DTFT and Digital Filters

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 11
(with material from Lathi and Cannon (Discrete Systems))

elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

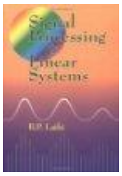
April 11, 2016

http://elec3004.com

Lecture Schedule:

Week	Date	Lecture Title
1	29-Feb	Introduction
	3-Mar	Systems Overview
2	7-Mar	Systems as Maps & Signals as Vectors
	10-Mar	Data Acquisition & Sampling
3	14-Mar	Sampling Theory
	17-Mar	Antialiasing Filters
4	21-Mar	Discrete System Analysis
	24-Mar	Convolution Review
	28-Mar	Holiday
	31-Mar	
5	4-Apr	Frequency Response & Filter Analysis
	7-Apr	Filters
6	11-Apr	Digital Filters
	14-Apr	Digital Filters
7	18-Apr	Digital Windows
	21-Apr	FFT
8	25-Apr	Holiday
	28-Apr	Feedback
9	3-May	Introduction to Feedback Control
	5-May	Servoregulation/PID
10	9-May	Introduction to (Digital) Control
	12-May	Digitial Control
11	16-May	Digital Control Design
	19-May	Stability
12	23-May	Digital Control Systems: Shaping the Dynamic Response & Estimation
	26-May	Applications in Industry
13	30-May	System Identification & Information Theory
	2-Jun	Summary and Course Review

Follow Along Reading:



B. P. Lathi
Signal processing and linear systems
1998
TK5102.9.L38 1998

Today

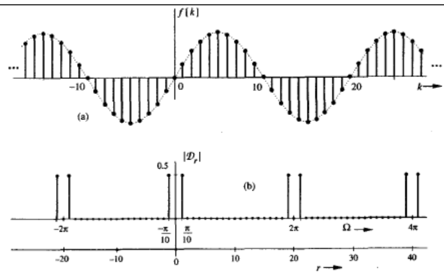
- Chapter 10
  (**Discrete-Time System Analysis Using the *z*-Transform**)
  - § 10.3 Properties of DTFT
  - § 10.5 Discrete-Time Linear System analysis by DTFT
  - § 10.7 Generalization of DTFT to the $\mathcal{Z}$–Transform

- Chapter 12
  (**Frequency Response and Digital Filters**)
- § 12.1 Frequency Response of Discrete-Time Systems
- § 12.3 Digital Filters
- § 12.4 Filter Design Criteria
- § 12.7 Nonrecursive Filters

Next Time

DTFT

Digital Filters → DTFT



Lathi, p. 621

- First Thought:

FFT → Crop → Un-FFT → Go!

- How to get DTFT? FFT?
- Slightly Naïve ∵
  - H(ω) cannot be exactly zero over any *band* of frequencies (**Paley-Wiener Theorem)**

DTFT is a Convolution

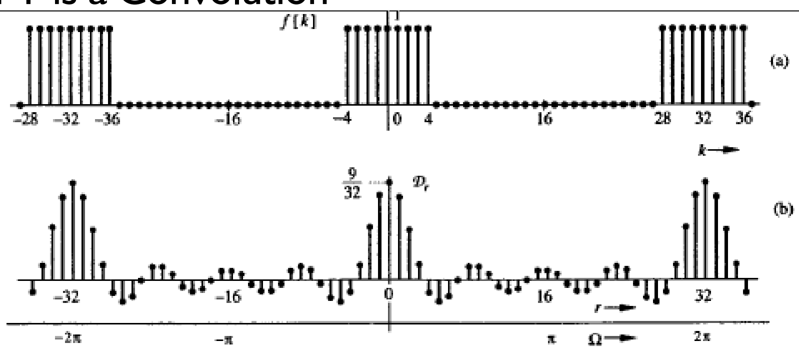


Fig. 10.2 Periodic sampled gate pulse and its Fourier spectrum.

