like” interpolation of $x[n]$, where $y[nM] = x[n]$.

In MATLAB this is performed using the `interpft` command as follows: `y = interpft(x, N);` where `N` is desired signal length, *e.g.*, `N = 3 * length(x)` for upsampling by 3.

Example.

```
% fig_interp.m
% illustrate FFT-based interpolation

N = 8;
n=0:N-1;
x = 0.5 + (n > N/4 & n < 3*N/4); % simply boxcar signal
clf, subplot(211)
stem(n, x, 'filled'), stem_fix
xlabel 'n', ylabel 'x[n]'
axis([0 N 0 2])

M = 4;
X = fft(x);
middle = X(N/2+1)/2;
Y = M * [X(1:N/2) middle zeros(1, (M-1)*N-1) middle X(N/2+2:end)];
y = reale(ifft(Y));

t = linspace(0, N, N*2*M);
xt = 0.5 + sinc(t-3) + sinc(t-4) + sinc(t-5);

subplot(212)
plot(0:length(y)-1, y, 'o', t*M, xt, '--')
legend('y[n]', 'x(t)')
xlabel 'n', ylabel 'y[n]'
axis([0 M*N 0 2])
title 'FFT-based interpolation of x[n]'

hold on
stem(n*M, x, 'filled'), stem_fix
hold off

% savefig 'fig_interp'
```



5.5

Summary

For a time-limited signal $x[n]$ with L samples, the N-point DFT is an invertible transform that computes samples of the DTFT $\mathcal{X}(\omega)$ of $x[n]$.

The DFT can be used for filtering, via zero-padding and multiplication of DFT coefficients. This approach is particularly useful because of the FFT.

$$x[n] \overset{N\text{-point periodic superposition}}{\longrightarrow} x_{\text{ps}}[n] = \sum_{l=-\infty}^{\infty} x[n - lN] \overset{\text{DTFS analysis}}{\longrightarrow} \{c_k\}_{k=0}^{N-1} \overset{\text{DTFS synthesis}}{\longrightarrow} x_{\text{ps}}[n]$$

$$x[n] \overset{\text{DTFT}}{\longrightarrow} \mathcal{X}(\omega) \overset{\text{Sample}}{\longrightarrow} \left\{ \mathcal{X}\left(\frac{2\pi}{N}k\right) \right\}_{k=0}^{N-1} \overset{\text{IDFT}}{\longrightarrow} x_{\text{ps}}[n]$$

$$x[n] \overset{\text{DFT}}{\longrightarrow} \left\{ \mathcal{X}\left(\frac{2\pi}{N}k\right) \right\}_{k=0}^{N-1} \overset{\text{IDFT}}{\longrightarrow} x[n \bmod N]$$

In time-limited case with $L \le N$,

$$X[k] = \mathcal{X}\left(\frac{2\pi}{N}k\right) = N c_k$$

and

$$x_{\text{ps}}[n] = x[n \bmod N], \; \forall n.$$

FT family tree diagram

DFT-based "real-time" filtering *skip*

Application: DFT-based removal of low-frequency "hum."

Analog approach: $\tilde{x}_{\text{a}}(t) \to \boxed{H_{\text{a}}(F)} \to y_{\text{a}}(t)$
with $H_{\text{a}}(F) = 1$ for $|F| \ge 100\text{Hz}$ ***picture***

Analog design inflexible. Can we do it digitally?

System:

$$\tilde{x}_{\text{a}}(t) \to \boxed{\text{anti-alias}} \to x_{\text{a}}(t) \to \boxed{\text{sampler}} \to x[n] \to \boxed{N\text{-point DFT}} \to \underset{\substack{\uparrow \\ H[k]}}{\otimes} \to \boxed{\text{IDFT}} \to y[n] \to \boxed{\text{D/A}} \to y_{\text{a}}(t)$$

Want $x[n] = h[n] * x[n]$ but get $y[n] = x[n] \; N \; h[n]$

Design issues
- anti-alias cutoff
- sampling frequency (relates to bandwidth of $\tilde{x}_{\text{a}}(t)$
- N (larger causes more lag)
- $H[k]$'s (filter)

Suppose $F_{\text{s}} = 1600\text{Hz}$, so $F_{\text{s}}/2 = 800\text{Hz}$.

Is is real time? Not exactly due to lag to buffer in N samples, compute DFT/Multiply/IDFT.