Fundamentals of Sigma-Delta Data Converters

Chun-Hsien Su
中央大學電機系　蘇純賢
chunsu@ee.ncu.edu.tw

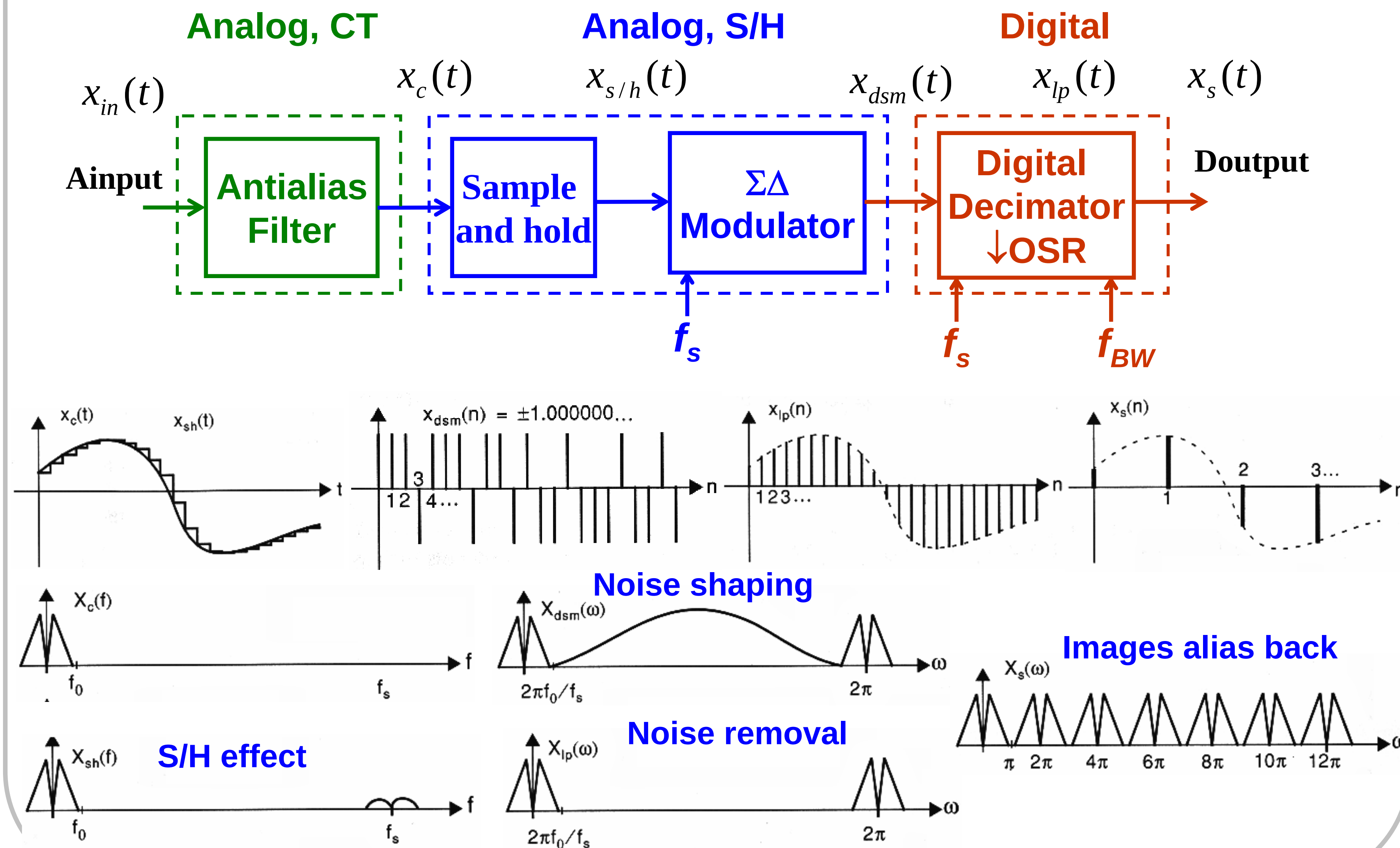
July, 2006

國立中央大學 電機工程學系
NATIONAL CENTRAL UNIVERSITY
DEPARTMENT OF ELECTRICAL ENGINEERING

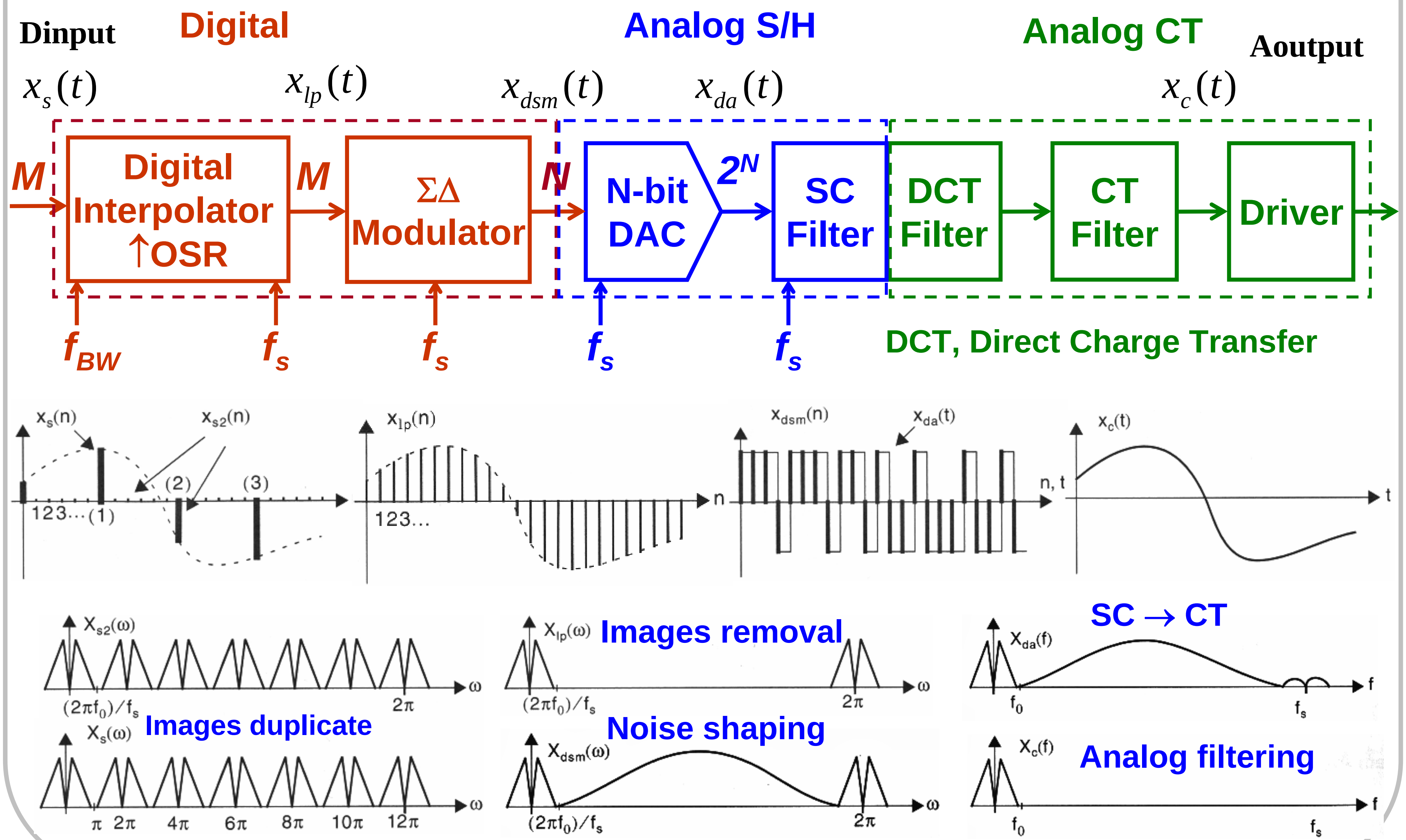
OUTLINE

1. System Architectures
2. Linearized Model for ΣΔ Modulators
3. Oversampling with Noise Shaping
4. Stability of ΣΔ Modulators
5. Modulator Architecture Summary
6. DC Input Behaviors of ΣΔ Modulators
7. Behavioral Simulations
8. ΣΔ Technique Applications