Lathi, p. 623

- The frequency response is limited to 2π
- DTFT is a convolution responses in time domain…

$$\underbrace{\mathcal{F}\{x * h\}}_{Y(\omega)} = \underbrace{\mathcal{F}\{x\}}_{X(\omega)} \cdot \underbrace{\mathcal{F}\{h\}}_{H(\omega)}$$

$$y[n] = x[n] * h[n] = \mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\},$$

DTFT → z-Transform

The above results motivate the definitions of the z transform, the discrete-time Fourier transform (DTFT), and the discrete Fourier series (DFS) to be presented in this chapter and the next. In particular, if the basis functions for the input can be enumerated as

$$\phi_k[n] = z_k^n,$$

that is, if $x(t)$ can be expressed in the form of Eq. (6.1.1) as

$$x[n] = \sum_k a_k z_k^n, \quad (6.1.10)$$

then the corresponding output is simply, from Eqs. (6.1.2) and (6.1.8),

$$y[n] = \sum_k a_k H(z_k) z_k^n. \quad (6.1.11)$$

The discrete Fourier series for periodic signals is of this form, with $z_k = e^{j2\pi k/N}$. If, on the other hand, the required basis functions cannot be enumerated, we must utilize the continuum of functions $\phi[n] = z^n$ to represent $x[n]$ and $y[n]$ in the form of integrals. When z is restricted to have unit magnitude (that is, $z = e^{j\Omega}$), the resulting representation is called the *discrete-time Fourier transform*, while if z is an arbitrary complex variable, the full *z-transform* representation results.

The Discrete-Time Fourier Transform

- Synthesis:

The function $X(e^{j\Omega})$ defined by

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \quad (7.1.1)$$

(if it converges) is called the *discrete-time Fourier transform* (*DTFT*) of the signal $x[n]$. In particular, if the region of convergence for the z transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

includes the unit circle, then the DTFT equals $X(z)$ evaluated on the unit circle, that is,

$$X(e^{j\Omega}) = X(z)\big|_{z=e^{j\Omega}}. \quad (7.1.2)$$

The Discrete-Time Fourier Transform

- Analysis/Inverse:

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(e^{j\Omega})e^{j\Omega n}\, d\Omega.$$

- *x[n]* is the (limiting) sum of sinusoidal components of the form $\left[\frac{1}{2\pi} X\left(e^{j\Omega}\right)d\Omega\right] e^{j\Omega n}$

- Together: Forms the DTFT Pair

The Discrete-Time Fourier Transform

- Ex:

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

has the z transform

$$X(z) = \frac{1}{1 - az^{-1}}, \qquad |z| > |a|,$$

and thus $X(e^{j\Omega})$ exists for $|a| < 1$ because the ROC then contains the unit circle. Specifically,

$$X(e^{j\Omega}) = \frac{1}{1 - ae^{-j\Omega}}, \qquad |a| < 1. \qquad (7.1.8)$$

The corresponding *magnitude spectrum* $|X(e^{j\Omega})|$ and *phase spectrum* $\angle X(e^{j\Omega})$ are shown in Fig. 6.8. Clearly, from the defining sum in Eq. (7.1.1), the DTFT of $x[n]$ does not converge for $|a| > 1$, and we defer until later the case of $|a| = 1$.

On the other hand, the anticausal exponential

$$w[n] = -a^n u[-n-1]$$

has the z transform

$$W(z) = \frac{1}{1 - az^{-1}}, \qquad |z| < |a|,$$

and thus $W(e^{j\Omega})$ exists for $|a| > 1$, but not for $|a| < 1$. That is,

$$W(e^{j\Omega}) = \frac{1}{1 - ae^{-j\Omega}}, \qquad |a| > 1. \qquad (7.1.9)$$

Again the case of $|a| = 1$ is deferred until later.

