پردازش سیگنال دیجیتال

درس ۱۱

تبدیل A/D و D/A

A/D and D/A Conversion

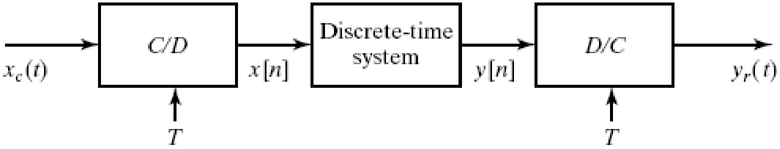
کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

http://courses.fouladi.ir/dsp

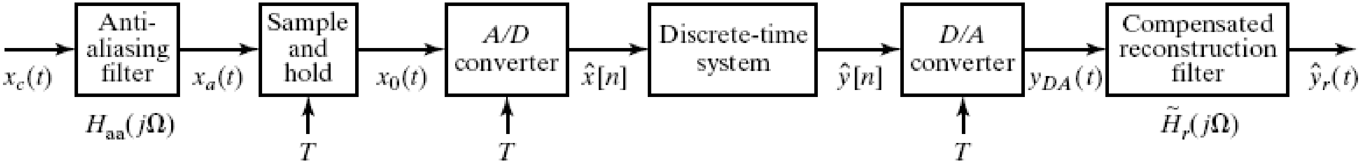
A/D and D/A Conversion

Ideal Conversion

- Up to this point we assumed **ideal** D/C and C/D conversion



- In practice, however
  - Continuous-time signals are not perfectly bandlimited
  - D/C and C/D converters can only be approximated with D/A and A/D converters
- A more realistic model for digital signal processing:



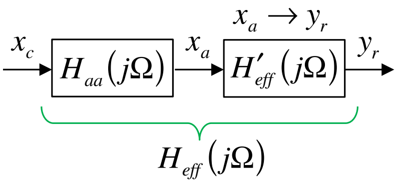
Prefiltering to Avoid Aliasing

- Desirable to **minimize sampling rate**
  - Minimizes amount of data to process
- No point of sampling high frequencies that are not of interest
  - Frequencies we don’t expect any signal in only contribute as noise
- A **low-pass anti-aliasing filter** would improve both aspects
- An ideal anti-aliasing filter

$$H_{aa}(j\Omega) = \begin{cases} 1 & |\Omega| < \Omega_c < \pi / T \\ 0 & |\Omega| > \Omega_c \end{cases}$$

- In this case the effective response is

$$H_{eff}(j\Omega) = \begin{cases} H\left(e^{j\Omega T}\right) & |\Omega| < \Omega_c \\ 0 & |\Omega| > \Omega_c \end{cases}$$



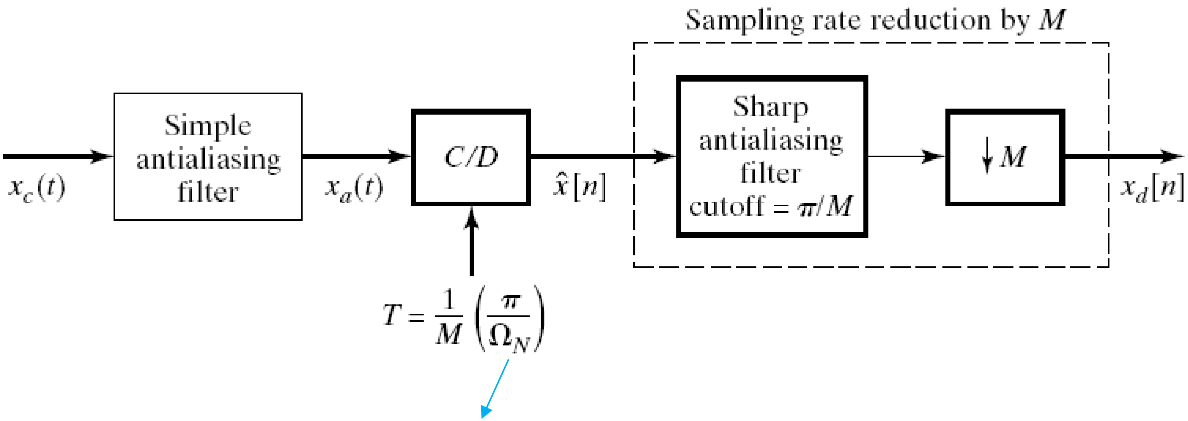
- In practice an ideal low-pass filter is not possible hence

$$H_{eff}(j\Omega) \approx H_{aa}(j\Omega) H\left(e^{j\Omega T}\right)$$

- This would require sharp-cutoff analog filters which are expansive.

