ELCE 705
DIGITAL SIGNAL PROCESSING

Sampling of
continuous-time signals
(4.6- 4.8 exclude 4.7.6)

Contents

- Change the sampling Rate
- Multirate signal processing
- Digital Processing of Analog Signals

Changing the Sampling Rate

3

- A continuous-time signal can be represented by its samples as

$$x[n] = x_c(nT)$$

- Some applications require us to change the sampling rate
  - To obtain a new discrete-time signal from the same CT signal

$$x'[n] = x_c(nT') \qquad \text{where } T \neq T'$$

- One way of accomplishing this is to
  - Reconstruct the continuous-time signal from x[n]
  - Resample the continuous-time signal using new rate to get x'[n]
  - This requires analog processing which is often undesired
- Target: get x'[n] directly from x[n]

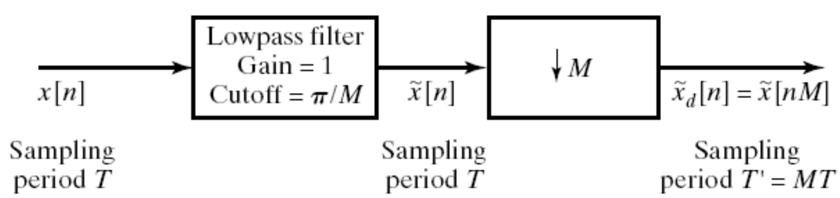
Downsampling (Compressor)
- Sampling Rate Reduction by an Integer Factor

4

- Reduce the sampling rate of a sequence by "sampling" it

$$x_d[n] = x[nM] = x_c(nMT) \qquad \boxed{T_d = MT}$$

- This is accomplished with a **sampling rate compressor**



- $x_d[n]$ is identical to what being got by reconstructing the signal and resampling it with T'=MT
- There will be no aliasing if $\dfrac{\pi}{T'} = \dfrac{\pi}{MT} > \Omega_N$

Frequency Domain Representation

- Recall the DTFT of $x[n]=x_c(nT)$

$$X\left(e^{j\omega}\right)=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{T}-\frac{2\pi k}{T}\right)\right)$$

- The DTFT of the downsampled signal can similarly written as

$$X_d\left(e^{j\omega}\right)=\frac{1}{T'}\sum_{r=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{T'}-\frac{2\pi r}{T'}\right)\right)=\frac{1}{MT}\sum_{r=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{MT}-\frac{2\pi r}{MT}\right)\right)$$

- Let's represent the summation index as

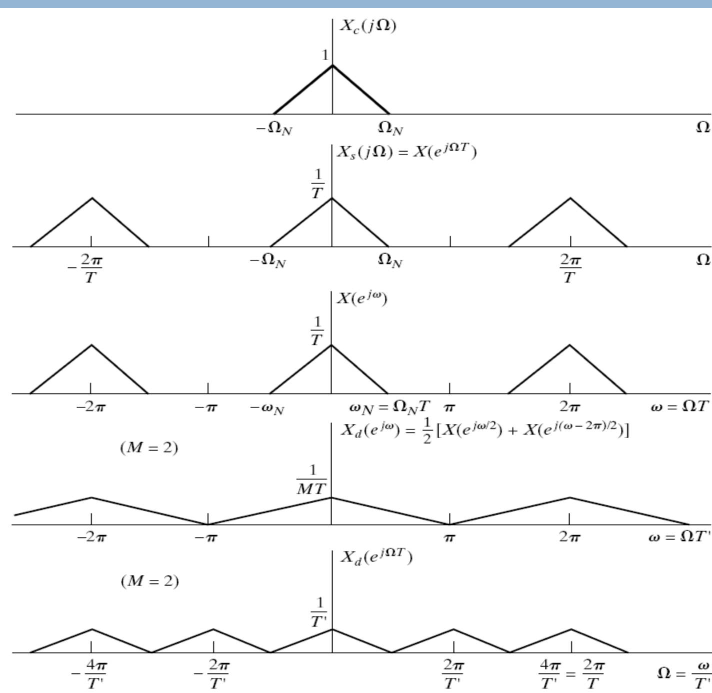
$$r=i+kM \quad \text{where} \quad -\infty<k<\infty \quad \text{and} \quad 0\le i<M$$

$$X_d\left(e^{j\omega}\right)=\frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_{k=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{MT}-\frac{2\pi k}{T}-\frac{2\pi i}{MT}\right)\right)\right]$$

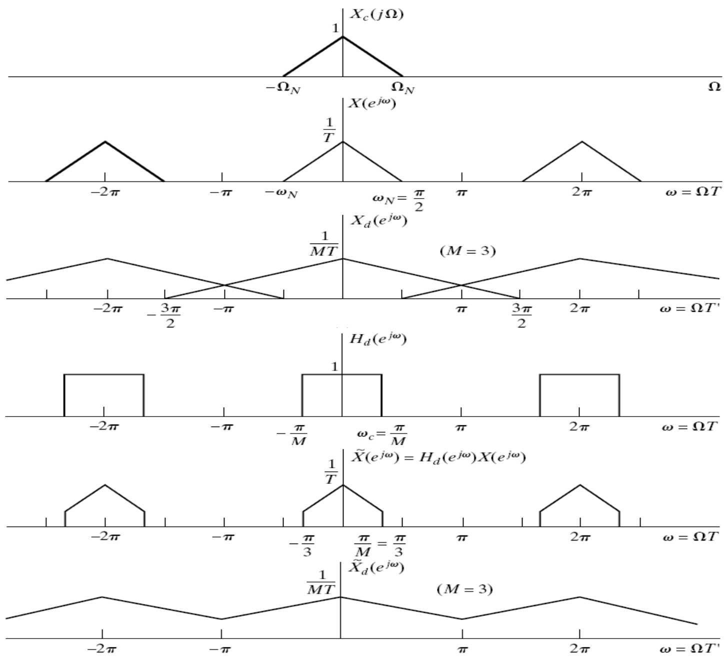
- And finally

$$X_d\left(e^{j\omega}\right)=\frac{1}{M}\sum_{i=0}^{M-1}X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi i}{M}\right)}\right)$$

