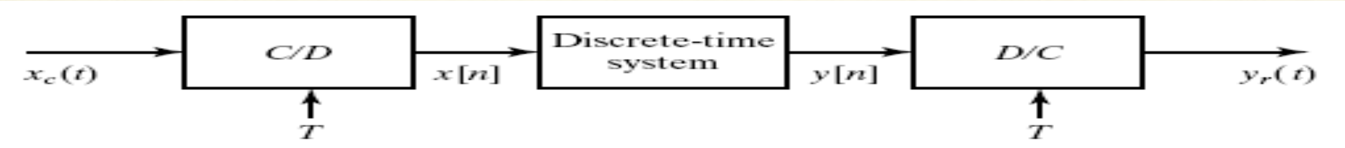
Digital Signal Processing A-to-D & D-to-A Conversion

Lecture-9
13-April-16

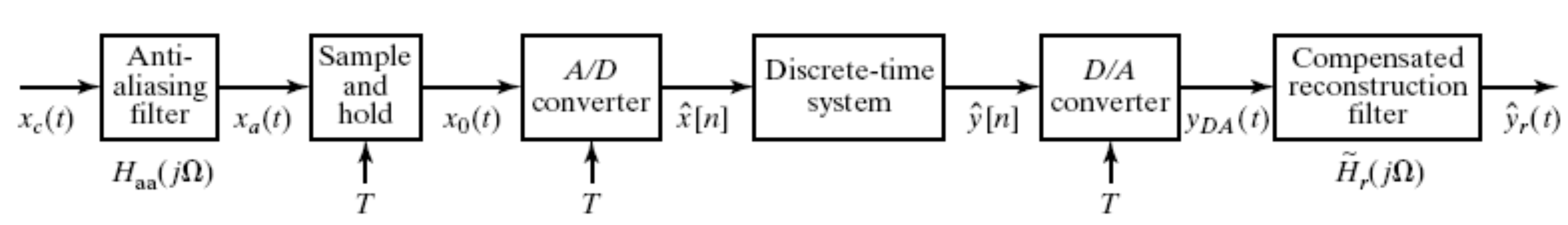
Ideal Conversion

- Until now we assumed ideal D/C and C/D conversion:



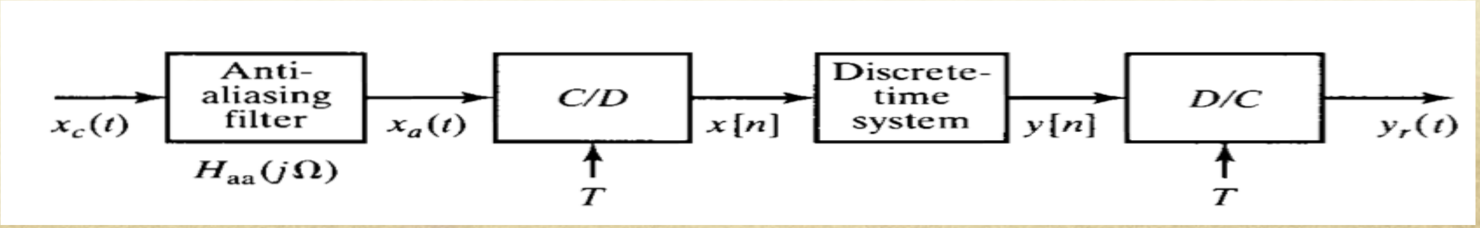
- However, in practical:
  - Continuous time signals are not precisely band limited.
  - Ideal filters cannot be realized.
  - The ideal C/D and D/C converters can only be approximated by devices.