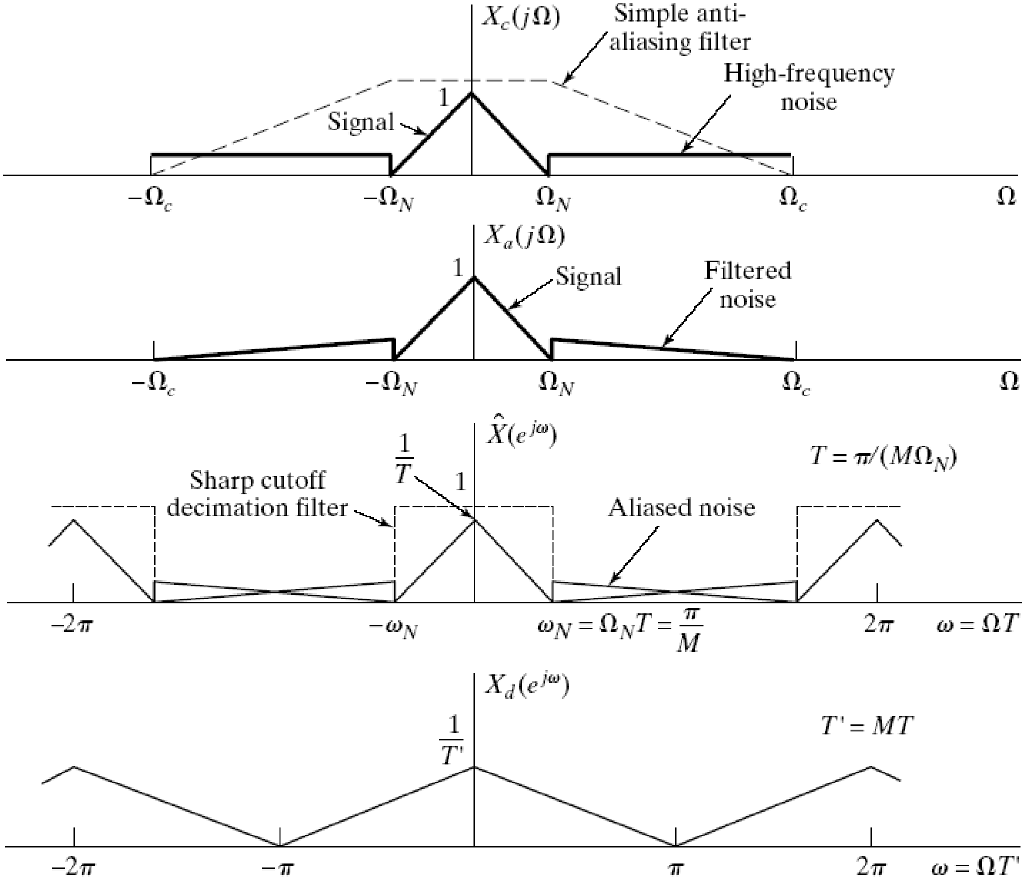
Oversampled A/D Conversion

- The idea is
  - To have a simple (non-sharp) analog anti-aliasing filter
  - Use higher than required sampling rate
  - Implement sharp anti-aliasing filter in **discrete-time**
  - Downsample to desired sampling rate


Sampling rate reduction by M
$x_c(t)$
Simple antialiasing filter
$x_a(t)$
C/D
$\hat{x}[n]$
Sharp antialiasing filter cutoff = π/M
$\downarrow M$
$x_d[n]$
$T = \frac{1}{M}\left(\frac{\pi}{\Omega_N}\right)$

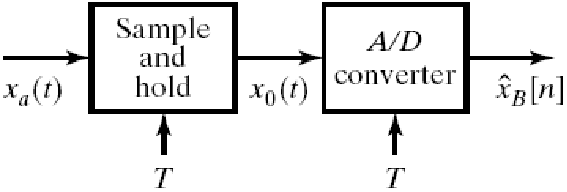
largest frequency component after antialising

Oversampled A/D Conversion: Example



Analog-to-Digital (A/D) Conversion

- Ideal C/D converters convert continuous-time signals into infinite-precision discrete-time signals
- **In practice** we implement C/D converters as the cascade of