The Discrete-Time Fourier Transform

- Observe:
  "Kinship Of Difference Equations To Differential Equations"

$$\frac{dy}{dt} + cy(t) = x(t) \qquad (3.15a)$$

Consider uniform samples of $x(t)$ at intervals of T seconds. As usual, we use the notation $x[n]$ to denote $x(nT)$, the nth sample of $x(t)$. Similarly, $y[n]$ denotes $y[nT]$, the nth sample of $y(t)$. From the basic definition of a derivative, we can express Eq. (3.15a) at $t = nT$ as

$$\lim_{T\to 0} \frac{y[n] - y[n-1]}{T} + cy[n] = x[n]$$

Clearing the fractions and rearranging the terms yields (assuming nonzero, but very small T)

$$y[n] + \alpha y[n-1] = \beta x[n] \qquad (3.15b)$$

where

$$\alpha = \frac{-1}{1+cT} \quad \text{and} \quad \beta = \frac{T}{1+cT}$$

We can also express Eq. (3.15b) in advance operator form as

$$y[n+1] + \alpha y[n] = \beta x[n+1] \qquad (3.15c)$$

The Discrete-Time Fourier Transform

- Ex(2): The DTFT of the real sinusoid

$$x[n] = \sin \Omega_0 n = \frac{1}{2j}(e^{j\Omega_0 n} - e^{-j\Omega_0 n})$$

is simply

$$X(e^{j\Omega}) = 2\pi\left(\frac{1}{2j}\right)[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$$

$$= -j\pi[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$$

for $|\Omega|, |\Omega_0| \le \pi$, while that of the cosine signal

$$y[n] = \cos \Omega_0 n = \tfrac{1}{2}(e^{j\Omega_0 n} + e^{-j\Omega_0 n})$$

is likewise

$$Y(e^{j\Omega}) = 2\pi(\tfrac{1}{2})[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)]$$

$$= \pi[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)].$$

In addition, the DTFT pair for the dc signal $x[n] = 1$ is simply

$$1 \leftrightarrow 2\pi\,\delta(\Omega), \qquad |\Omega| \le \pi,$$

as opposed to the dual relationship

$$\delta[n] \leftrightarrow 1, \qquad \text{all } \Omega.$$

Now: (digital) **Filters!**

➔ Digital Filters

- Wikipedia Says:

A **digital filter** is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal.

- Basically we have a transfer function or
  … a difference equation

In the Z-domain:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}}$$

- This is a recursive form with inputs (Numerator) and outputs (Denominator)
  ➔ "IIR infinite impulse response" behaviour
- If the denominator is made equal to unity (i.e. no feedback)

➔ then this becomes an FIR or finite impulse response filter.

Flashback: LCC Difference Equations

- Start with: $\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k}$
  - **N**: Highest derivative of the output y(t)
  - **M**: Highest derivative of the input x(t)
- $\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$
- Solution strategy: Find the particular solution to: $\sum_{k=0}^{N} a_k y[n-k] = 0$
- Leads to a Recursive Equation:

$$y[n-k] = \frac{1}{a_0}\{\sum_{k=0}^{M} b_k x[n-k] - \sum_{k=0}^{N} a_k y[n-k]\}\backslash$$

➔ If N=0: Non-Recursive Solution:

$$y[n] = \sum_{k=0}^{M} \left(\frac{b_k}{a_0}\right) x[n-k]$$

Two Types of Systems

- Linear shift-invariant:

$$y = \sum_{k=0}^{N-1} u[k] Z^k h$$

Z: Shift operator

$$Z \cdot [u_0, u_1, u_2, u_3, \ldots, u_{n-1}]^T = [u_{n-1}, u_0, u_1, u_2, \ldots, u_{n-2}]^T$$

- Linear time-invariant system

$$y = \sum_{k=-\infty}^{\infty} u[k] R^k h$$

R: Unit delay operator

$$R \cdot [\ldots, u_0, u_1, u_2, u_3, \ldots]^T = [\ldots, u_{-1}, u_0, u_1, \ldots]^T$$

