EI331 Signals and Systems

Lecture 13

Bo Jiang

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

April 9, 2019

Contents

Discrete Fourier Transform (DFT)

DTFS pair for N-periodic sequence

Analysis equation

$$\hat{x}[k] = \frac{1}{N} \sum_{n \in [N]} x[n] e^{-jk\frac{2\pi}{N}n}$$

Synthesis equation

$$x[n] = \sum_{k \in [N]} \hat{x}[k] e^{jk\frac{2\pi}{N}n}$$

DFT pair for finite sequence of length N

DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n}$$

Inverse DFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{jk\frac{2\pi}{N}n}$$

Both pairs of equations essentially the **same** up to constant factor $\frac{1}{N}$; efficient computation by Fast Fourier Transform (FFT)

DFT in Matrix Form

With $W_N = e^{-j\frac{2\pi}{N}}$ (note sign change from last lecture)

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \\ X[4] \\ X[5] \\ X[6] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^0 & W_8^3 & W_8^4 & W_8^6 \\ W_8^0 & W_8^3 & W_8^6 & W_8^1 & W_8^4 & W_8^7 & W_8^2 & W_8^5 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^5 & W_8^2 & W_8^7 & W_8^4 & W_8^1 & W_8^6 & W_8^3 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 & W_8^0 & W_8^6 & W_8^4 & W_8^2 \\ W_8^0 & W_8^7 & W_8^6 & W_8^5 & W_8^4 & W_8^3 & W_8^2 & W_8^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}$$

Direct matrix multiplication has complexity $O(N^2)$

FFT is divide-and-conquer algorithm (Cooley & Tukey 1965)

Fast Fourier Transform (FFT)

Divide and conquer

Assume $N = 2^M$ (radix 2). Divide x into two subsequences

$$x_e[n] = x[2n], \qquad n = 0, 1, \ldots, 2^{M-1} - 1$$
$$x_o[n] = x[2n+1], \quad n = 0, 1, \ldots, 2^{M-1} - 1$$

N-point DFT X of x. For $k = 0, 1, \ldots, 2^M - 1$,

$$\begin{aligned} X[k] &= \sum_{n=0}^{2^M-1} x[n] e^{-jk\frac{2\pi}{2^M}n} \\ &= \sum_{n=0}^{2^{M-1}-1} x_e[n] e^{-jk\frac{2\pi}{2^M}2n} + \sum_{n=0}^{2^{M-1}-1} x_o[n] e^{-jk\frac{2\pi}{2^M}(2n+1)} \\ &= \sum_{n=0}^{2^{M-1}-1} x_e[n] e^{-jk\frac{2\pi}{2^{M-1}}n} + e^{-jk\frac{2\pi}{2^M}} \sum_{n=0}^{2^{M-1}-1} x_o[n] e^{-jk\frac{2\pi}{2^{M-1}}n} \end{aligned}$$

Fast Fourier Transform (FFT)

Divide and conquer

Divide X into two halves. For $k = 0, 1, \ldots, 2^{M-1} - 1$

$$X[k] = \underbrace{\sum_{n=0}^{2^{M-1}-1} x_e[n] e^{-jk\frac{2\pi}{2^{M-1}}n}}_{\frac{N}{2}\text{-point DFT of } x_e} + e^{-jk\frac{2\pi}{2^M}} \underbrace{\sum_{n=0}^{2^{M-1}-1} x_o[n] e^{-jk\frac{2\pi}{2^{M-1}}n}}_{\frac{N}{2}\text{-point DFT of } x_o}$$

$$X[2^{M-1} + k] = \sum_{n=0}^{2^{M-1}-1} x_e[n] e^{-jk\frac{2\pi}{2^{M-1}}n} - e^{-jk\frac{2\pi}{2^M}} \sum_{n=0}^{2^{M-1}-1} x_o[n] e^{-jk\frac{2\pi}{2^{M-1}}n}$$