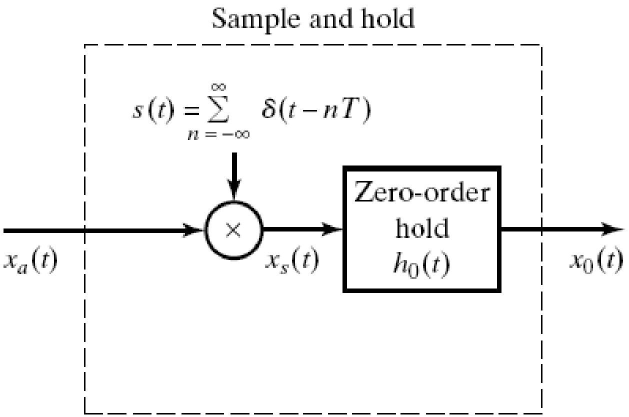
- The sample-and-hold device holds current/voltage constant
- The A/D converter converts current/voltage into finite-precisions number
- The ideal sample-and-hold device has the output

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT)$$

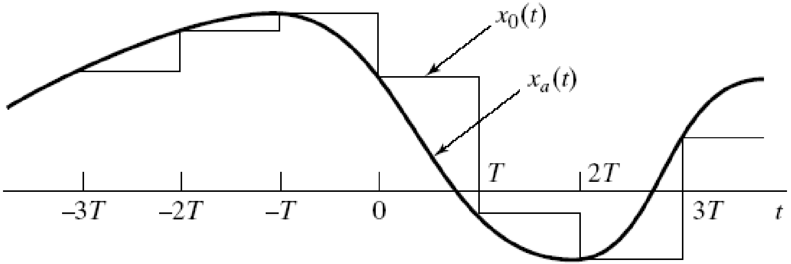
$$h_0(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{otherwise} \end{cases}$$

Sample and Hold

- An ideal sample-and-hold system

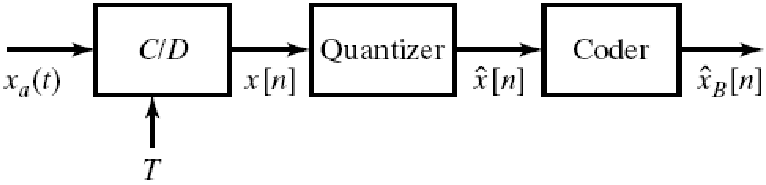


- Time-domain representation of sample-and-hold operation



A/D Converter Model

- A practical A/D converter can be modeled as:



- The C/D converter represent the sample-and-hold operation
- **Quantizer** transforms input into a **finite** set of numbers

$$\hat{x}[n] = Q(x[n])$$

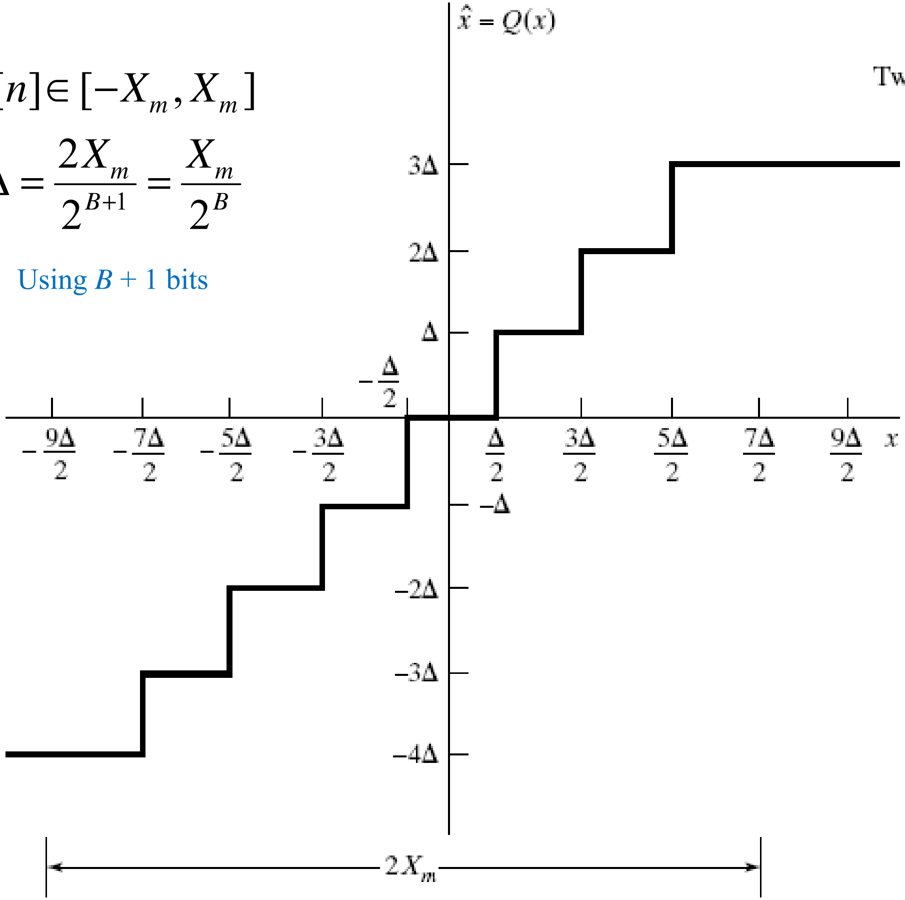
- Most of the time, uniform quantizers are used.