Frequency Domain Representation of Downsampling: No Aliasing



Frequency Domain Representation



$$\boxed{\omega_N M \le \pi}$$

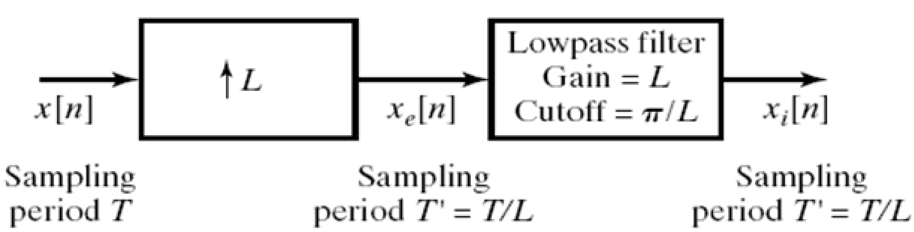
Upsampling

-Increasing the Sampling Rate by an Integer Factor

- Increase the sampling rate of a sequence interpolating it

$$x_i[n] = x[n/L] = x_c(nT/L) \qquad \boxed{T_i = T/L}$$

- This is accomplished with a sampling rate expander



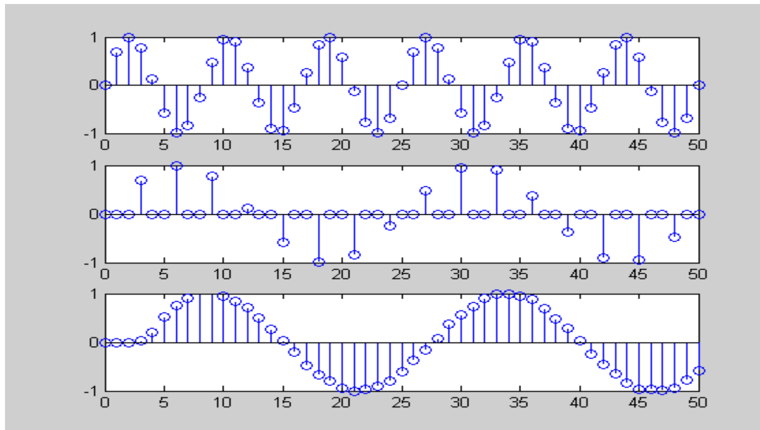
- $x_i[n]$ is identical to sample with $T_i = T/L$

Upsampling

- Upsampling consists of two steps
  - **Expander**

$$x_e[n] = \begin{cases} x[n/L] & n = 0, \mp L, \mp 2L, \ldots \\ 0 & \text{else} \end{cases} = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$$

  - **Interpolating** (Low pass filter)

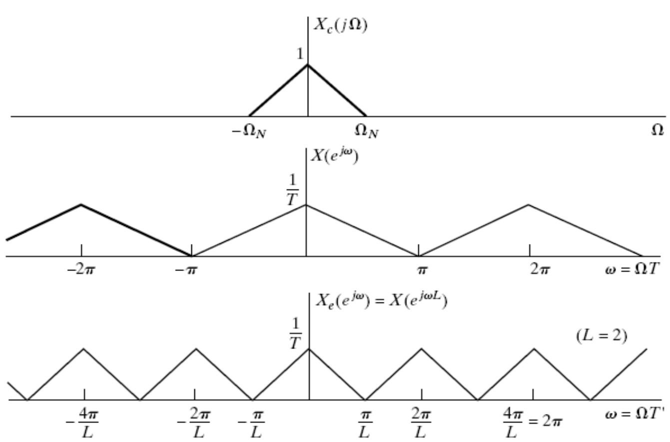


Frequency Domain Representation of Expander

- The DTFT of $x_e[n]$ can be written as

$$X_e\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right)e^{-j\omega n} = \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk} = X\left(e^{j\omega L}\right)$$

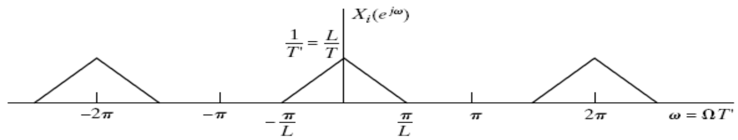
- The output of the expander is frequency-scaled



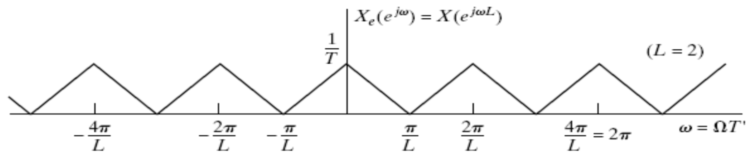
$$\boxed{\omega = \Omega T_i}$$

Frequency Domain Representation of Interpolator

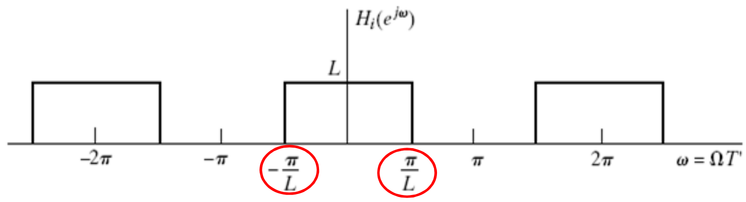
- The DTFT of the desired interpolated signals is



- The extrapolator output is given as



- Apply the LPF



Interpolator in Time Domain

- Get an interpolation formula for $x_i[n]$ in terms of $x[n]$
- The low-pass filter impulse response is

$$h_i[n] = \frac{\sin(\pi n / L)}{\pi n / L}$$

- Hence the interpolated signal is written as

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin(\pi(n - kL)/L)}{\pi(n - kL)/L}$$