ΣΔ ADC Block Diagram and F/T Behaviors

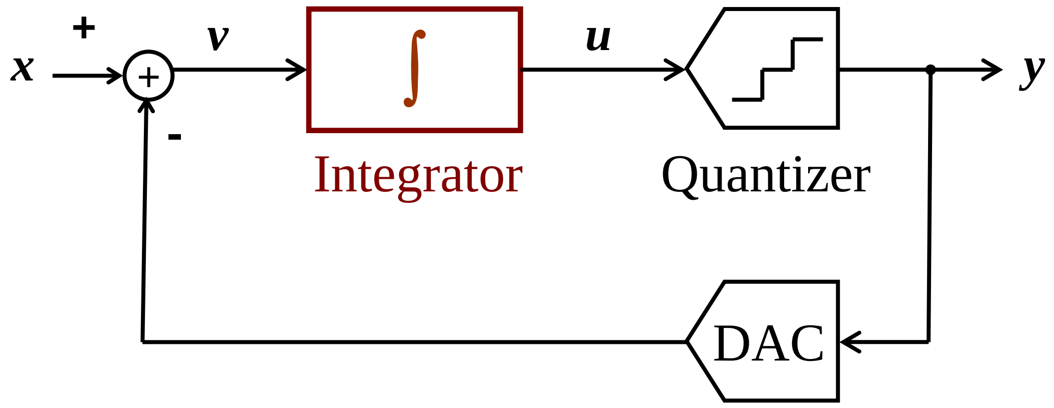


ΣΔ DAC Block Diagram and F/T Behaviors

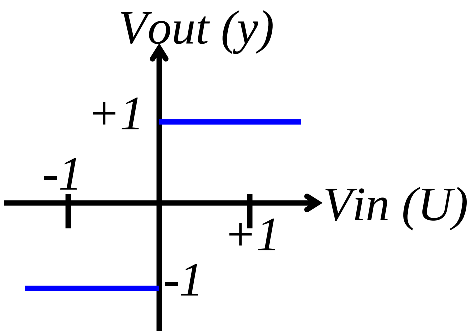


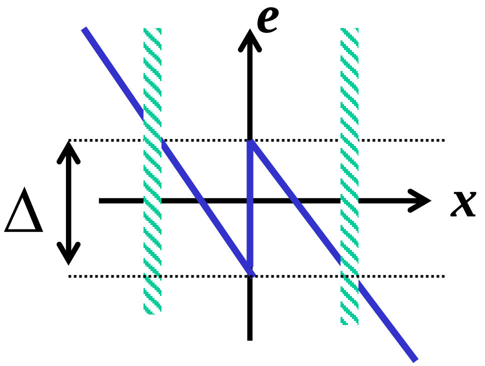
Principle of Sigma-Delta Modulation

The First-Order ΣΔ Modulator ⇒ Nonlinear System



Single-bit quantizer

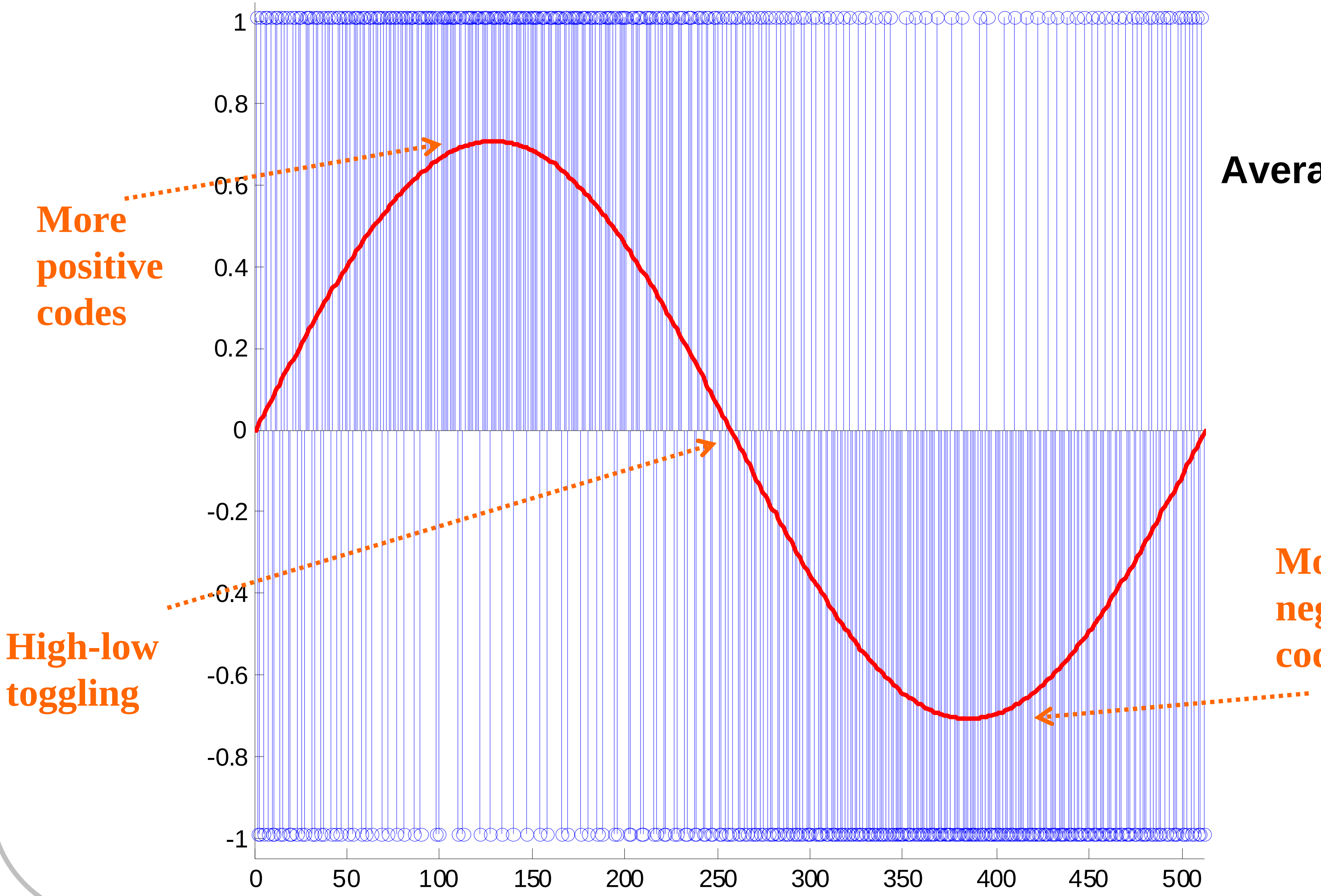




⇒ The modulator tracks its input and generates low-resolution pulse-density-modulated (PDM) output. (v: tracking error)

⇒If multibit, instead of single-bit, quantization is used, an N-bit D/A converter is needed in the feedback path.