$\frac{N}{2}$-point DFT of x_e $\qquad$ $\frac{N}{2}$-point DFT of x_o

Recursive algorithm

$$\mathrm{DFT}(x)[k] = \mathrm{DFT}(x_e)[k] + e^{-jk\frac{2\pi}{N}}\, \mathrm{DFT}(x_o)[k]$$

$$\mathrm{DFT}(x)[2^{M-1} + k] = \mathrm{DFT}(x_e)[k] - e^{-jk\frac{2\pi}{N}}\, \mathrm{DFT}(x_o)[k]$$

Fast Fourier Transform (FFT)

Naive implementation of radix-2 FFT for $N = 2^M$

```python
import numpy as np

# compute DFT of sequences of length N=2^M
def FFT(x):
    N = x.size

    # base case, compute DFT directly
    if N == 2:
        return np.array([x[0] + x[1], x[0] - x[1]])

    # recursively compute N/2 point DFT of even and odd subsquences
    Xe = FFT(x[0::2])
    Xo = FFT(x[1::2])

    # compute phase factor exp(-jk(2*pi/N))
    K = np.arange(N/2)
    phase = np.exp(-1j * K * 2 * np.pi / N)
    Xo_phase = phase * Xo

    # compute N ponint DFT of x
    X = np.append(Xe + Xo_phase, Xe - Xo_phase)
    return X
```

Fast Fourier Transform (FFT)

Time complexity

Denote by $T(N)$ time complexity of N-point FFT

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

Let $\tilde{T}(M) = T(2^M)$ and assume $O(N) = cN$ for contant c,

$$\tilde{T}(M) = 2\tilde{T}(M-1) + c2^M \implies \tilde{T}(M) = O(M2^M)$$

or

$$T(N) = O(N \log N)$$

Much lower than $O(N^2)$ for direct matrix multiplication!

Contents

Frequency Response

Recall response of DT LTI system to exponential input z^n

$$T\left(\sum_k a_k z_k^n\right) = \sum_k a_k H(z_k) z_k^n$$

where $H(s)$ is system function

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n}$$

When restricted to $z = e^{j\omega}$, $H(e^{j\omega})$ as function of ω is called frequency response of the system

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n}$$

Frequency Response

Periodic input x in Fourier series representation

$$x[n] = \sum_{k\in[N]} \hat{x}[k] e^{jk\omega_0 n}, \quad \text{where } \omega_0 = \frac{2\pi}{N}.$$

Output of LTI system with frequency response $H(e^{j\omega})$

$$y[n] = T\left(\sum_{k\in[N]} \hat{x}[k] e^{jk\omega_0 n}\right) = \sum_{k\in[N]} H(e^{jk\omega_0}) \hat{x}[k] e^{jk\omega_0 n}$$

periodic with same periodic, Fourier coefficients related by

$$\hat{y}[k] = H(e^{jk\omega_0}) \hat{x}[k]$$

Filtering

Filtering changes relative amplitudes of frequency components or eliminates some frequency components entirely

Frequency-shaping vs frequency-selective filters as in CT case

LTI systems as filters

- cannot create new frequency components
- can only scale magnitudes or shift phases of existing components

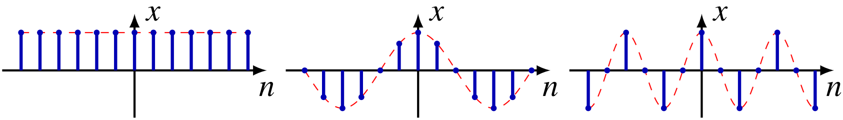
Examples of nonlinear filter

- max filter: $y[n] = \max_{-n_1 \le k \le n_2} x[n+k]$
- median filter: $y[n] = \text{median}\{x[n-n_1], \ldots, x[n+n_2]\}$

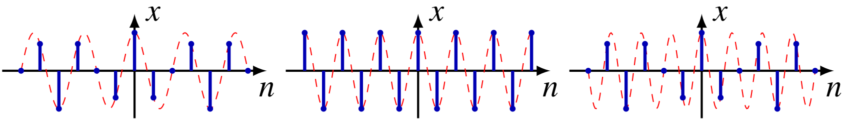
Recall for DT signals, suffices to consider frequencies on an interval of length 2π, e.g. $[0, 2\pi)$ or $(-\pi, \pi]$