Uniform Quantizer

$$x[n] \in [-X_m, X_m]$$

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$

Using $B + 1$ bits



Two's-complement code	Offset binary code
011	111
010	110
001	101
000	100
111	011
110	010
101	001
100	000

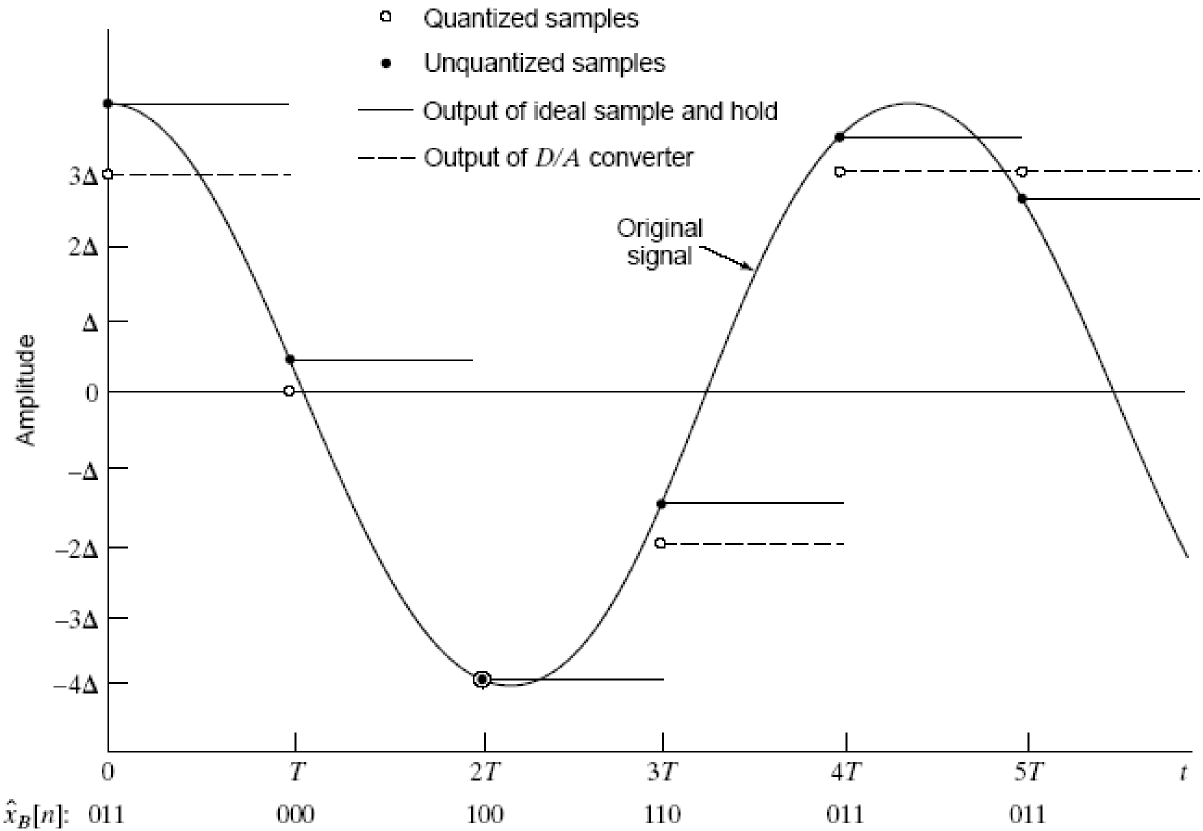
Two’s Complement Numbers

- Representation for signed numbers in computers
- **Integer** two’s-complement $-a_0 2^B + a_1 2^{B-1} + \ldots + a_B 2^0$
- **Fractional** two’s-complement $-a_0 2^0 + a_1 2^{-1} + \ldots + a_B 2^{-B}$
- Example $B + 1$ = 3 bit two’s-complement numbers