Pulse Density Modulation (PDM) – One-Bit

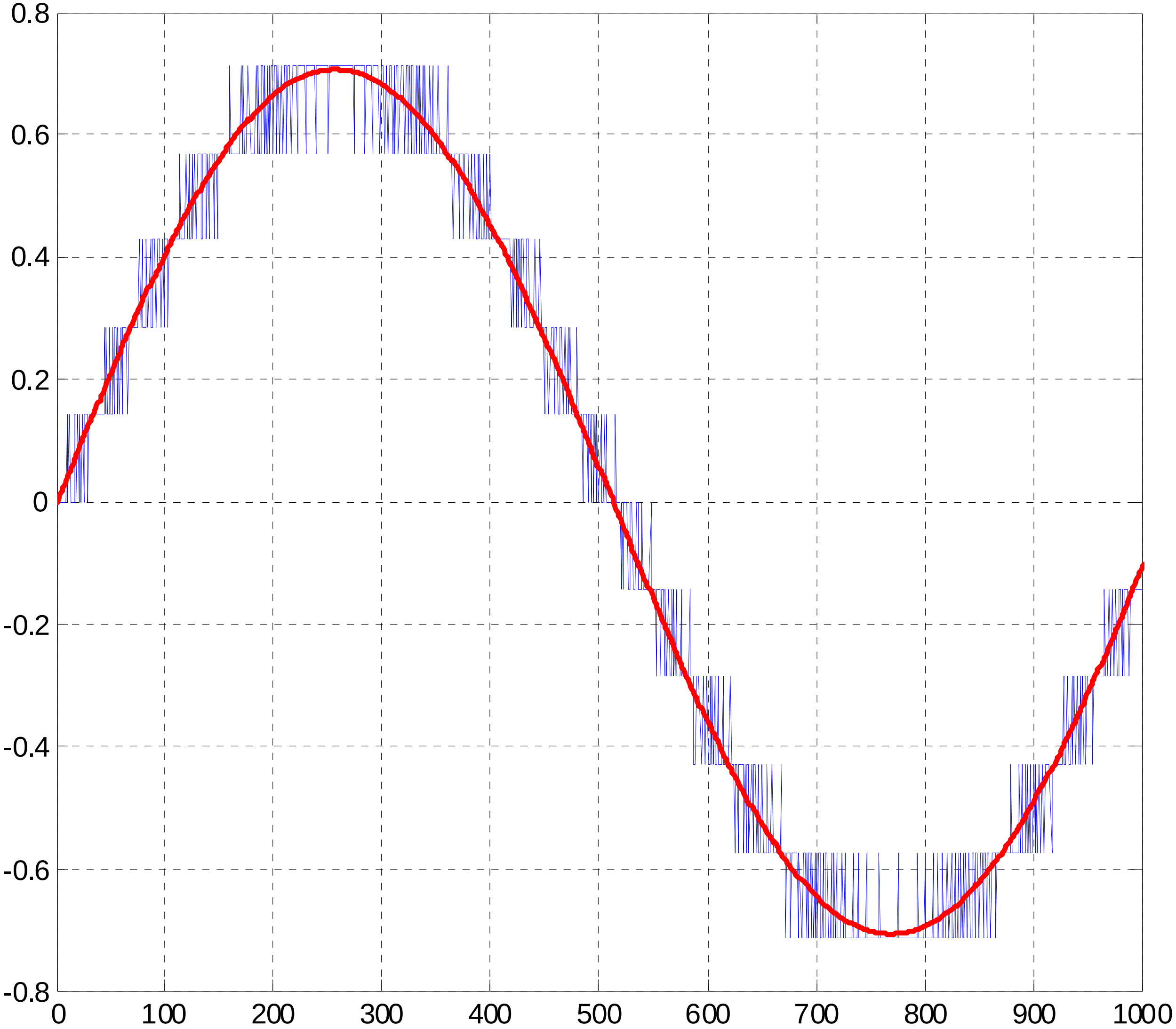


Average(y)=x

Pulse Density Modulation (PDM) – Multi-Bit

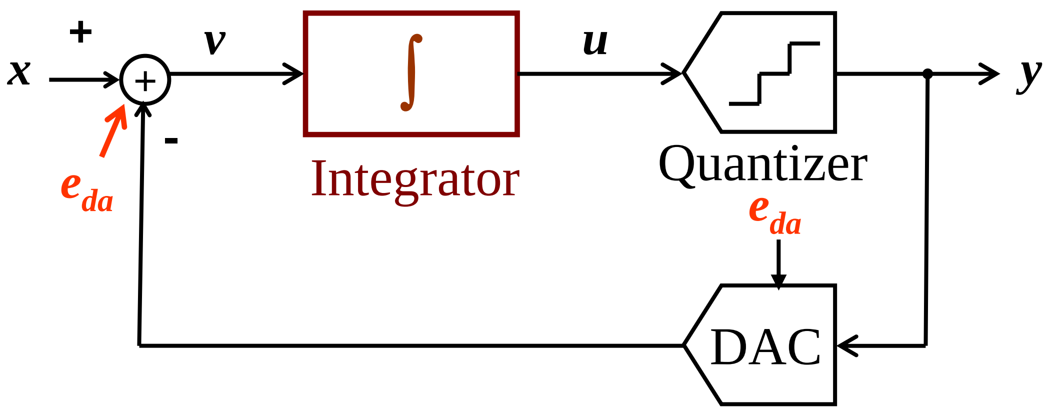
Four-bit quantizer

Each segment Average(y)=x



The Advantage of 1-Bit D/A Converters

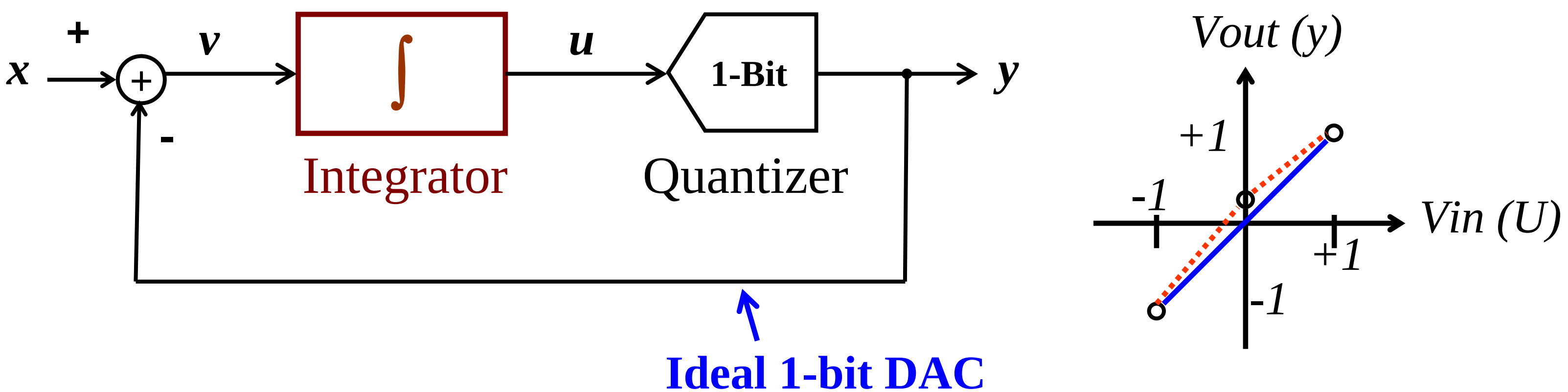
⇒ Multi-Bit DAC: Linearity Problem



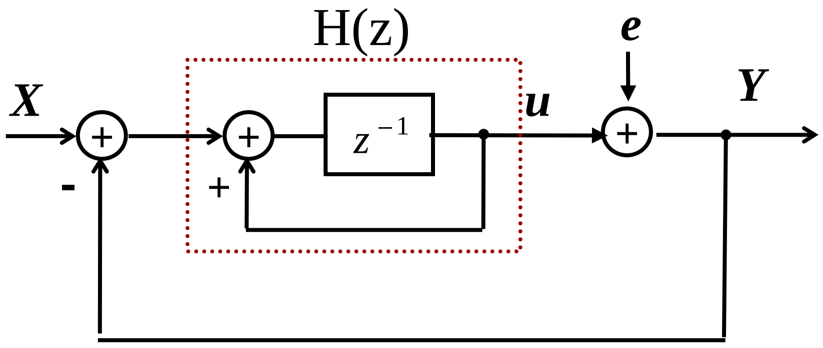
⇒ N-bit ADC with internal M-bit DAC: DAC accuracy INL< $1/2^N$ LSB

⇒ DAC errors show at the input terminal and affect the input signals.

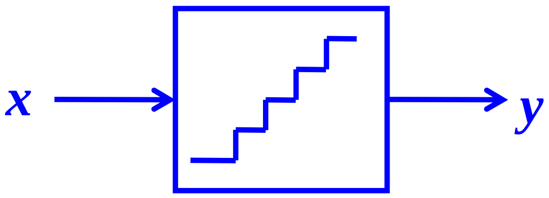
⇒ One-Bit DAC: inherently linear



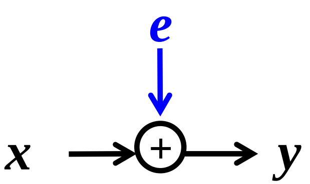
Linearized Model in z-Domain



***e*: quantization error**
(additive white noise)