High vs. Low Frequencies for DT Signals

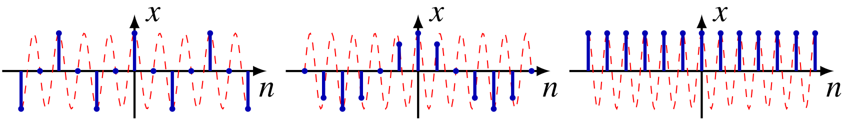
High frequencies around $(2k+1)\pi$, low frequencies around $2k\pi$



$\phi_N^0[n] = \cos(0 \cdot n) = 1$ $\phi_N^1[n] = \cos(\pi n/4)$ $\phi_N^2[n] = \cos(\pi n/2)$



$\phi_N^3[n] = \cos(3\pi n/4)$ $\phi_N^4[n] = \cos(\pi n)$ $\phi_N^5[n] = \cos(5\pi n/4)$

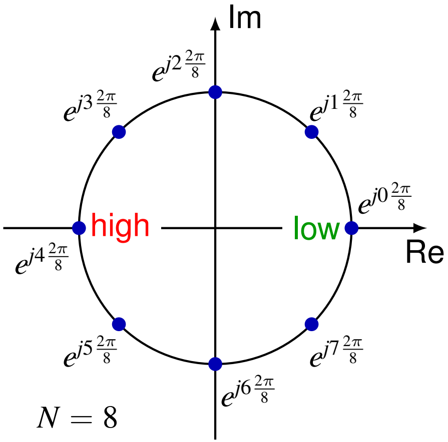


$\phi_N^6[n] = \cos(3\pi n/2)$ $\phi_N^7[n] = \cos(7\pi n/4)$ $\phi_N^8[n] = \cos(2\pi n) = 1$

DT Frequencies

Discrete frequencies of N-periodic signals

- evenly spaced points on unit circle
- low frequencies close to 1; high frequencies close to -1

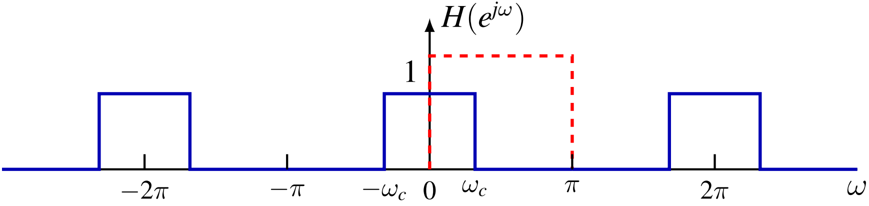


Ideal Frequency-selective Filters

Ideal lowpass filter

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

ω_c: cutoff frequency

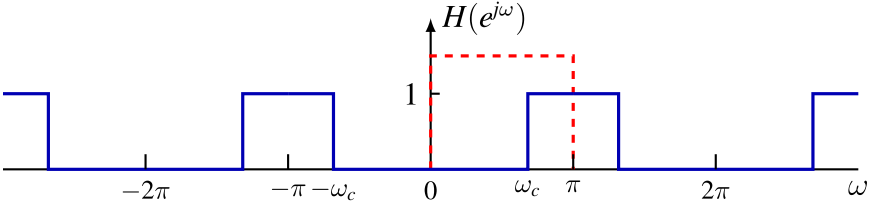


Ideal Frequency-selective Filters

Ideal highpass filter

$$H(e^{j\omega}) = \begin{cases} 1, & \omega_c \le |\omega| \le \pi \\ 0, & |\omega| < |\omega_c| \end{cases}$$

ω_c: cutoff frequency

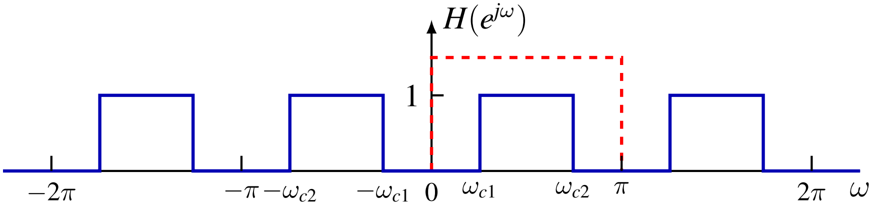


Ideal Frequency-selective Filters

Ideal bandpass filter

$$H(e^{j\omega}) = \begin{cases} 1, & \omega_{c1} \le |\omega| \le \omega_{c2} \\ 0, & |\omega| < \omega_{c1} \text{ or } \omega_{c2} < |\omega| \le \pi \end{cases}$$