Impulse Response of Both Types

$$y[n] = \frac{1}{2}u[n-1] + \frac{1}{2}u[n]$$

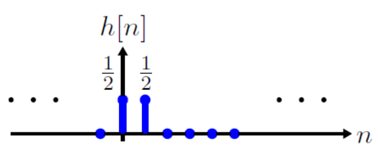
$$y[-1] = 0$$
$$y[0] = \frac{1}{2}$$
$$y[1] = \frac{1}{2}$$
$$y[2] = 0$$
$$\vdots$$

$$y[n] = \frac{1}{2}y[n-1] + u[n]$$

$$h[-1] = 0$$
$$h[0] = 1$$
$$h[1] = \frac{1}{2}$$
$$h[2] = \frac{1}{4}$$
$$\vdots$$
$$h[n] = \begin{cases} 0 & n < 0 \\ \left(\frac{1}{2}\right)^n & n \geq 0 \end{cases}$$

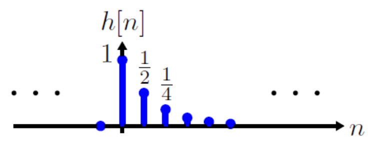
Impulse Response of Both Types

$$y[n] = \frac{1}{2}u[n-1] + \frac{1}{2}u[n]$$



'Finite impulse response" (FIR)

$$y[n] = \frac{1}{2}y[n-1] + u[n]$$



"Infinite impulse response" (IIR)

➔ Digital Filters Types

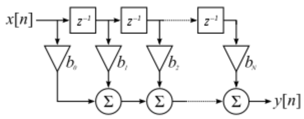
FIR

From H(z):

$$\rightarrow H(\omega) = h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega}$$
$$= \sum_{t=0}^{n-1} h_t \cos t\omega - i \sum_{t=0}^{n-1} h_t \sin t\omega$$

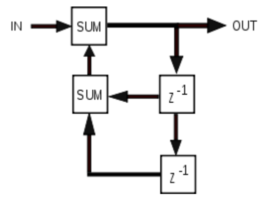
→ Filter becomes a "multiply, accumulate, and delay" system:

$$y(t) = \sum_{\tau=0}^{n-1} h_\tau u(t-\tau)$$
$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$



IIR

- Impulse response function that is non-zero over an infinite length of time.



FIR Properties

- Require no feedback.
- Are inherently stable.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric
- Flexibility in shaping their magnitude response
- Very Fast Implementation (based around FFTs)

- The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed.

FIR as a class of LTI Filters

- Transfer function of the filter is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

- Finite Impulse Response (FIR) Filters: (N = 0, no feedback)

➔From H(z):

$$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-\mathrm{i}\omega} + \cdots + h_{n-1} e^{-\mathrm{i}(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega - \mathrm{i} \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

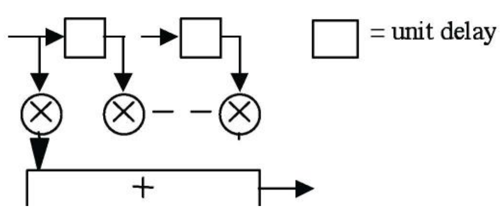
∵ H(ω) is periodic and conjugate

∴ Consider ω ∈ [0, π]

FIR Filters

- Let us consider an FIR filter of length *M*
- Order *N*=*M*-1 **(watch out!)**
- Order → number of delays

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) = \sum_{k=0}^{M-1} h(k)\, x(n-k)$$