- Shown below:



Prefiltering to Avoid Aliasing

- It is desirable to minimize the sampling rate, i.e., minimizing amount of data to process.
- If the input is not band limited or if the Nyquist frequency is too high, prefiltering may be necessary.
- Frequencies we don't expect in any signal only contribute as noise.
- To avoid aliasing the input signal must be forced to be band limited to frequencies below one-half the desired sampling rate.
- This can be accomplished by low pass filtering the continuous time signal prior to C/D conversion.
- That filters is known as antialiasing filter.
- Shown in the following figure below:



Prefiltering to Avoid Aliasing

- Frequency response of Anti-aliasing filter is:

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

- The output of the antialiasing filter $x_a(t)$ to the output $y_r(t)$ will always behave as a linear time-invariant system.
- The input to the C/D converter $x_a(t)$ is forced by the antialiasing filter to be band limited to frequencies below π/T radians/s.
- In this case the effective response is:

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

Prefiltering to Avoid Aliasing

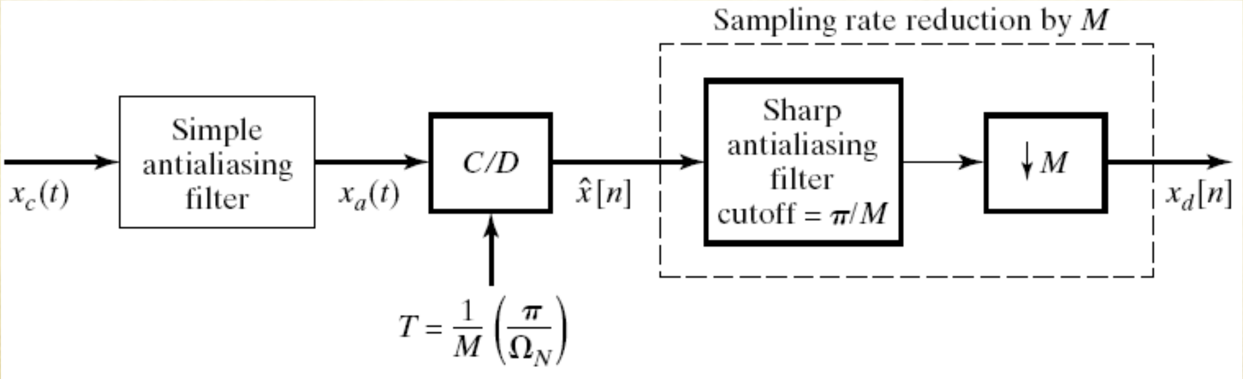
- In practice the frequency response $H_{aa}(j\Omega)$ cannot be ideally band limited but can be made smaller so that aliasing is minimized.
- In this case the overall frequency response of the system is:

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

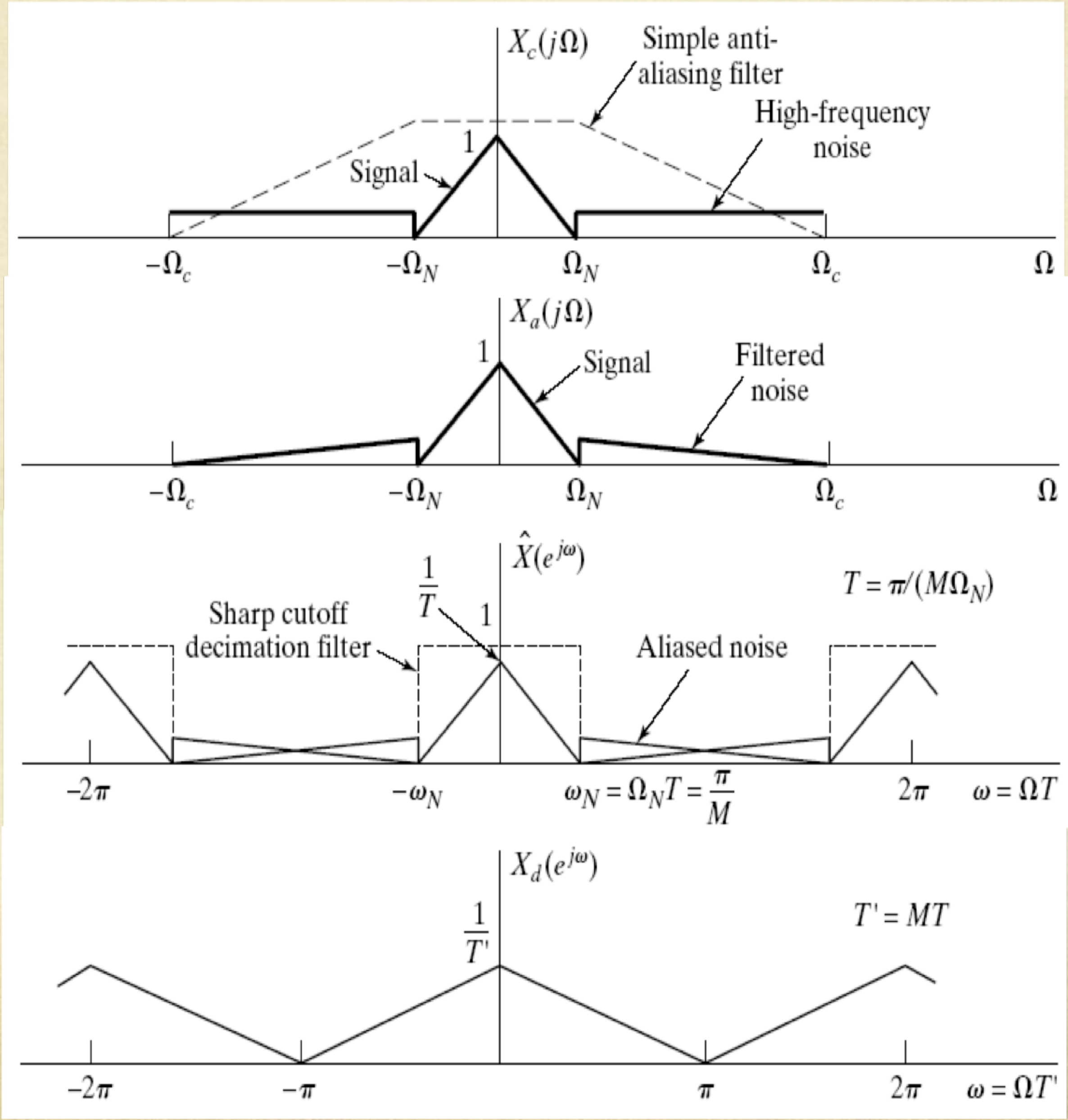
- This would require sharp-cutoff analog filters which are expensive.

Oversampled A/D Conversion

- Apply a very simple antialiasing filter that has a gradual cutoff with significant attenuation at $M\Omega_N$.
- Use higher sampling rate than required sampling rate.
- Implement sharp antialiasing filtering in the discrete-time domain.
- Down sample the desired sampling rate.



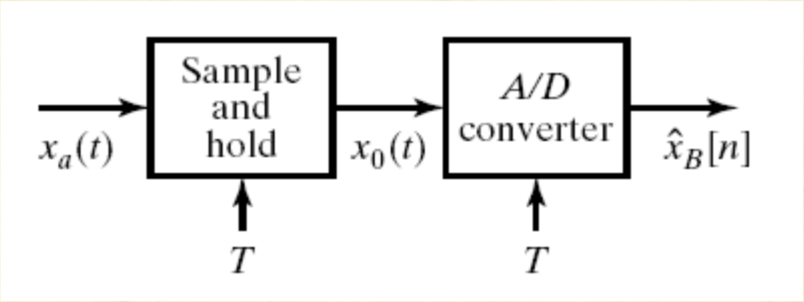
Example



$X_c(j\Omega)$
Simple anti-
aliasing filter
High-frequency
noise
1
Signal
$-\Omega_c$
$-\Omega_N$
Ω_N
Ω_c
Ω
$X_a(j\Omega)$
1
Signal
Filtered
noise
$-\Omega_c$
$-\Omega_N$
Ω_N
Ω_c
Ω
$\hat{X}(e^{j\omega})$
$\frac{1}{T}$
1
$T = \pi/(M\Omega_N)$
Sharp cutoff
decimation filter
Aliased noise
-2π
$-\omega_N$
$\omega_N = \Omega_N T = \frac{\pi}{M}$
2π
$\omega = \Omega T$
$X_d(e^{j\omega})$
$\frac{1}{T'}$
$T' = MT$
-2π
$-\pi$
π
2π
$\omega = \Omega T'$