ω_{c1}: lower cutoff frequency
ω_{c2}: upper cutoff frequency



First-order Recursive DT Filters

$$y[n] - ay[n-1] = x[n]$$

For input $x[n] = e^{j\omega n}$, output $y[n] = H(e^{j\omega})e^{j\omega n}$

Frequency response (well-defined if $|a| < 1$)

$$H(e^{j\omega}) = \frac{1}{1 - ae^{-j\omega}}, \quad |a| < 1$$

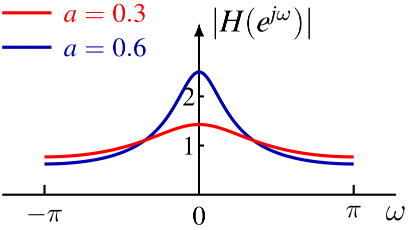
For $a = |a|e^{j\phi}$,

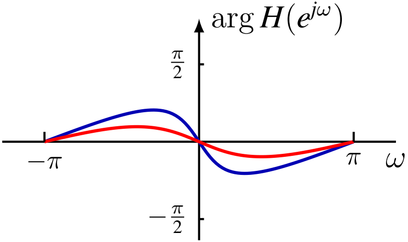
$$|H(e^{j\omega})| = \frac{1}{\sqrt{1 + |a|^2 - 2|a|\cos(\omega - \phi)}}$$

$$\arg H(e^{j\omega}) = \arctan \frac{-|a|\sin(\omega - \phi)}{1 - |a|\cos(\omega - \phi)}$$

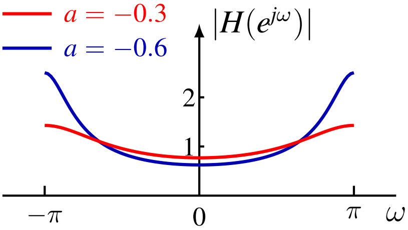
First-order Recursive DT Filters

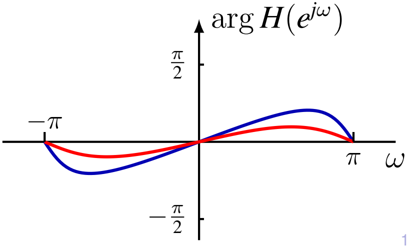
For $a > 0$, lowpass filter (exponential smoothing)





For $a < 0$, highpass filter





First-order Recursive DT Filters

Impulse response (IIR filter)

$$h[n] = a^n u[n]$$

$$H(e^{j\omega}) = \sum_{n=0}^{\infty} (ae^{-j\omega})^n = \frac{1}{1 - ae^{-j\omega}}$$

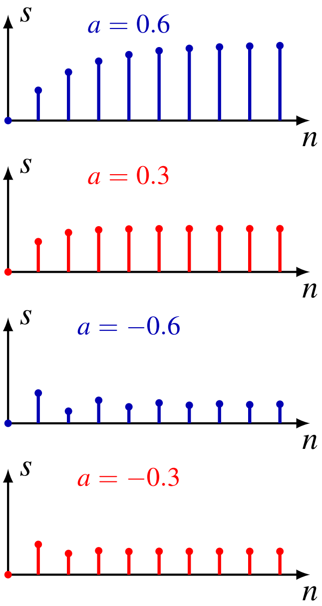
Need $|a| < 1$ for convergence

Step response

$$s[n] = (h * u)[n] = \frac{1 - a^{n+1}}{1 - a} u[n]$$

Tradeoff

- larger $|a|$, narrower passband, slower response
- smaller $|a|$, faster response, broader passband