White noise approximation



Quantizer



Input Signal

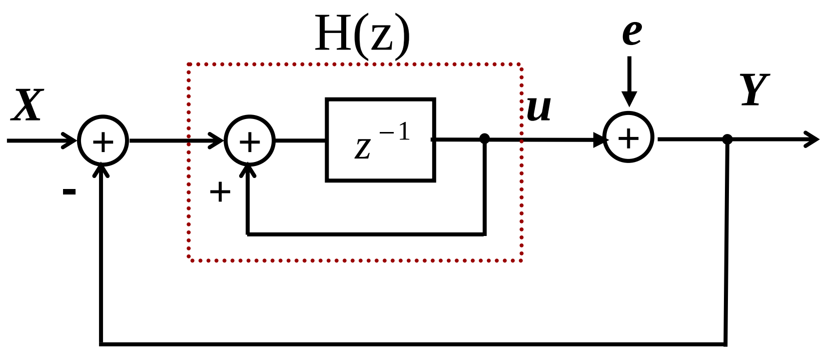
Quantization Error

$$Y(z) = H_X(z)\cdot X(z) + H_E(z)\cdot E(z)$$

Noise Transfer Function (NTF)

Signal Transfer Function (STF)

First-Order Modulator – Frequency Domain



***e*: quantization error (additive white noise)**

$$\begin{cases} U = H(X - Y) \\ Y = U + E \end{cases} \quad \& \quad H(z) = \frac{z^{-1}}{1 - z^{-1}}$$

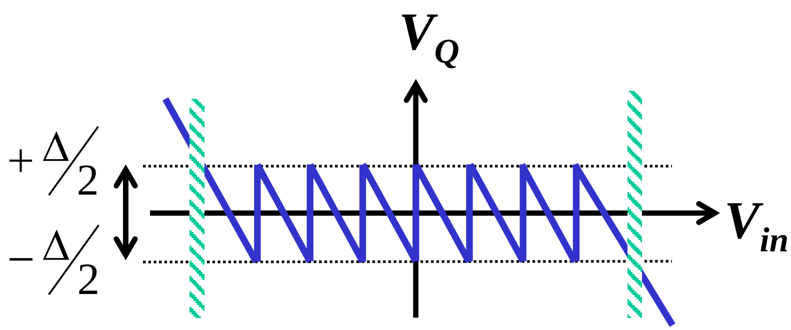
$$Y(z) = H_X(z) \cdot X(z) + H_E(z) \cdot E(z) = \frac{H(z)}{1 + H(z)} \cdot X(z) + \frac{1}{1 + H(z)} \cdot E(z)$$

$$Y(z) = z^{-1} \cdot X(z) + (1 - z^{-1}) \cdot E(z)$$

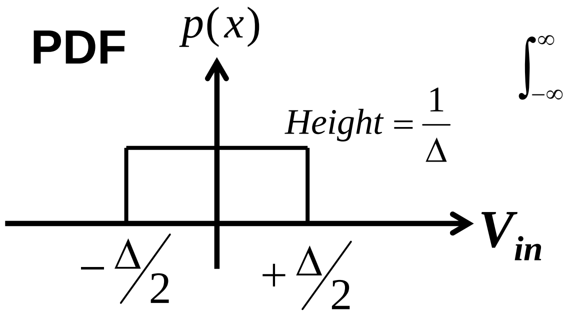
A unit delay **A highpass function: *E* being "shaped"**