- Note that

$$h_i[0] = 1$$
$$h_i[n] = 0 \qquad n = \mp L, \mp 2L, \ldots$$

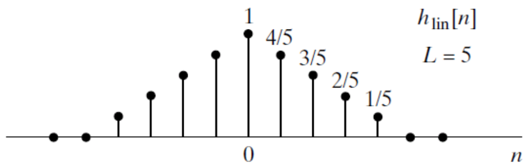
- Therefore the filter output can be written as

$$x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT') \qquad \text{for } n = 0, \mp L, \mp 2L, \ldots$$

Linear Interpolation

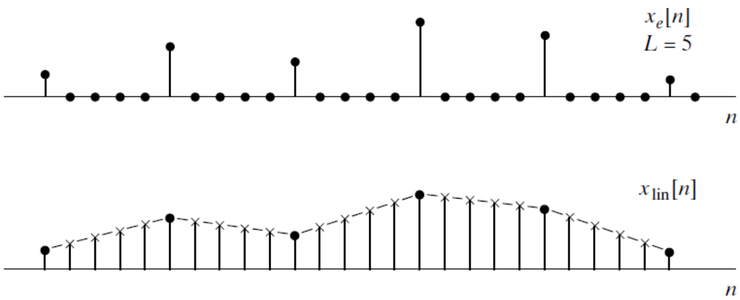
- The impulse response of the filter is

$$h_{lin}[n] = \begin{cases} 1 - |n|/L & |n| \le L \\ 0 & otherwise \end{cases}$$



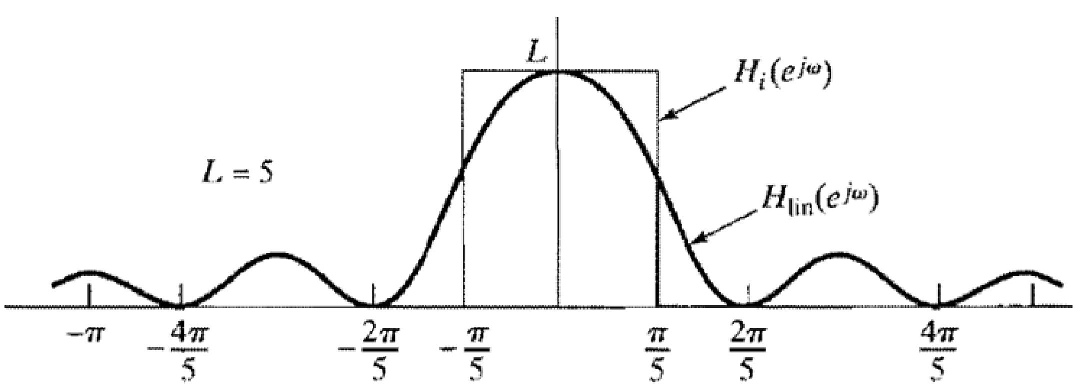
- The interpolated output

$$x_i[n] = \sum_{k=-\infty}^{\infty} x_e[k] h_{lin}[n-k] = \sum_{k=n-L+1}^{k=n+L-1} x_e[k] h_{lin}[n-k]$$



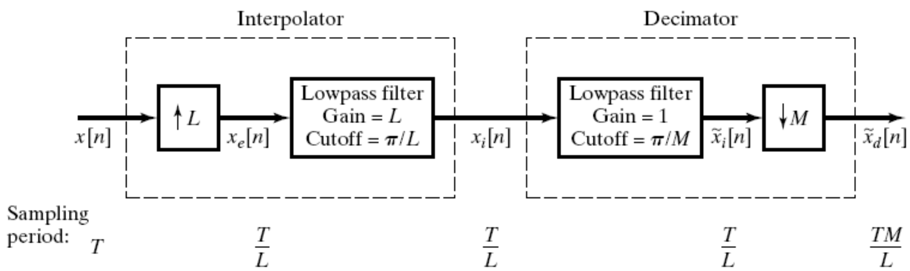
Compare the frequency response

$$H_{lin}(e^{j\omega}) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2$$

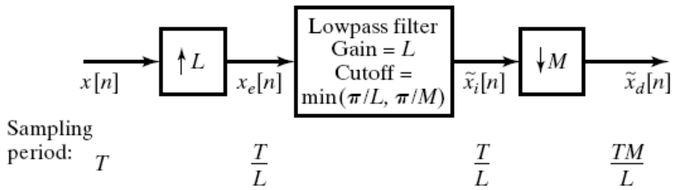


Changing the Sampling Rate by Non-Integer Factor

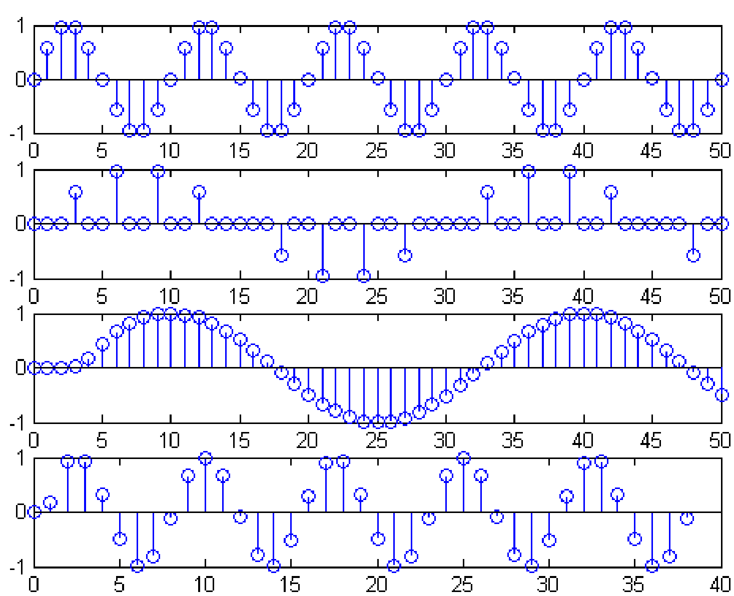
- Combine decimation and interpolation for non-integer factors