FIR Impulse Response

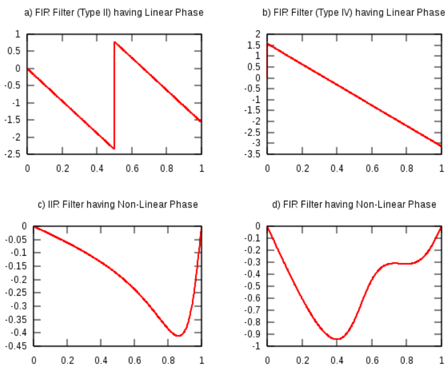
Obtain the impulse response immediately with $x(n) = \delta(n)$:

$$h(n) = y(n) = \sum_{k=0}^{M-1} b_k \delta(n-k) = b_n$$

- The impulse response is of finite length *M* (good!)
- FIR filters have only zeros (no poles) (as they must, N=0 !!)
  - Hence known also as **all-zero** filters
- FIR filters also known as **feedforward** or **non-recursive,** or **transversal** filters

FIR & Linear Phase

- The phase response of the filter is a linear function of frequency
- Linear phase has constant group delay, all frequency components have equal delay times. ∴ No distortion due to different time delays of different frequencies

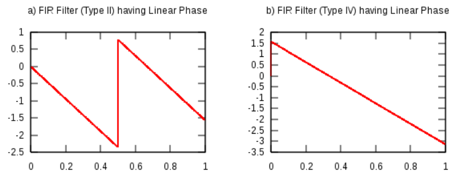


Ref: Wikipedia (Linear Phase)

- FIR Filters with:

$$\sum_{n=-\infty}^{\infty} h[n] \cdot \sin(\omega \cdot (n-\alpha) + \beta) = 0$$

FIR & Linear Phase → Four Types



Ref: Wikipedia (Linear Phase)

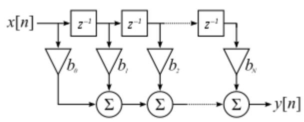
Impulse response	# coefs	$H(\omega)$	Type
$h(n) = h(M-1-n)$	Odd	$e^{-j\omega(M-1)/2}\left(h\left(\frac{M-1}{2}\right) + 2\sum_{k=1}^{(M-3)/2} h\left(\frac{M-1}{2}-k\right)\cos(\omega k)\right)$	1
$h(n) = h(M-1-n)$	Even	$e^{-j\omega(M-1)/2} 2\sum_{k=1}^{(M-3)/2} h\left(\frac{M}{2}-k\right)\cos\left(\omega\left(k-\frac{1}{2}\right)\right)$	2
$h(n) = -h(M-1-n)$	Odd	$e^{-j[\omega(M-1)/2-\pi/2]}\left(2\sum_{k=1}^{(M-1)/2} h\left(\frac{M-1}{2}-k\right)\sin(\omega k)\right)$	3
$h(n) = -h(M-1-n)$	Even	$e^{-j[\omega(M-1)/2-\pi/2]} 2\sum_{k=1}^{(M-1)/2} h\left(\frac{M}{2}-k\right)\sin\left(\omega\left(k-\frac{1}{2}\right)\right)$	4

- Type 1: most versatile
- Type 2: frequency response is always 0 at $\omega=\pi$
  (not suitable as a high-pass)
- Type 3 and 4: introduce a $\pi/2$ phase shift, 0 at $\omega=0$
  (not suitable as a high-pass)

** FIR Filter Design **

- How to get all these coefficients?

$$H(\omega) = h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega}$$

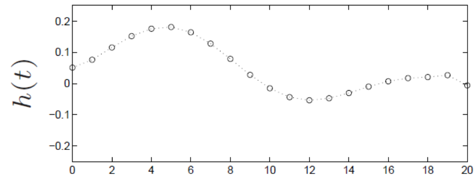


FIR Design Methods:

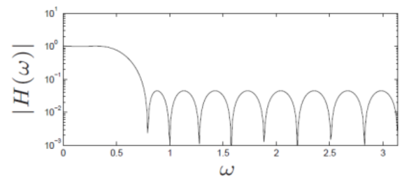
1. Impulse Response Truncation
   \+ Simplest
   – Undesirable frequency domain-characteristics, not very useful
2. Windowing Design Method
   \+ Simple
   – Not optimal (not minimum order for a given performance level)
3. Optimal filter design methods
   \+ "More optimal"
   – Less simple…