White Noise Approximation

White noise approximation : if the quantizer input keeps busy and changes randomly between samples, the quantization error can be treated as an additive white noise. Under such conditions the generated white noise (quantization noise) has a uniformly-distributed probability density function (pdf) lying in the range of $\pm V_{LSB}/2$ statistically.



$\Rightarrow$

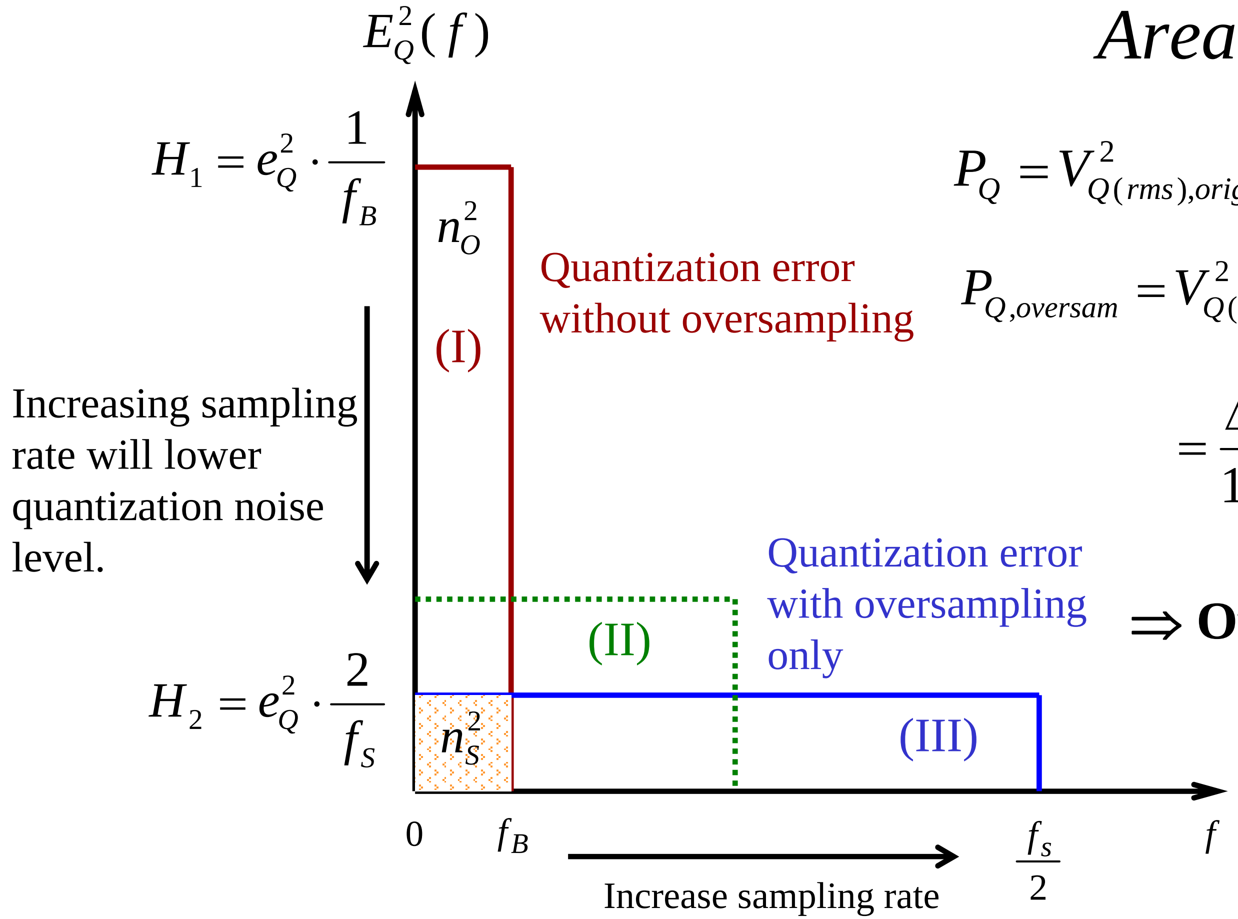


$$\int_{-\infty}^{\infty} p(x)dx = 1$$

$$\Rightarrow V_{Q(avg)}^2 = \int_{-\infty}^{\infty} x \cdot p(x)dx = \frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} xdx = 0$$ **($\Rightarrow$ avergae=0)**

$$\Rightarrow V_{Q(rms)}^2 = \int_{-\infty}^{\infty} x^2 \cdot p(x)dx = \frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} x^2 dx = \frac{\Delta^2}{12}$$

Oversampling



$$Area_I = Area_{III}$$

$$P_Q = V^2_{Q(rms),orig} = \frac{\Delta^2}{12}$$

$$P_{Q,oversam} = V^2_{Q(rms),orig} \cdot \frac{2f_B}{f_S} = \frac{\Delta^2}{12} \cdot \frac{2f_B}{f_S}$$