Analog-to-Digital Conversion

- An ideal C/D converter converts a continuous-time signal into a discrete-time signal, where each sample is known with infinite precision.
- In practice C/D converters are implemented as the cascade of:



- The sample and hold device holds current/voltage constant.
- The A/D converter converts current/voltage into finite precisions number.

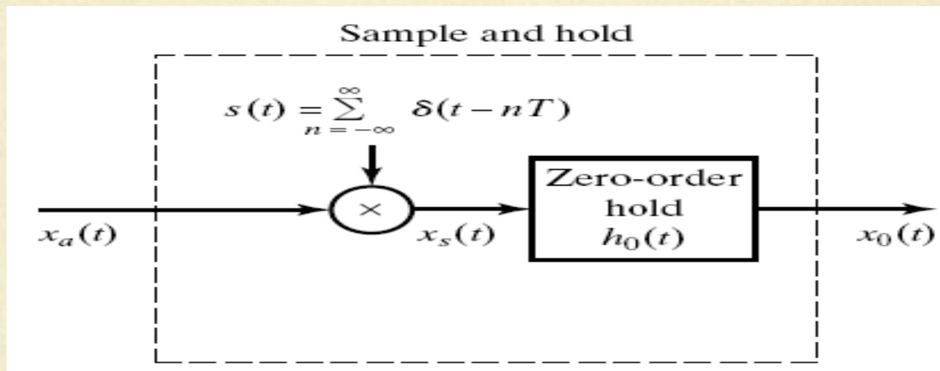
Analog-to-Digital Conversion (cont.)

- The ideal sample and hold device has the output:

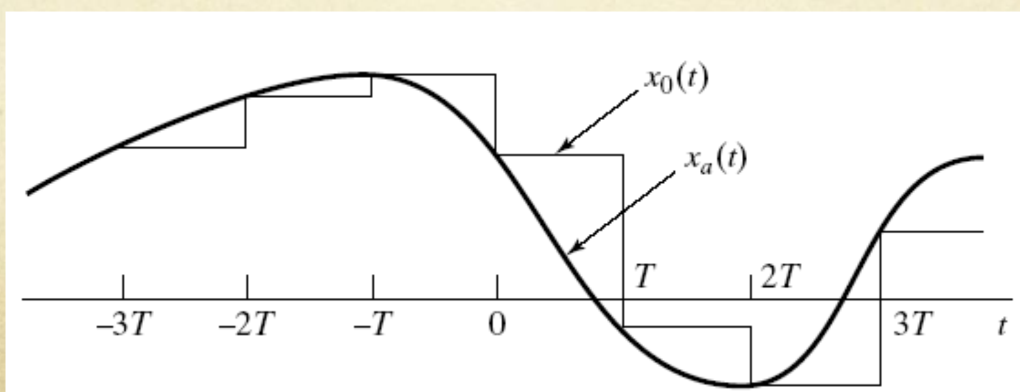
$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT) \qquad h_0(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{else} \end{cases}$$

Sample and Hold

- An ideal sample and hold is equivalent o impulse train modulation followed by linear filtering with zero-order-hold system. As shown below:

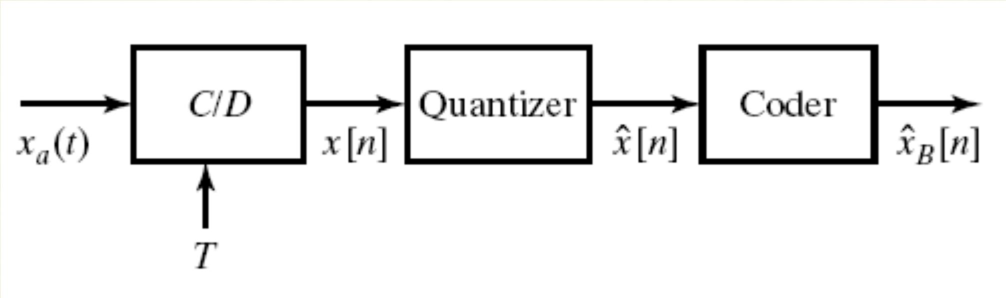


- Time-domain representation of sample and hold operation is:



A/D Converter Model

- A/D converter can be modeled as:

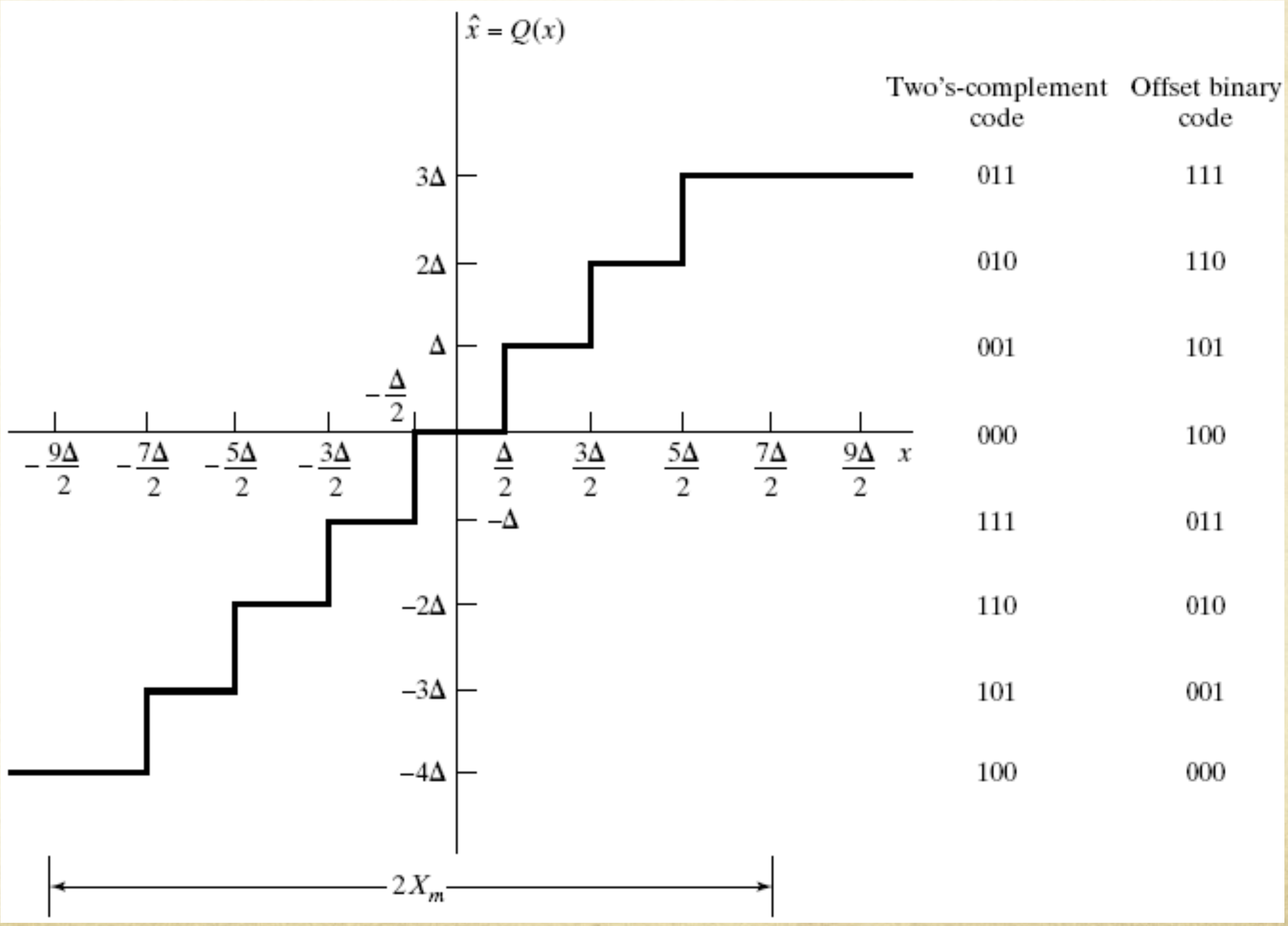


- The C/D converter represent the sample-hold-operation.
- Quantizer transforms input into a finite set of numbers.