$-a_0 2^2 + a_1 2^1 + a_2 2^0$	
Binary Symbol	**Numerical Value**
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

$-a_0 2^0 + a_1 2^{-1} + a_2 2^{-2}$	
Binary Symbol	**Numerical Value**
0.11	3/4
0.10	2/4
0.01	1/4
0.00	0
1.11	-1/4
1.10	-2/4
1.01	-3/4
1.00	-4/4

Example



○ Quantized samples
● Unquantized samples
Output of ideal sample and hold
Output of D/A converter
Original signal
Amplitude
3Δ
2Δ
Δ
0
$-\Delta$
-2Δ
-3Δ
-4Δ
0
T
$2T$
$3T$
$4T$
$5T$
t
$\hat{x}_B[n]$: 011
000
100
110
011
011

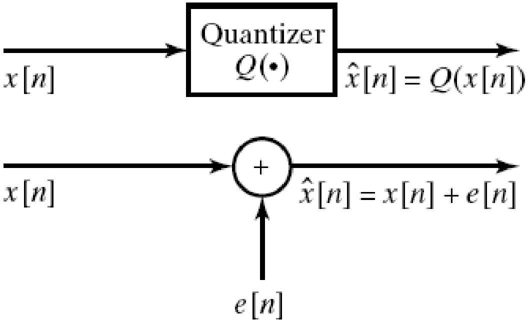
Quantization Error

- Quantization error: $e[n] = \hat{x}[n] - x[n]$
  - difference between the original and quantized value
- If quantization step is Δ the quantization error will satisfy

$$-\Delta/2 < e[n] < \Delta/2$$

  - As long the input does not clip
- Based on this fact we may use the following simplified model
- In most cases we can assume that
  - $e[n]$ is uniformly distributed
  - Is uncorrelated with the signal $x[n]$
- The variance of $e[n]$ is then

Variance of uniform distribution: $\sigma_e^2 = \dfrac{\Delta^2}{12}$



- And the **signal-to-noise ratio** of quantization noise for $B + 1$ bits

$$SNR = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right)$$

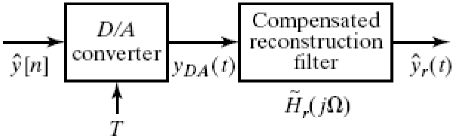
D/C Conversion

- Perfect reconstruction requires filtering with ideal LPF

$$Y_r(j\Omega) = Y\left(e^{j\Omega T}\right)H_r(j\Omega)$$

$$Y\left(e^{j\Omega T}\right): \quad \text{DTFT of sampled signal}$$

$$Y_r(j\Omega): \quad \text{FT of reconstructed signal}$$



- The ideal reconstruction filter

$$H_r(j\Omega) = \begin{cases} T & |\Omega| < \pi / T \\ 0 & |\Omega| > \pi / T \end{cases}$$

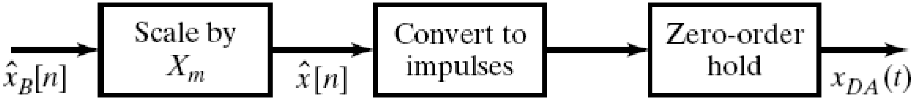
- The time domain reconstructed signal is

$$y_r(t) = \sum_{n=-\infty}^{\infty} y[n] \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}$$

- In practice we cannot implement an **ideal** reconstruction filter

D/A Conversion

- The practical way of D/C conversion is a D/A converter



- It takes a binary code and converts it into continuous-time output $x_c(t)$

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} X_m \hat{x}_B[n] h_0(t-nT) = \sum_{n=-\infty}^{\infty} \hat{x}[n] h_0(t-nT)$$

- Using the additive noise model for quantization $\hat{x}[n] = x[n] + e[n]$

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t-nT) + \sum_{n=-\infty}^{\infty} e[n] h_0(t-nT) = x_0(t) + e_0(t)$$

- The signal component in frequency domain can be written as

$$X_0(j\Omega) = X\left(e^{j\Omega T}\right) H_0(j\Omega)$$

- So to recover the desired signal component we need a compensated reconstruction filter of the form