Moving Average as Lowpass Filter

$$y[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n-k]$$

Impulse response (FIR filter)

$$h[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} \delta[n-k]$$

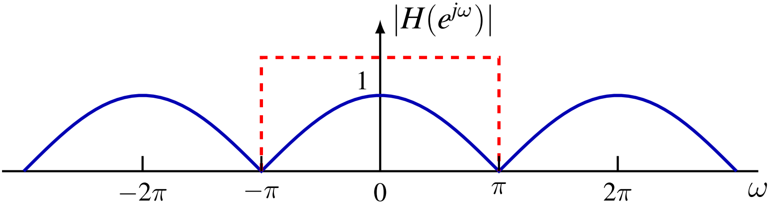
Frequency response

$$H(e^{j\omega}) = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} e^{-jk\omega} = \frac{e^{j\frac{M_1 - M_2}{2}\omega}}{M_1 + M_2 + 1} \frac{\sin(\frac{M_1+M_2+1}{2}\omega)}{\sin\frac{\omega}{2}}$$

Moving Average as Lowpass Filter

$M_1 = 0$, $M_2 = 1$,

$$y[n] = \frac{1}{2}(x[n] + x[n-1])$$
$$h[n] = \frac{1}{2}(\delta[n] + \delta[n-1])$$
$$H(e^{j\omega}) = e^{-j\frac{\omega}{2}} \cos\frac{\omega}{2}$$



Verify $y = x$ if $x = Ke^{j0 \cdot n}$ and $y = 0$ if $x = Ke^{j\pi n} = K(-1)^n$.

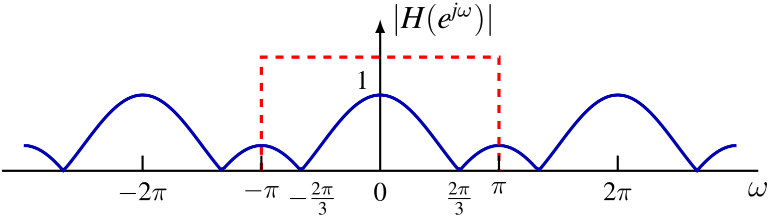
Moving Average as Lowpass Filter

$M_1 = M_2 = 1$,

$$y[n] = \frac{1}{3}(x[n+1] + x[n] + x[n-1])$$

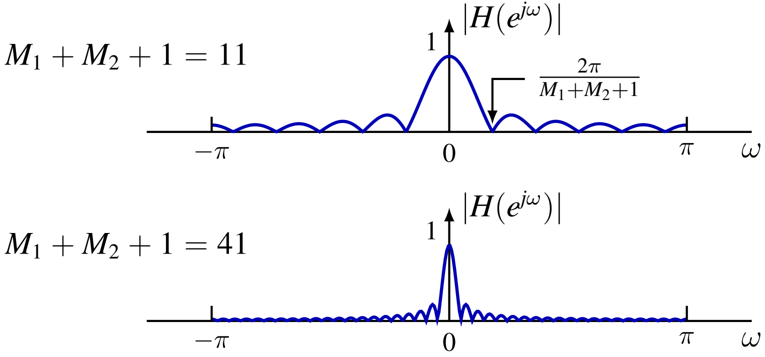
$$h[n] = \frac{1}{3}(\delta[n+1] + \delta[n] + \delta[n-1])$$

$$H(e^{j\omega}) = \frac{\sin(\frac{3}{2}\omega)}{3\sin\frac{\omega}{2}} = \frac{1}{3} + \frac{2}{3}\cos\omega$$