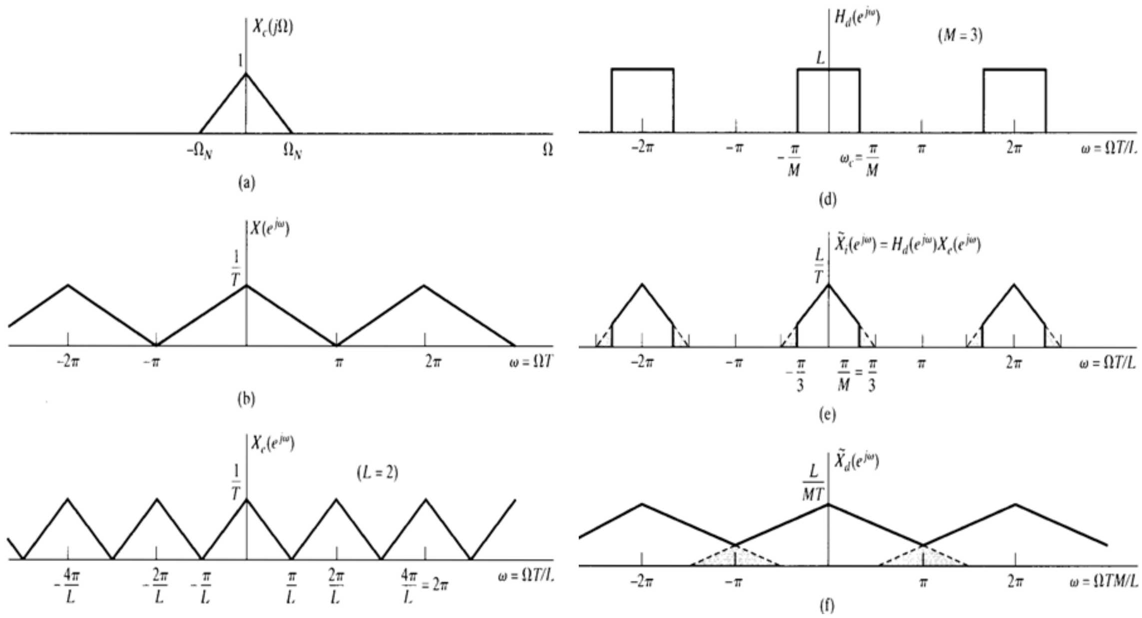
- **The two low-pass filters can be combined into a single one**



Changing the sampling rate by 4/3



Example



Contents

- Changing the sampling Rate
- Multirate signal processing
- Digital Processing of Analog Signals

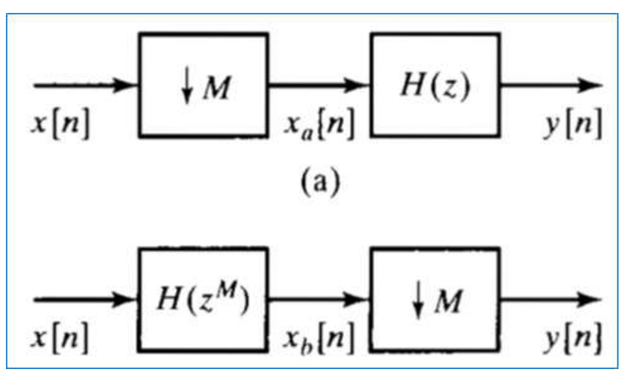
Multirate Signal Processing

- Change sampling rate by
  - Interpolation
  - Decimation
  - Combine them
- Multirate signal processing
  - Reduce the amount of computation required by sampling rate conversion
  - Two basic methods are discussed

Interchange of filtering with compressor

$$Y(e^{j\omega}) = H(e^{j\omega})\frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)})$$

$$= H(e^{j\omega})X_a(e^{j\omega}).$$

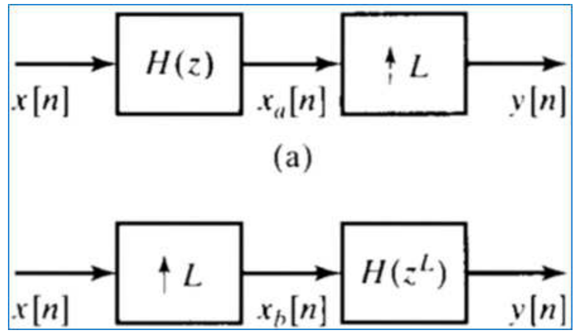


$$X_b(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega})$$

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)})H(e^{j(\omega - 2\pi i)})$$

Interchange of filtering with expander

$$\begin{aligned} Y(e^{j\omega}) &= X_a(e^{j\omega L}) \\ &= X(e^{j\omega L})H(e^{j\omega L}) \end{aligned}$$