$$= \frac{\Delta^2}{12} \cdot \frac{1}{OSR}$$

$\Rightarrow$ **Oversampling ratio (OSR)**

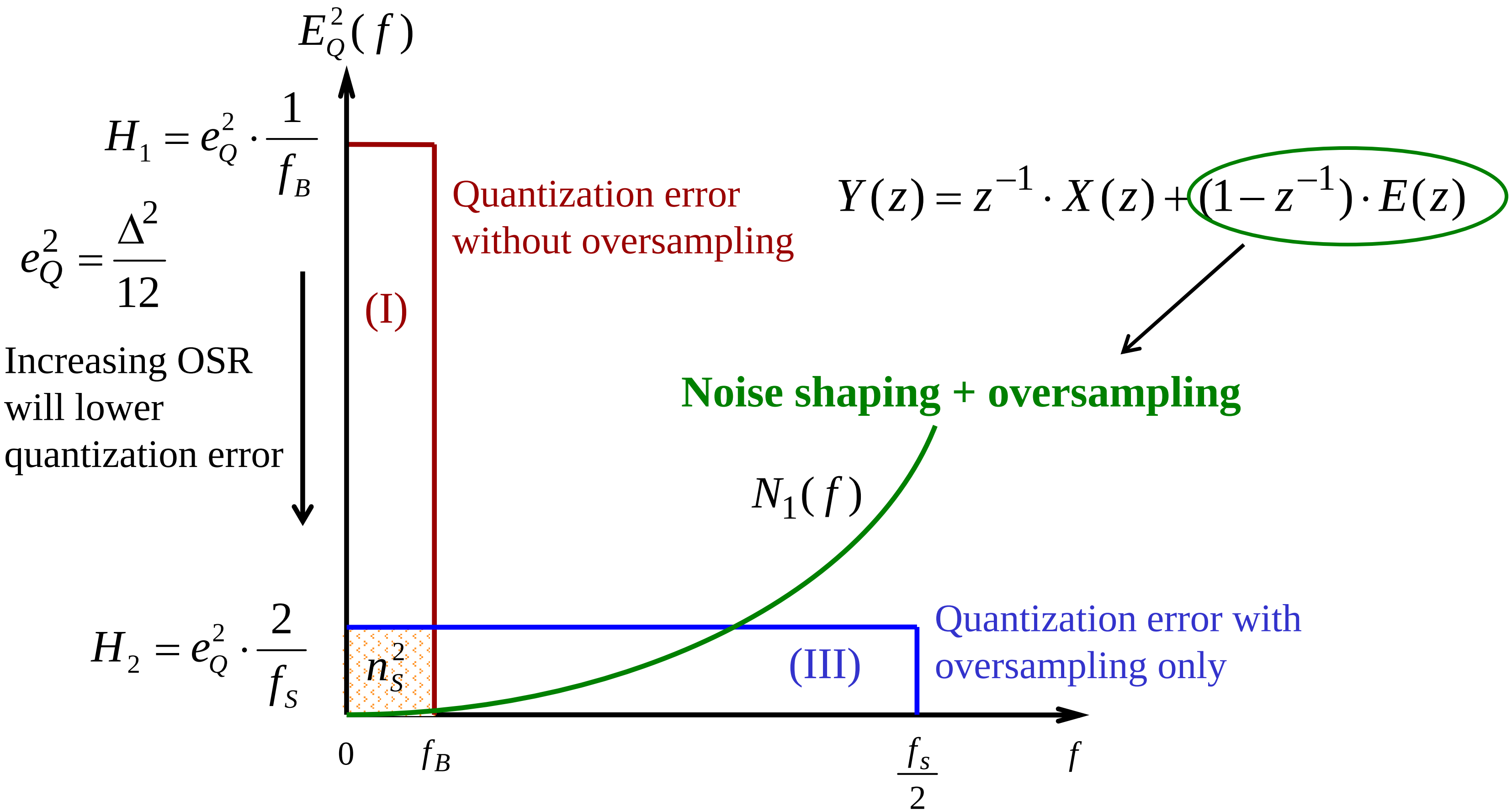
$$OSR = \frac{n_O^2}{n_S^2} = \frac{f_S}{2f_B}$$

$$SNR = 10\log\left(\frac{P_{in}}{P_Q}\right) = 6.02b + 1.761 + 10\log(OSR)$$

$\Rightarrow$ Each doubling OSR increases SNR by 3dB.