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

- Mostly uniform quantizers are used.

Uniform Quantizer



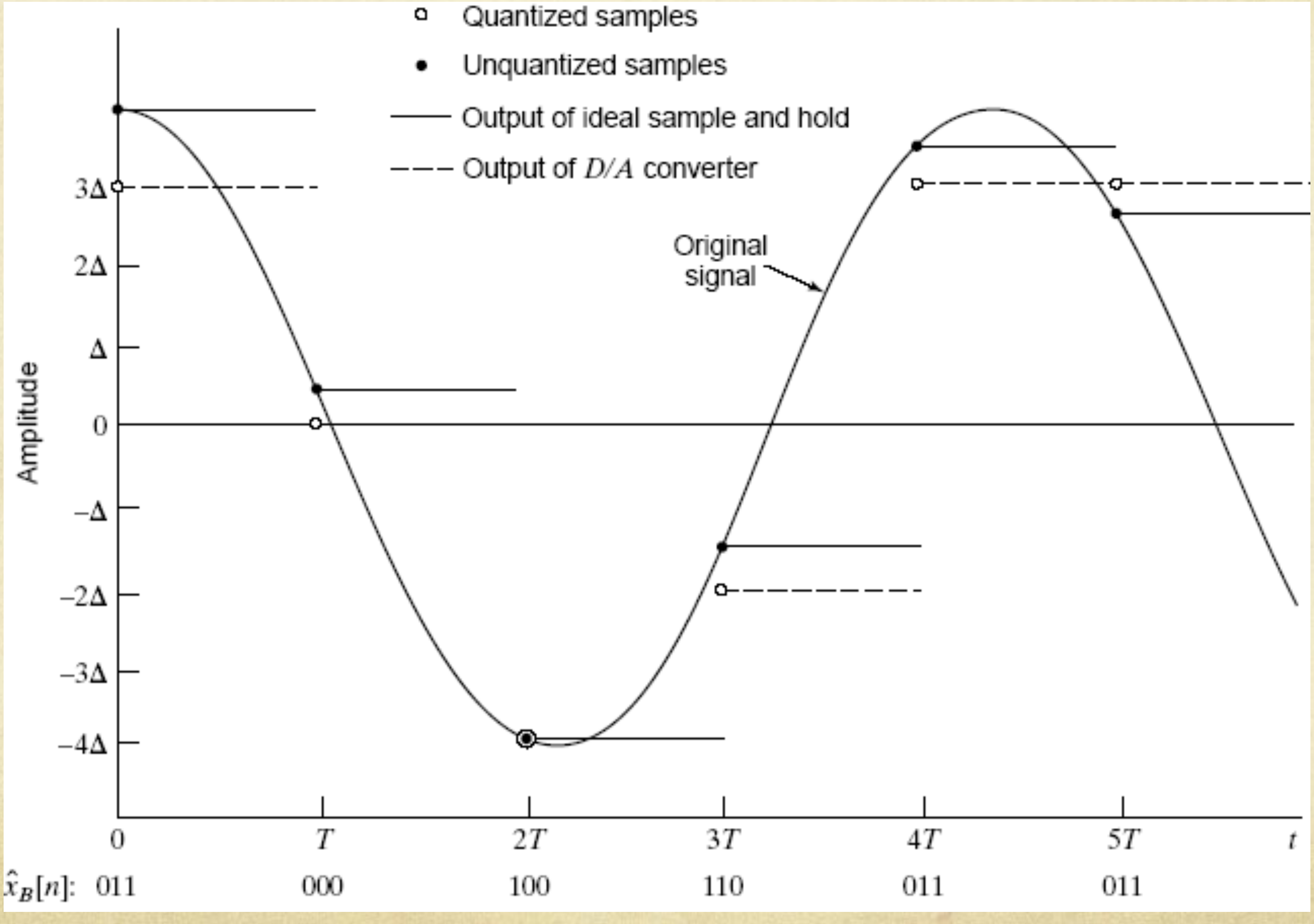
2's Complement Numbers

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



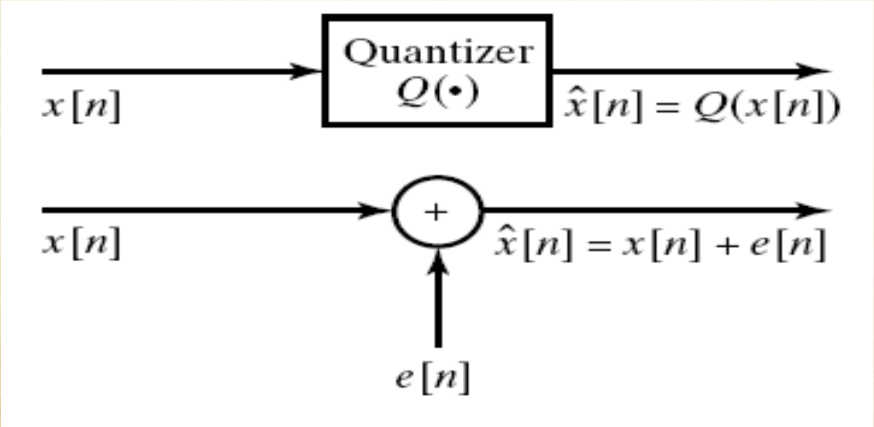
Quantized samples
Unquantized samples
Output of ideal sample and hold
Output of D/A converter
Original signal
Amplitude
3Δ
2Δ
Δ
0
−Δ
−2Δ
−3Δ
−4Δ
0
T
2T
3T
4T
5T
t
$\hat{x}_B[n]$: 011
000
100
110
011
011

Quantization Error

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

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

- A simplified but useful model of the quantizer is depicted below:



Quantization Error (cont.)

- The statistical representation of quantization error is based on the following assumptions:
  - The error sequence e[n] is a sample sequence of a stationery random process.
  - The error sequence is uncorrelated with the sequence x[n].
  - The random variables of the error process are uncorrelated i.e., the error is a white noise process.
  - The probability distribution of the error process is uniform over the range of quantization error.

D/C Conversion