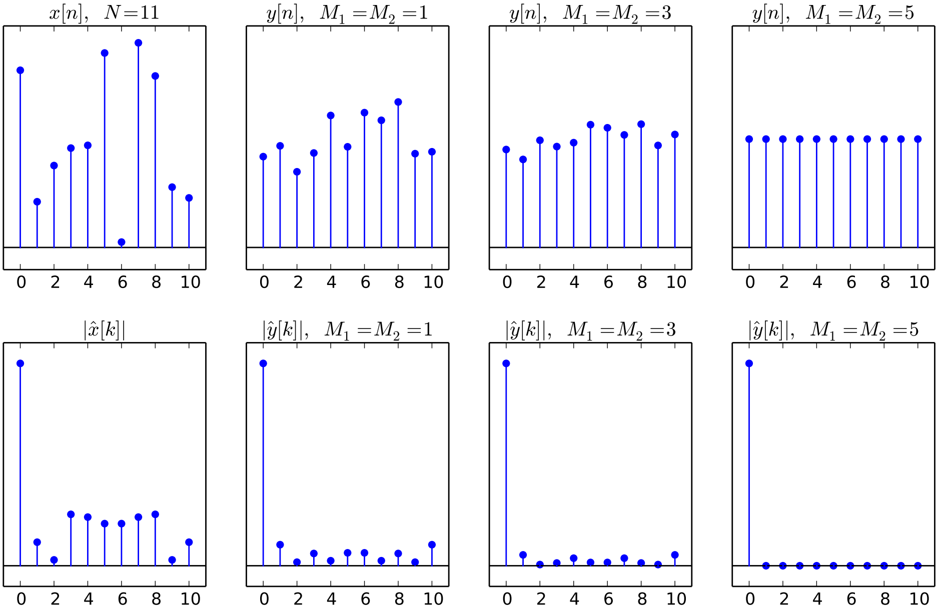
Moving Average as Lowpass Filter

$$H(e^{j\omega}) = \frac{e^{j\frac{M_1 - M_2}{2}\omega}}{M_1 + M_2 + 1} \frac{\sin(\frac{M_1+M_2+1}{2}\omega)}{\sin\frac{\omega}{2}}$$



Larger $M_1 + M_2$, narrower passband, smoother output

Moving Average as Lowpass Filter



Moving Average as Lowpass Filter

noncausal $$y_1[n] = \frac{1}{2M+1} \sum_{k=-M}^{M} x[n-k]$$

causal $$y_2[n] = \frac{1}{2M+1} \sum_{k=0}^{2M} x[n-k]$$

Note $y_2 = \tau_M y_1$

For real-time system

- noncausal version not realizable
- causal version realizable
- larger M, narrower passband, smoother output, but longer delay, more sluggish response

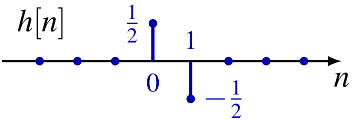
First Difference as Highpass Filter

Scaled first difference

$$y[n] = \frac{1}{2}(x[n] - x[n-1])$$

Impulse response (FIR filter)

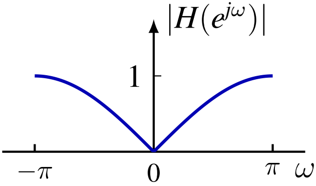
$$h[n] = \frac{1}{2}(\delta[n] - \delta[n-1])$$



Frequency response

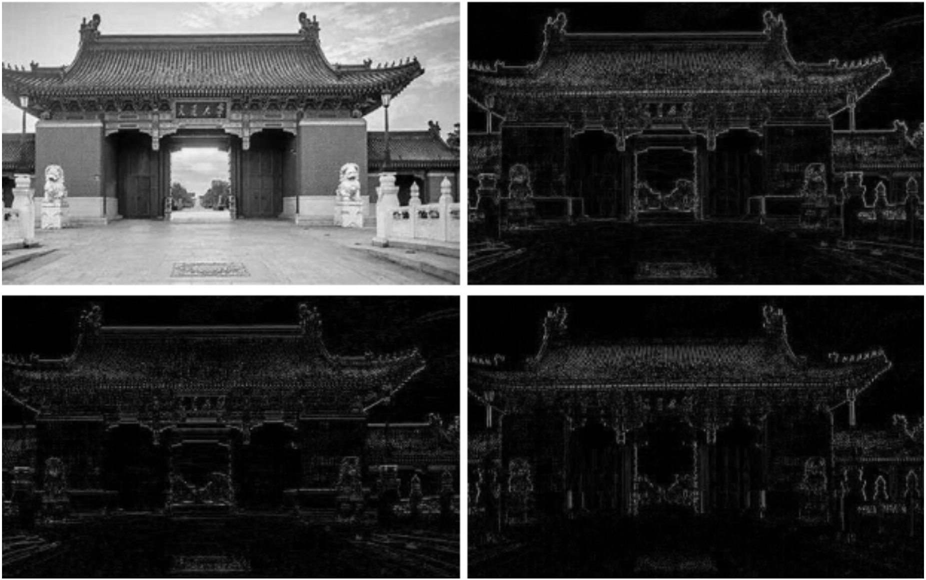
$$H(e^{j\omega}) = \frac{1}{2}(1 - e^{-j\omega}) = je^{-j\frac{\omega}{2}} \sin\frac{\omega}{2}$$

$$|H(e^{j\omega})| = \left|\sin\frac{\omega}{2}\right|$$



Verify $y = 0$ if $x = Ke^{j0\cdot n}$ and $y = x$ if $x = Ke^{j\pi n} = K(-1)^n$.