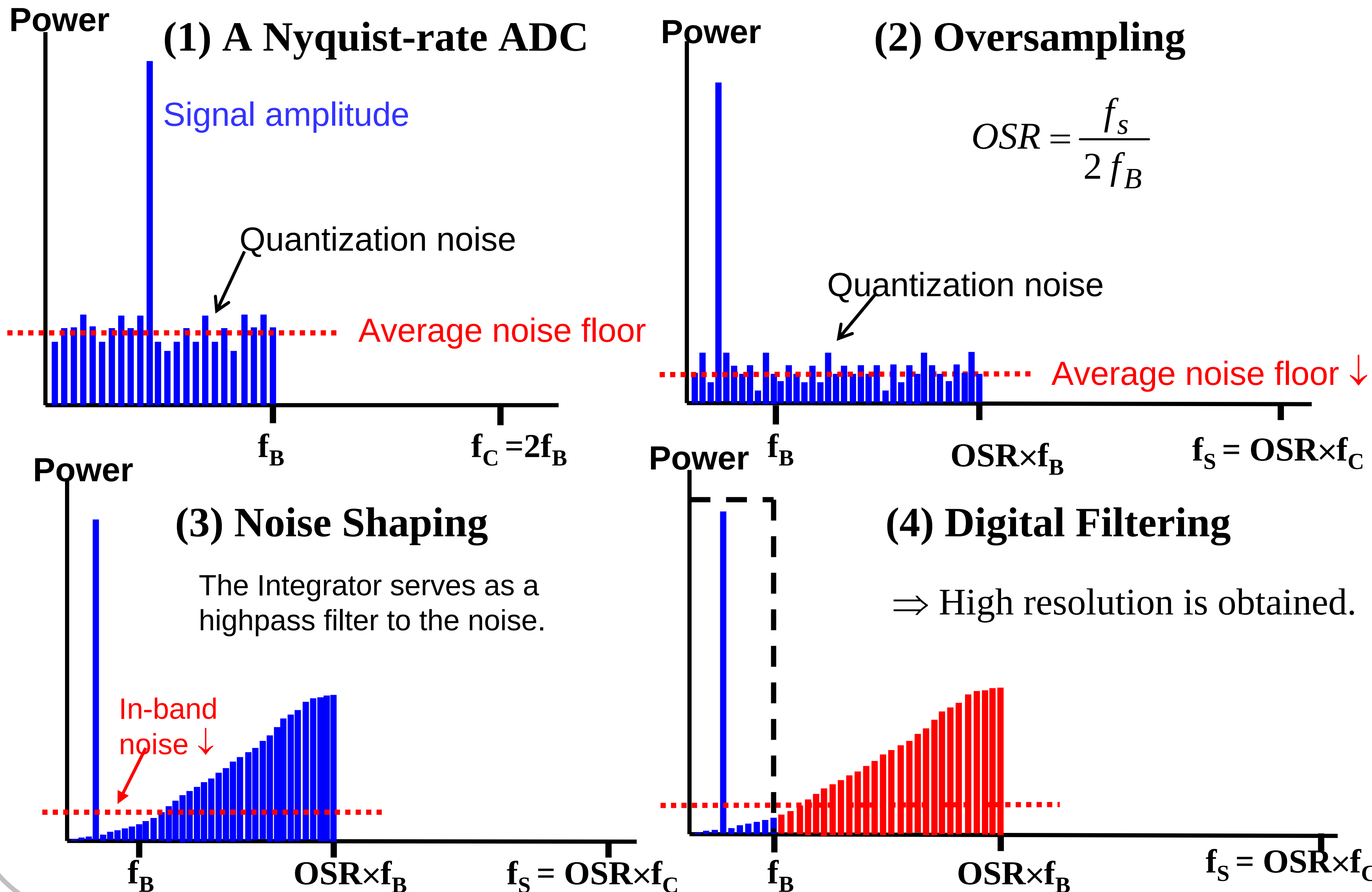
Oversampling with Noise Shaping

Power Spectral Density of Shaped Quantization Noise



$\Rightarrow$ **Inband quantization noise power greatly reduced.**

Conceptual Figures of ΣΔ Modulation



Noise Power Calculations of First-Order ΣΔM

NTF of the First-Order ΣΔM

$$NTF(f) = 1 - z^{-1}\Big|_{z=e^{\frac{2\pi f}{f_s}}} = 1 - e^{-\frac{2\pi f}{f_s}} = \frac{e^{\frac{\pi f}{f_s}} - e^{-\frac{\pi f}{f_s}}}{2j} \times 2j \times e^{-\frac{\pi f}{f_s}} = \sin\left(\frac{\pi f}{f_s}\right) \times 2j \times e^{-\frac{\pi f}{f_s}}$$

Quantization Spectral Density

$T = 1/f_s$

$$N_1(f) = E(f) \cdot \left|1 - z^{-1}\right| = \sqrt{\frac{\Delta^2}{12} \cdot \frac{2}{f_s}} \cdot \sqrt{\left|1 - z^{-1}\right|^2_{z=e^{j2\pi fT}}} = 2e_Q \cdot \sqrt{\frac{2}{f_s}} \cdot \sin(\pi f T)$$

In-band Quantization Noise Power:

$$n_1^2 = \int_0^{f_B} \left|N_1(f)\right|^2 df \approx e_Q^2 \cdot \frac{\pi^2}{3} \cdot \left(\frac{1}{OSR}\right)^3$$

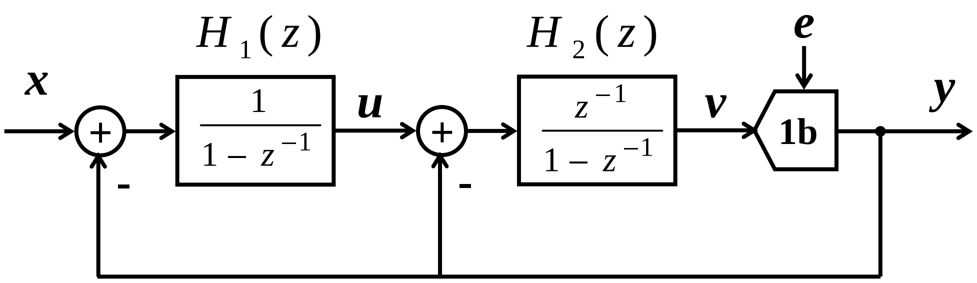
$\Rightarrow \sin(\pi f T) \approx \pi f T$ **is used.**

$$SNR = 10\log\left(\frac{P_{in}}{P_Q}\right) = 6.02b + 1.761 - 5.17 + 30\log(OSR)$$

⇒ Each doubling OSR increases SNR by 9dB.

Second-Order ΣΔ Modulation

Ideal Model



$$\begin{cases} U = H_1(X - Y) \\ V = H_2(U - Y) \\ Y = V + E \end{cases} \quad \& \quad \begin{cases} H_1(z) = \dfrac{1}{1-z^{-1}} \\ H_2(z) = \dfrac{z^{-1}}{1-z^{-1}} \end{cases}$$

Note: $H_1(z)$ is delayless and is hard to be implemented by analog circuits.

$$\Rightarrow \boxed{Y(z) = z^{-1} \cdot X(z) + (1 - z^{-1})^2 \cdot E(z)}$$

Quantization Noise Power of the Second-Order ΣΔM

NTF of the Second-Order ΣΔM

$$\left|NTF(f)\right| = \left|(1 - z^{-1})^2\Big|_{z=e^{\frac{2\pi f}{f_s}}}\right| = \left[2\sin\left(\frac{\pi f}{f_s}\right)\right]^2$$

Quantization Spectral Density

$$N_2(f) = E(f)\cdot\left|1 - z^{-1}\right|^2 = \sqrt{\frac{\Delta^2}{12}\cdot\frac{2}{f_s}}\cdot\left|1 - z^{-1}\right|^2_{z=e^{j2\pi fT}} = e_Q\cdot\sqrt{\frac{2}{f_s}}\cdot 4\cdot\sin^2(\pi fT)$$

In-band Quantization Noise Power:

$$n_2^2 = \int_0^{f_B}\left|E(f)\cdot(1 - z^{-1})^2\right|^2 df = e_Q\cdot\frac{\pi^4}{5}\cdot\left(\frac{1}{OSR}\right)^5$$

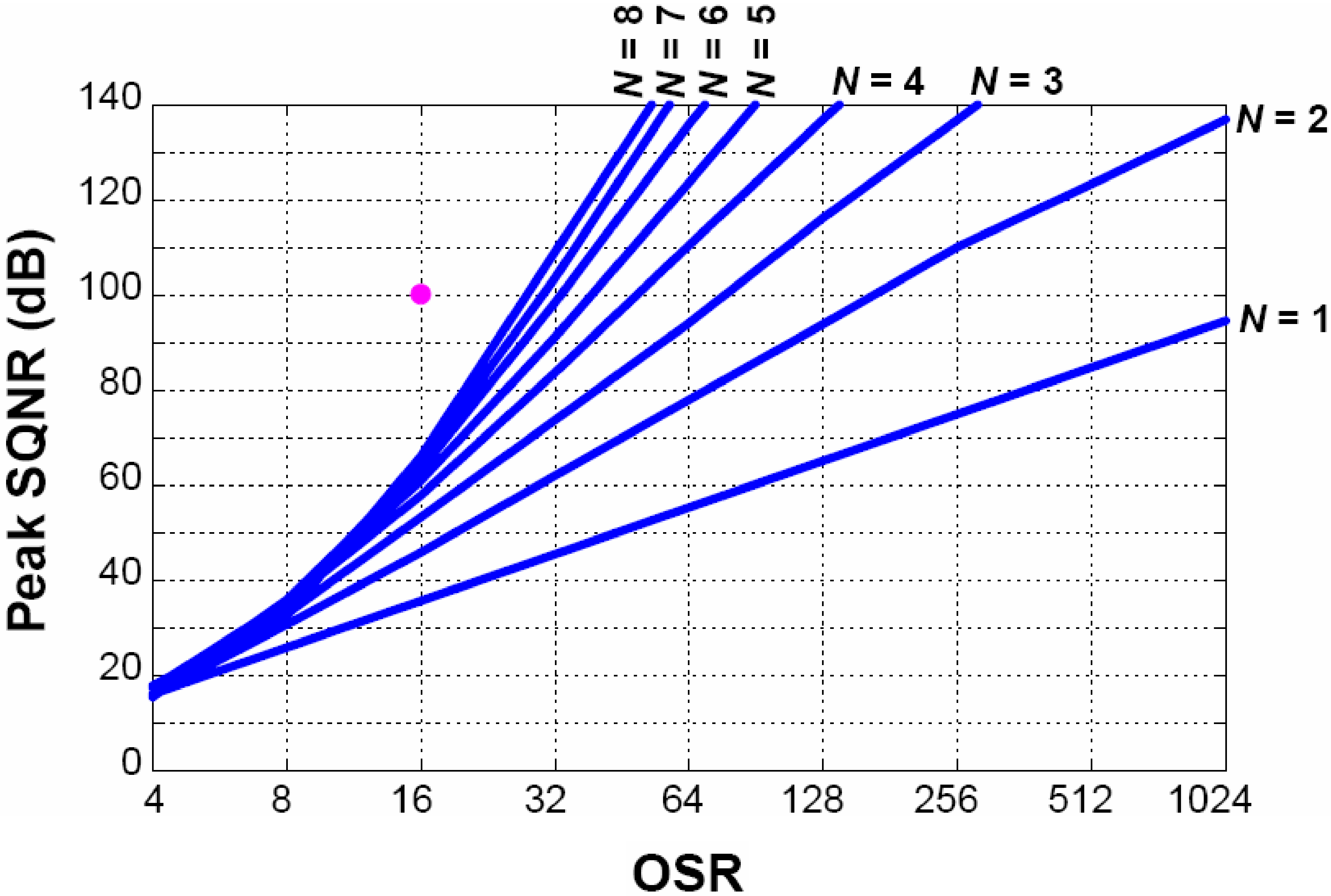
$$SNR = 10\log\left(\frac{P_{in}}{P_Q}\right) = 6.02b + 1.761 - 12.9 + 50\log(OSR)$$