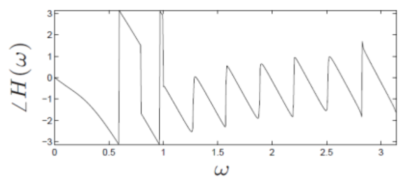
FIR Filter Design & Operation
Ex: Lowpass FIR filter

- Set Impulse response (order n = 21)
- "Determine" h(t)
  - h(t) is a 20 element vector that we'll use to as a weighted sum



- FFT ("Magic") gives Frequency Response & Phase





Why is this "hard"? Looking at the Low-Pass Example

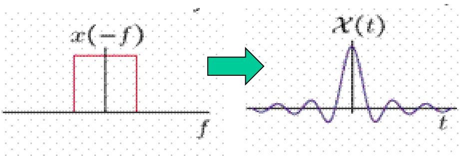
$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \le \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

- Why is this hard?
  - Shouldn't it be "easy" ??
    … just hit it with some FFT "magic" and then keep the bands we want and then hit it with some Inverse-FFT "supermagic"???
  - Remember we need a "system" that does this "rectangle function" in frequency
  - Let's consider what that means…
    - It basically suggests we need an **Inverse FFT** of a **"rectangle function"**

Flashback: Fourier Series & Rectangular Functions

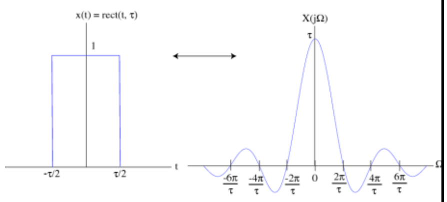
$\mathfrak{F}$: Fourier Tranform

$$\mathfrak{F}^{-1}\left\{rect\left(\frac{\omega}{2}\right)\right\} = \frac{sinc\,(t)}{\pi}$$



Ref: http://cnx.org/content/m26719/1.1/
http://www.wolframalpha.com/input/?i=IFFT%28sinc%28f%29%29

$$\mathfrak{F}\{rect\,(t)\} = sinc\left(\frac{\omega}{2}\right)$$



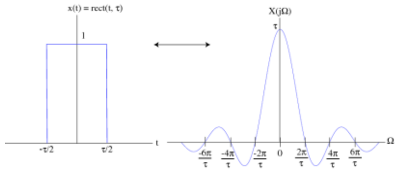
Ref: http://cnx.org/content/m32899/1.8/
http://www.thefouriertransform.com/pairs/box.php

See:

- Table 7.1 (p. 702) Entry 17 & Table 9.1 (p. 852) Entry 7

Flashback: Fourier Series & Rectangular Functions [2]

- The sinc function might look familiar
  – This is the frequency content of a square wave (box)



Ref: http://www.wolframalpha.com/input/?i=FFT%28rect%28t%29%29
http://cnx.org/content/m32899/1.8/

- This also applies to **signal reconstruction!**

➔ Whittaker–Shannon interpolation formula
  – This says that the "better way" to go from Discrete to Continuous (i.e. D to A) is not ZOH, but rather via the sinc!

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \mathrm{sinc}\left(\frac{t-nT}{T}\right)$$

∴ FIR and Low Pass Filters…

∴

$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \le \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

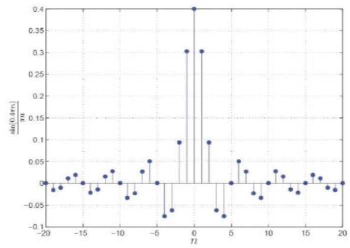
Has impulse response:

$$h_d(n) = \frac{\omega_c}{\pi} \frac{\sin \omega_c n}{\omega_c n}$$

Thus, to filter an impulse train with an ideal low-pass filter use:

$$x(t) = \left(\sum_{n=-\infty}^{\infty} x[n] \cdot \delta(t - nT)\right) * \text{sinc}\left(\frac{t}{T}\right)$$