First Difference for Edge Detection

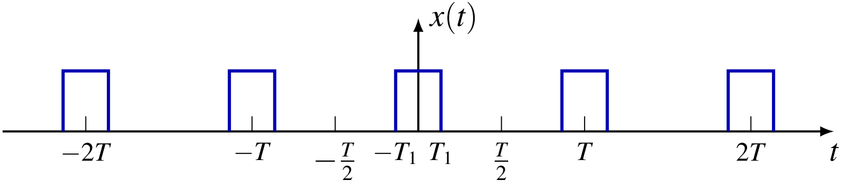


Contents

Motivating Example: Periodic Square Wave

In one period,

$$x_T(t) = \begin{cases} 1, & |t| < T_1 \\ 0, & T_1 < |t| < T/2 \end{cases}$$



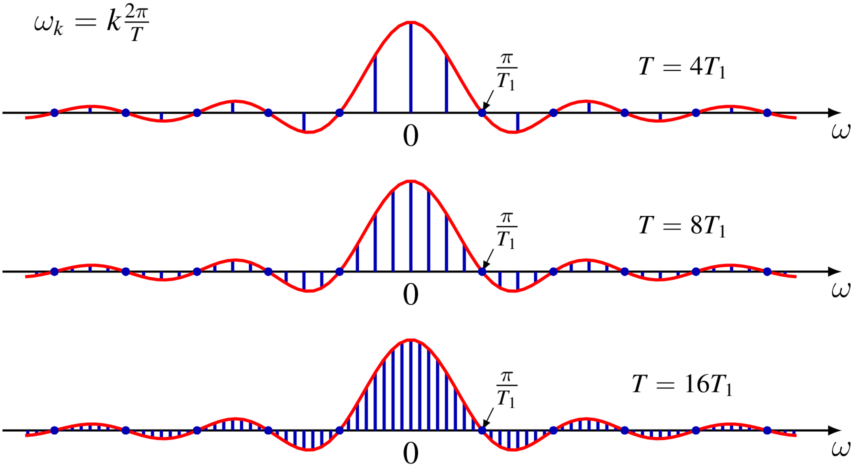
Frequency component at $\omega_k = k\omega_0$ satisfies

$$T\hat{x}_T[k] = \frac{2\sin(k\omega_0 T_1)}{k\omega_0} = \frac{2\sin(\omega T_1)}{\omega}\bigg|_{\omega = k\omega_0}$$

$T\hat{x}_T[k]$ is value of envelope $X(j\omega) \triangleq \frac{2\sin(\omega T_1)}{\omega}$ sampled at $\omega_k = k\omega_0$

Motivating Example: Periodic Square Wave

$X(j\omega_k)$ for fixed T_1 and different T



As $T \to \infty$, discrete frequencies sampled more densely

Motivating Example: Periodic Square Wave

As $T \to \infty$, $x_T(t) \to x(t) \triangleq u(t + \frac{T_2}{2}) - u(t - \frac{T_1}{2})$, rectangular pulse

$$x_T(t) = \sum_{k=-\infty}^{\infty} \hat{x}_T[k] e^{j\omega_k t} = \sum_{k=-\infty}^{\infty} \frac{1}{T} X(j\omega_k) e^{j\omega_k t} \qquad (T\hat{x}[k] = X(j\omega_k))$$

$$= \sum_{k=-\infty}^{\infty} \frac{\omega_0}{2\pi} X(j\omega_k) e^{j\omega_k t} \qquad (\omega_0 = \frac{2\pi}{T})$$

$$= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} X(j\omega_k) e^{j\omega_k t} \Delta\omega \qquad (\Delta\omega = \omega_0)$$

$$\to \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega \qquad (\Delta\omega = \omega_0 \to 0)$$