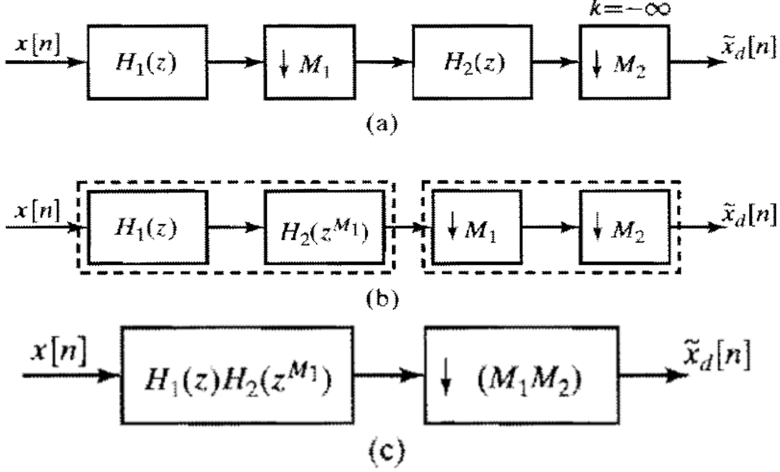


$$Y(e^{j\omega}) = H(e^{j\omega L})X_b(e^{j\omega})$$

Multistage Interpolation

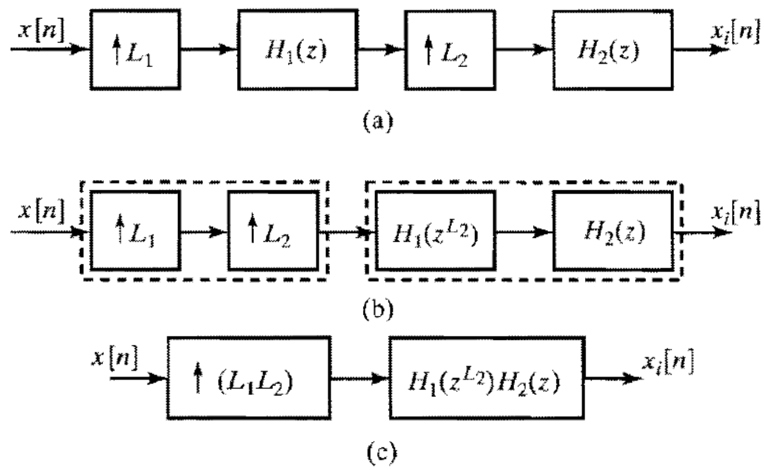
- Decimation or interpolation ratios are large
  - Significant reduction in computation through multistage
  - Interpolated FIR filter

$$H(z) = H_1(z)H_2(z^{M_1}) \qquad h[n] = h_1[n] * \sum_{k=-\infty}^{\infty} h_2[k]\delta[n - kM_1]$$



Multistage Decimation

- Low pass filter: pass band gain 0dB, stop band gain -40dB
- L=100: transition band $0.008\pi \sim 0.012\pi$, FIR order **709**
- L=10: transition band $0.09\pi \sim 0.11\pi$, FIR order **142**



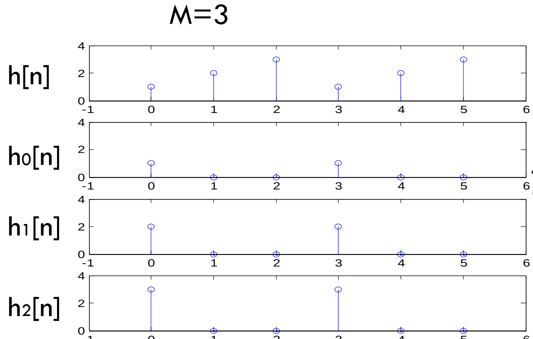
Polyphase decompositions

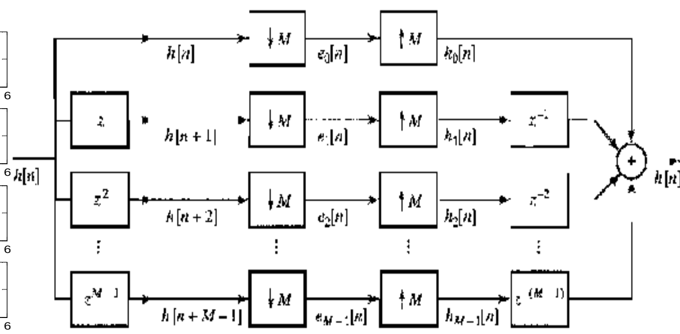
- h[n] as superposition of M subsequences

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k] \qquad k = 0, 1, \ldots, M-1$$

$$h_k[n] = \begin{cases} h[n+k], & n = \text{integer multiple of } M \\ 0, & \text{otherwise.} \end{cases}$$

M=3



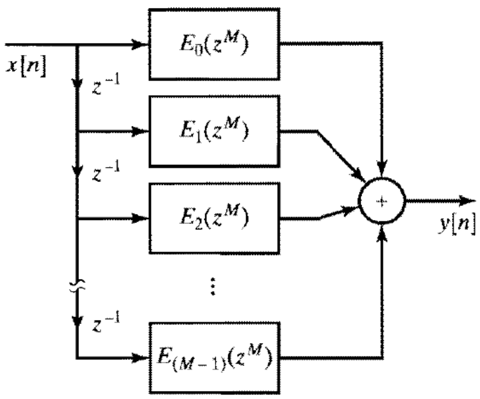


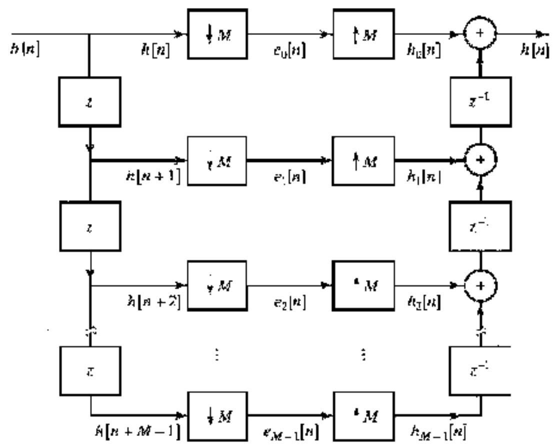
Polyphase decompositions

- Decompose to M parallel filters

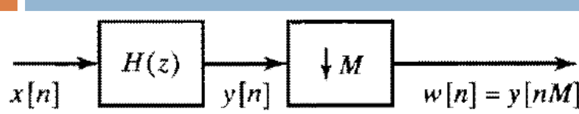
$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