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)}$$

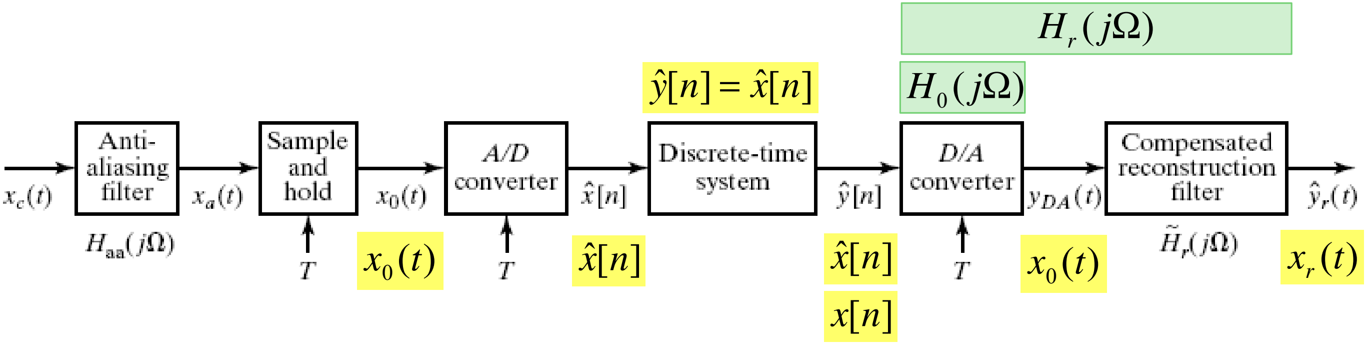
Compensated Reconstruction Filter

$$|\Omega| < \frac{\pi}{T}$$

$$X_r(j\Omega) = X\left(e^{j\Omega T}\right)H_r(j\Omega)$$ Ideal interpolating filter

$$X_0(j\Omega) = X\left(e^{j\Omega T}\right)H_0(j\Omega)$$ Zero-order hold

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)}$$



Compensated Reconstruction Filter

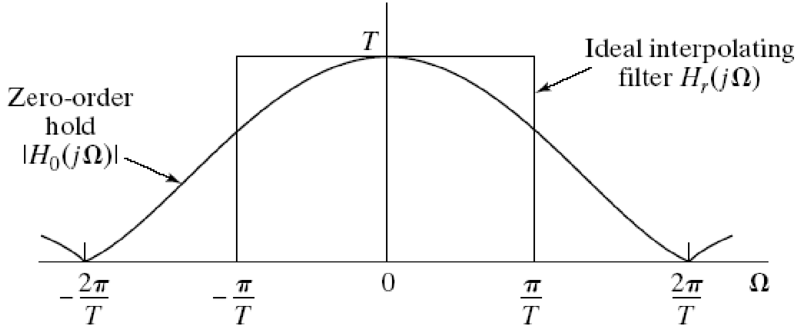
- The frequency response of zero-order hold is

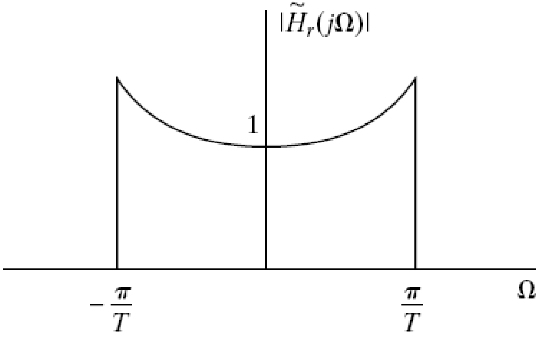
$$H_0(j\Omega) = \frac{2\sin(\Omega T/2)}{\Omega} e^{-j\Omega T/2}$$

- Therefore the compensated reconstruction filter should be

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)} \Rightarrow \quad \tilde{H}_r(j\Omega) = \begin{cases} \dfrac{\Omega T/2}{\sin(\Omega T/2)} e^{j\Omega T/2} & |\Omega| < \pi/T \\ 0 & |\Omega| > \pi/T \end{cases}$$

$$\boxed{H_r(j\Omega) = \begin{cases} T & |\Omega| < \pi/T \\ 0 & |\Omega| > \pi/T \end{cases}}$$


T
Ideal interpolating filter $H_r(j\Omega)$
Zero-order hold $|H_0(j\Omega)|$
$-\frac{2\pi}{T}$
$-\frac{\pi}{T}$
0
$\frac{\pi}{T}$
$\frac{2\pi}{T}$
Ω


$|\tilde{H}_r(j\Omega)|$
1
$-\frac{\pi}{T}$
$\frac{\pi}{T}$
Ω