Thus

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$$

Motivating Example: Periodic Square Wave

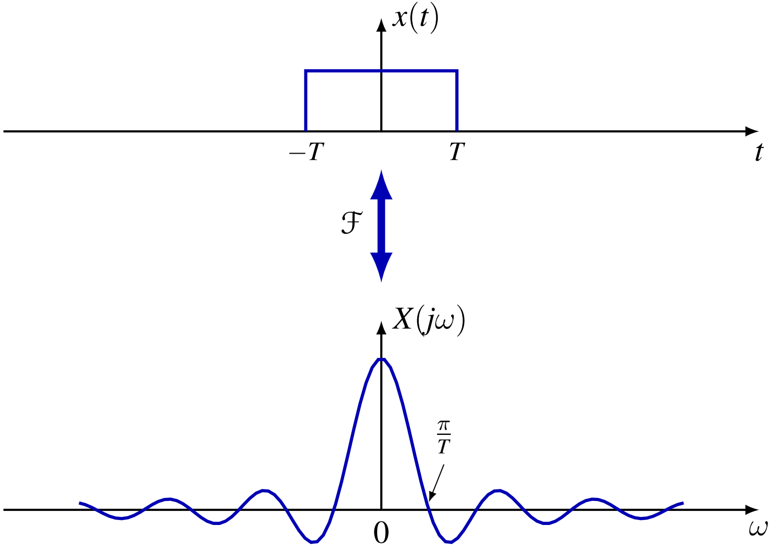
For envelope $X(j\omega)$,

$$
\begin{aligned}
X(j\omega_k) &= T\hat{x}_T[k] \\
&= \int_{-\frac{T}{2}}^{\frac{T}{2}} x_T(t)e^{-j\omega_k t}dt \\
&= \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{-j\omega_k t}dt \qquad (x_T(t) = x(t) \text{ for } |t| \le T/2) \\
&= \int_{-\infty}^{\infty} x(t)e^{-j\omega_k t}dt \qquad (x(t) = 0 \text{ for } |t| > T/2)
\end{aligned}
$$

so

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

Motivating Example: Periodic Square Wave



CT Fourier Transform of Aperiodic Signals

For aperiodic signal x with $\operatorname{supp} x \subset [-T_1, T_1]$, define periodic extension with period $T > 2T_1$,

$$x_T(t) = \sum_{k=-\infty}^{\infty} x(t - kT)$$

Then

$$x(t) = x_T(t), \quad |t| < \frac{T}{2}$$

As $T \to \infty$,

$$x_T(t) \to x(t), \quad \forall t \in \mathbb{R}$$

x_T has Fourier series representation

$$x_T(t) = \sum_{k=-\infty}^{\infty} \hat{x}_T[k] e^{jk\omega_0 t}, \quad \omega_0 = \frac{2\pi}{T}$$

CT Fourier Transform of Aperiodic Signals

Define

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

$X(j\omega)$ is envelope of $T\hat{x}_T[k]$,

$$\hat{x}_T[k] = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x_T(t)e^{-jk\omega_0 t}dt = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{-jk\omega_0 t}dt$$
$$= \frac{1}{T}\int_{-\infty}^{\infty} x(t)e^{-jk\omega_0 t}dt = \frac{1}{T}X(jk\omega_0) = \frac{\omega_0}{2\pi}X(jk\omega_0)$$

so

$$x(t) = \lim_{T\to\infty} x_T(t) = \lim_{T\to\infty} \sum_{k=-\infty}^{\infty} \frac{\omega_0}{2\pi}X(jk\omega_0)e^{jk\omega_0 t}$$
$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

CT Fourier Transform Pair

Fourier transform (analysis equation)

$$X(j\omega) = \mathcal{F}(x)(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

$X(j\omega)$ called spectrum of $x(t)$

Inverse Fourier transform (synthesis equation)

$$x(t) = \mathcal{F}^{-1}(X)(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

Superposition of complex exponentials at **continuum** of frequencies; frequency ω has “amplitude” $X(j\omega)\frac{d\omega}{2\pi}$