$$e_k[n] = h[nM + k] = h_k[nM]$$



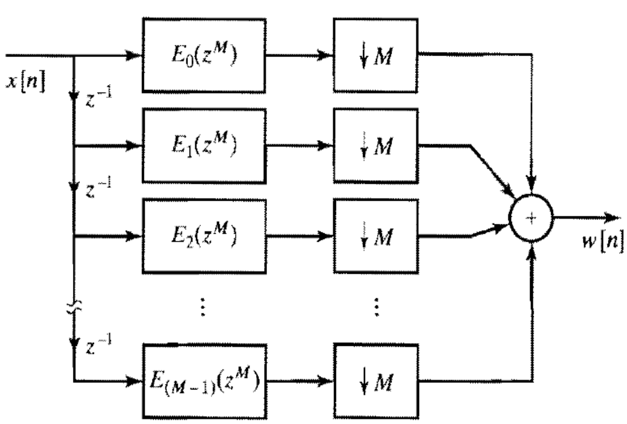


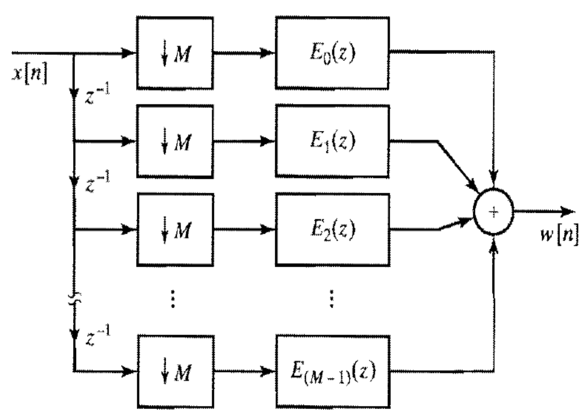
Polyphase implementation of decimation filters



$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

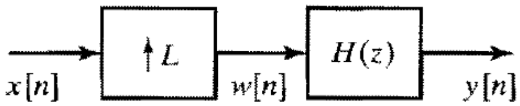
N mul + (N-1) add
N=100 total: 199





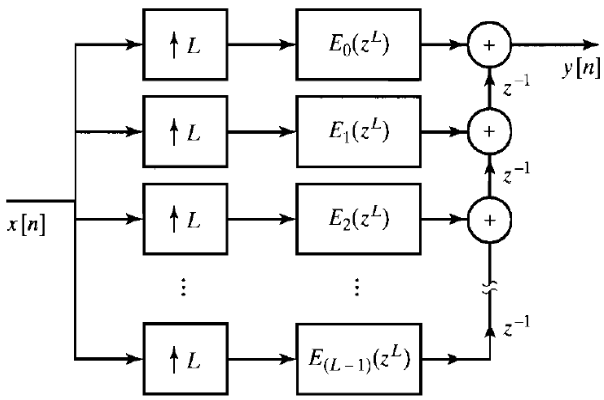
N/M mul + (N/M-1)+(M-1)/M add
N=100 M=10 total: 20

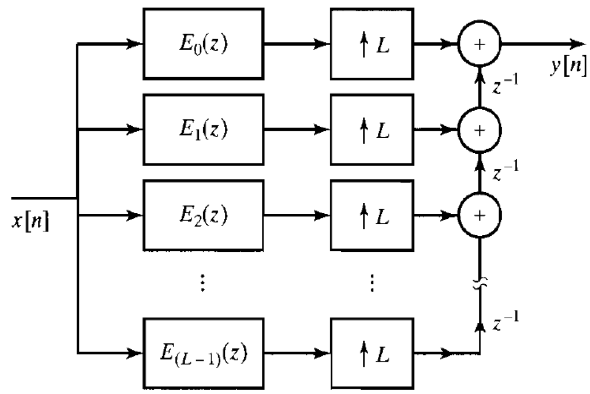
Polyphase implementation of interpolation filters



$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

NL mul + (NL-1) add
N=100 L=5 total: 999





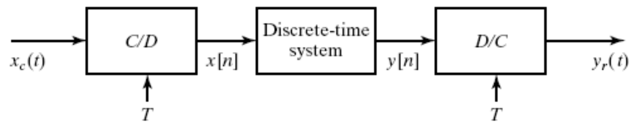
L(N/L) mul +L (N/L-1)+(L-1) add
N=100 L=5 total: 199

Contents

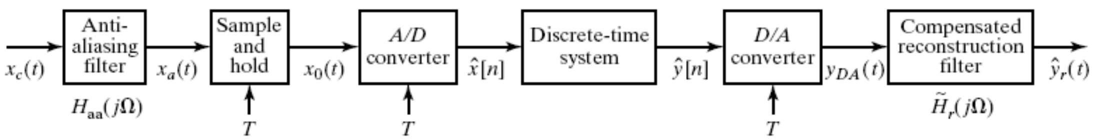
- Changing the sampling Rate
- Multirate signal processing
- Digital Processing of Analog Signals

Ideal Conversion

- Assume 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
  - Remove high frequencies (noises) that are not of interest
- A low-pass anti-aliasing filter would improve both aspects
- An ideal anti-aliasing filter

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

- The effective response is

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

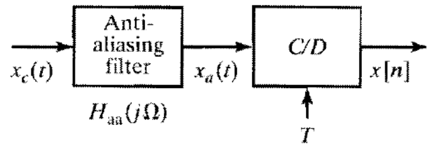
- In practice an ideal low-pass filter is not possible hence

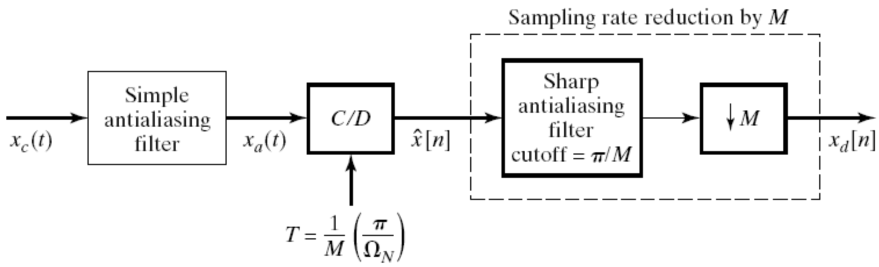
$$H_{eff}(j\Omega) \approx H_{aa}(j\Omega) H(e^{j\Omega T})$$