- Perfect reconstruction requires filtering with ideal LPF

$$X_r(j\Omega) = X(e^{j\Omega T})H_r(j\Omega)$$

$$X(e^{j\Omega T}): \text{DTFT of sampled signal}$$

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

- The ideal reconstruction filter

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

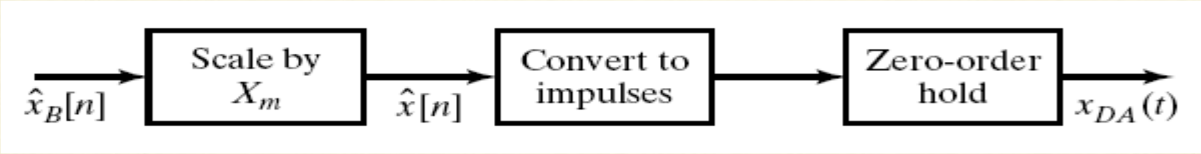
- The time domain reconstructed signal is

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

- In practice we cannot implement an ideal reconstruction filter

D/A Conversion

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



- It takes a binary code and converts it into continuous-time output

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

- Using the additive noise model for quantization

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

- The signal component in frequency domain can be written as

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

D/A Conversion (cont.)

- 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

- 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

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

Compensated Reconstruction Filter (cont.)