- **However**!!
  a sinc is non-causal and infinite in duration



And, this **cannot** be implemented **in practice** ☹

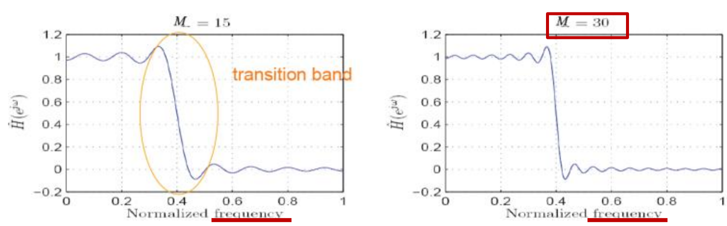
∵ we need to know all samples of the input, both in the **past** and in the **future**

Plan 0: Impulse Response Truncation

Maybe we saw this coming…

∴ Clip off the sinc at some large *n*

$$\hat{h}(n) = \frac{\sin(n\omega_c)}{\pi n} \text{ for } |n| \le M \text{ and } 0 \text{ otherwise}$$



- *Ripples* in both passband/stopband
  and the transition not abrupt (i.e., a *transition band*).
- As M→∞, transition band→ 0 (as expected!)

➔ FIR Filters: Window Function Design Method

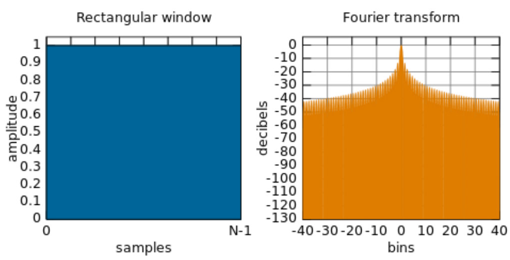
- Windowing: a generalization of the truncation idea

- There many, many "window" functions:
  - Rectangular
  - Triangular
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Lanczos
  - Many More … (see: http://en.wikipedia.org/wiki/Window_function)

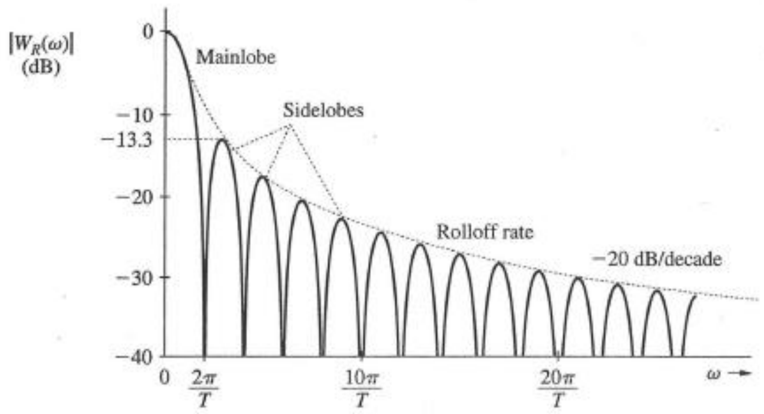
Some Window Functions [1]

1. Rectangular

$$w(n) = 1$$



Windowing and its effects/terminology



Lathi, Fig. 7.45

Next Time…

- **Digital Filters**

- Review:
  - Chapter 12 of Lathi

- Ponder?

$$y[k] = f[k] * h[k] \qquad Y(\Omega) = F(\Omega)H(\Omega)$$

where $F(\Omega)$, $Y(\Omega)$, and $H(\Omega)$ are DTFTs of $f[k]$, $y[k]$, and $h[k]$, respectively; that is,

$$f[k] \Longleftrightarrow F(\Omega), \quad y[k] \Longleftrightarrow Y(\Omega), \quad \text{and} \quad h[k] \Longleftrightarrow H(\Omega)$$