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

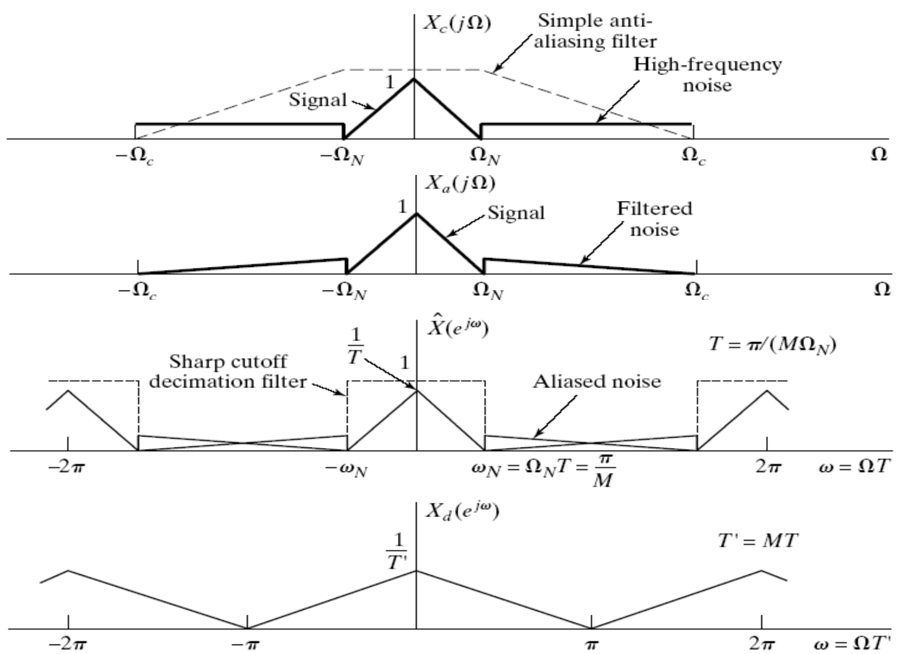
Oversampled A/D Conversion

- The idea is
  - to use a simple analog anti-aliasing filter
  - Use higher than required sampling rate
  - implement sharp anti-aliasing filter in discrete-time
  - Downsample to desired sampling rate



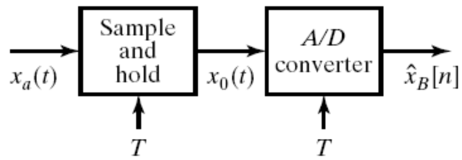


Example



Analog-to-Digital (A/D) Conversion

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

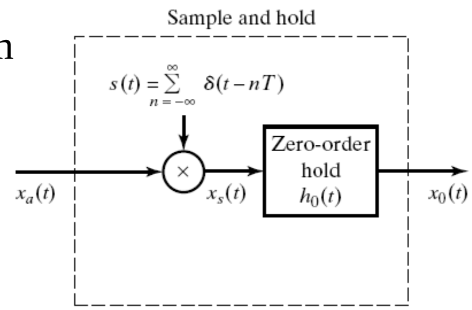


- The sample-and-hold device holds current/voltage constant
- The A/D converter converts current/voltage into finite-precisions number
- The ideal sample-and-hold device has the output

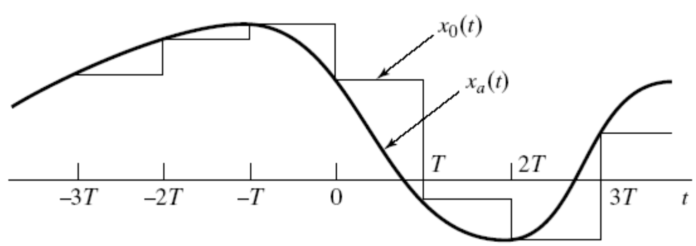
$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT) \qquad h_0(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{else} \end{cases}$$

Sample and Hold

- An ideal sample-and-hold system



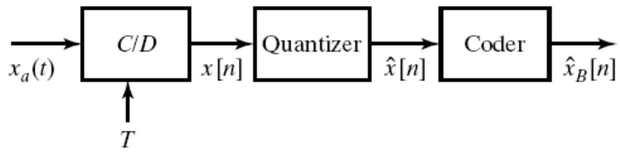
- Time-domain representation of sample-and-hold operation



A/D Converter Model

35

- An practical A/D converter can be modeled as



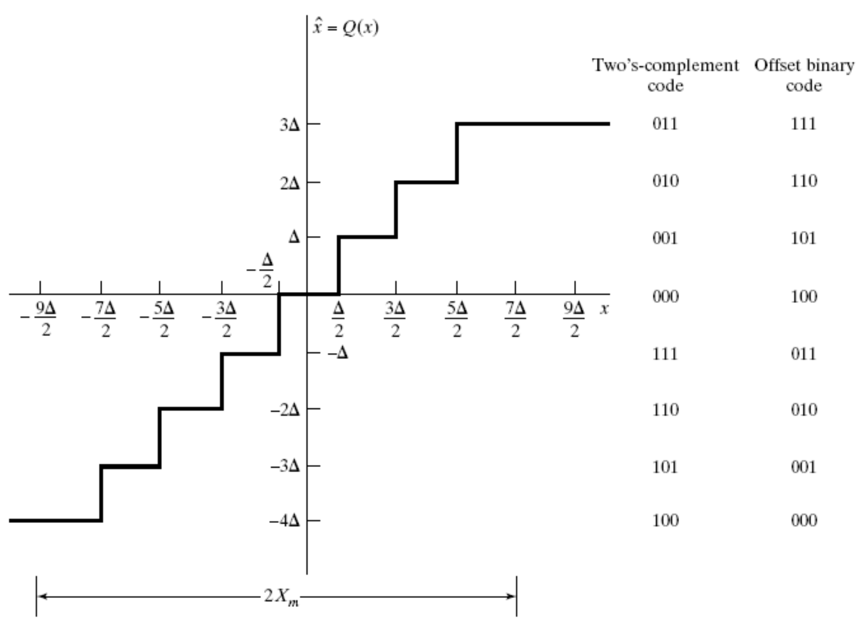
- The C/D converter represent the sample-hold-operation
- Quantizer transforms input into a finite set of numbers

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

- Most of the time uniform quantizers are used

Uniform Quantizer

36



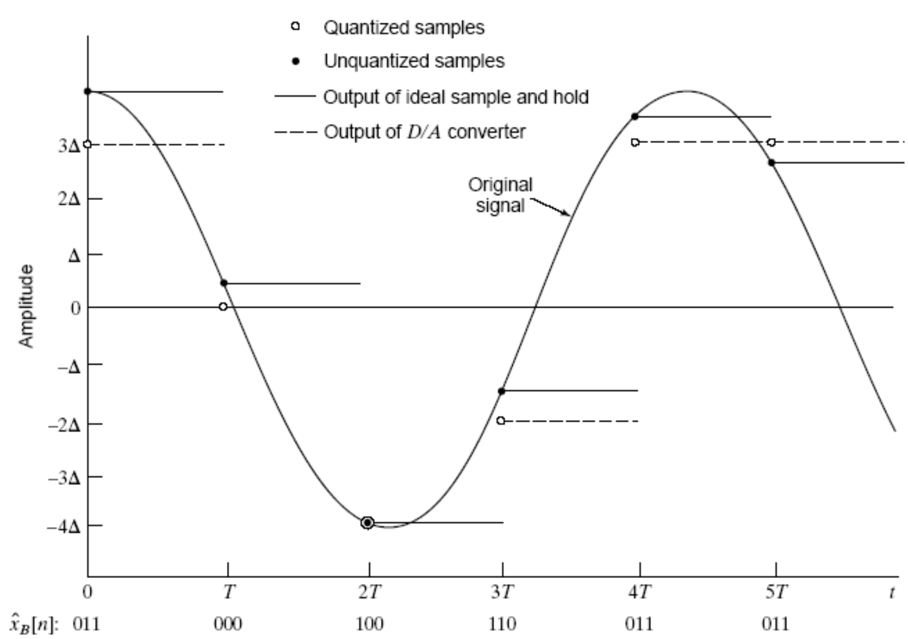
Two's Complement Numbers

- Representation for signed numbers in computers
- Integer two's-complement $\quad -a_0 2^B + a_1 2^{B-1} + \ldots + a_B 2^0$
- Fractional two's-complement $\quad -a_0 2^0 + a_1 2^{-1} + \ldots + a_B 2^{-B}$
- 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

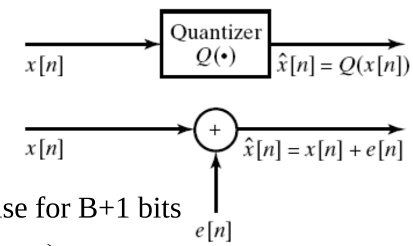


Quantization Error

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

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

  - As long the input does not clip
- Based on this fact we may use the following simplified model with assumption:
  - e[n] is uniformly distributed random variable
  - Is uncorrelated with the signal x[n]



- The variance of e[n] is then $\sigma_e^2 = \dfrac{\Delta^2}{12}$
- 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

$$X_r(j\Omega) = X\left(e^{j\Omega T}\right)H_r(j\Omega)$$
$$X\left(e^{j\Omega T}\right): \text{FT of sampled signal}$$
$$X_r(j\Omega): \text{FT of reconstructed signal}$$

- The ideal reconstruction filter

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

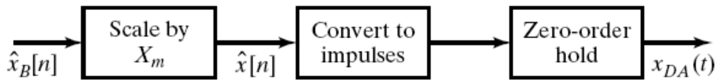
- The time domain reconstructed signal is

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

- 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

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

- To recover the desired signal component, a compensated reconstruction filter is needed:

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

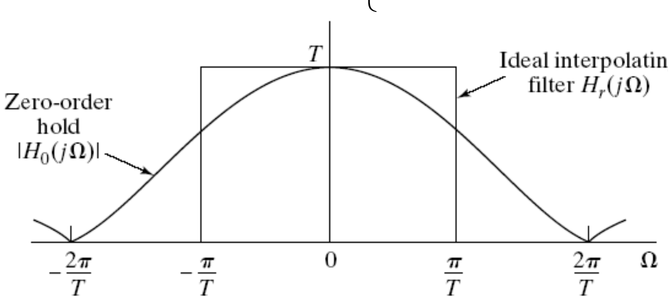
Compensated Reconstruction Filter

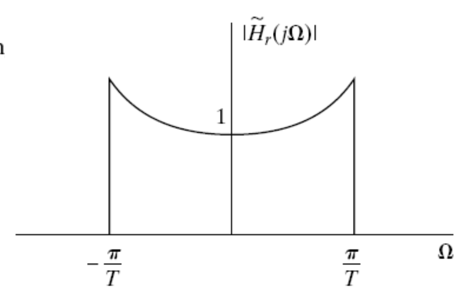
- The frequency response of zero-order hold is

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

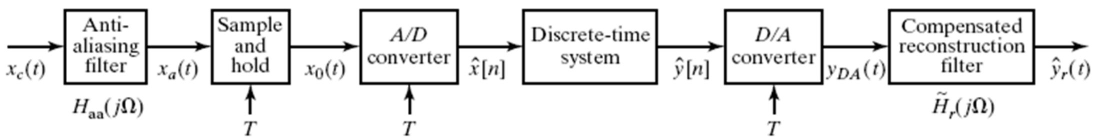
- Therefore the compensated reconstruction filter should be

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





Digital processing of analog signals



$$Y_a(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{aa}(j\Omega)X_c(j\Omega)$$

$$H_{\text{eff}}(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{aa}(j\Omega)$$

THE END