⇒ Each doubling OSR increases SNR by 15dB.

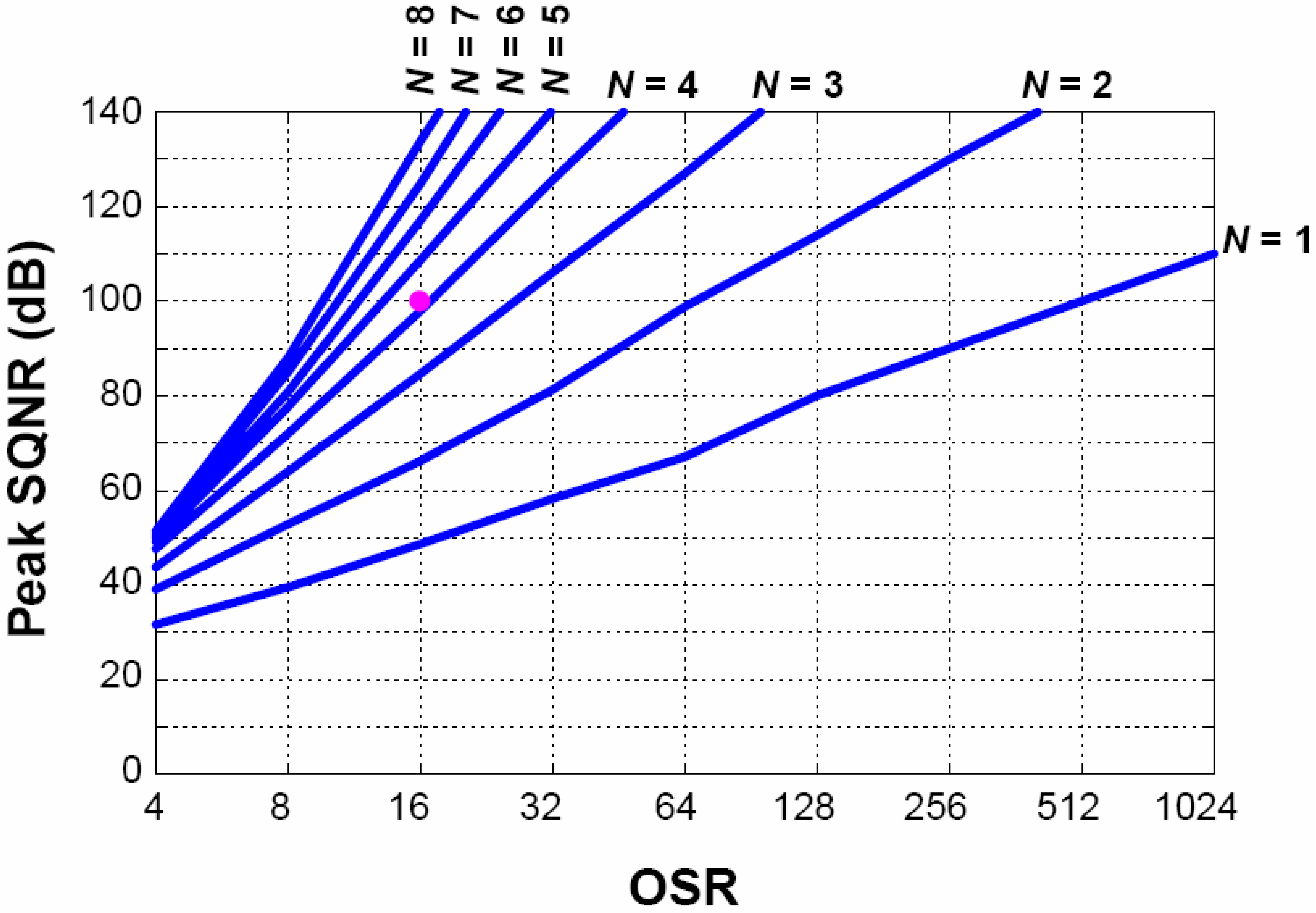
Noise Power up to *L*th-Order ΣΔ Modulators

Structure	In-band noise power	Noise reduced by each doubling the OSR
Nyquist-rate	$e_Q^2 = \frac{\Delta^2}{12}$	-
Oversampling only	$n_0^2 = \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot f_B = e_Q^2 \cdot \frac{1}{OSR}$	3dB (half a bit)
First-order ΣΔ modulation	$n_1^2 = e_Q^2 \cdot \frac{\pi^2}{3} \cdot (\frac{1}{OSR})^3$	9dB (1.5 bits)
Second-order ΣΔ modulation	$n_2^2 = e_Q^2 \cdot \frac{\pi^4}{5} \cdot (\frac{1}{OSR})^5$	15dB (2.5 bits)
Lth-order ΣΔ modulation	$n_L^2 = e_Q^2 \cdot \frac{\pi^{2L}}{2L+1} \cdot (\frac{1}{OSR})^{2L+1}$	3(2L+1) dB, (L+1/2) bits

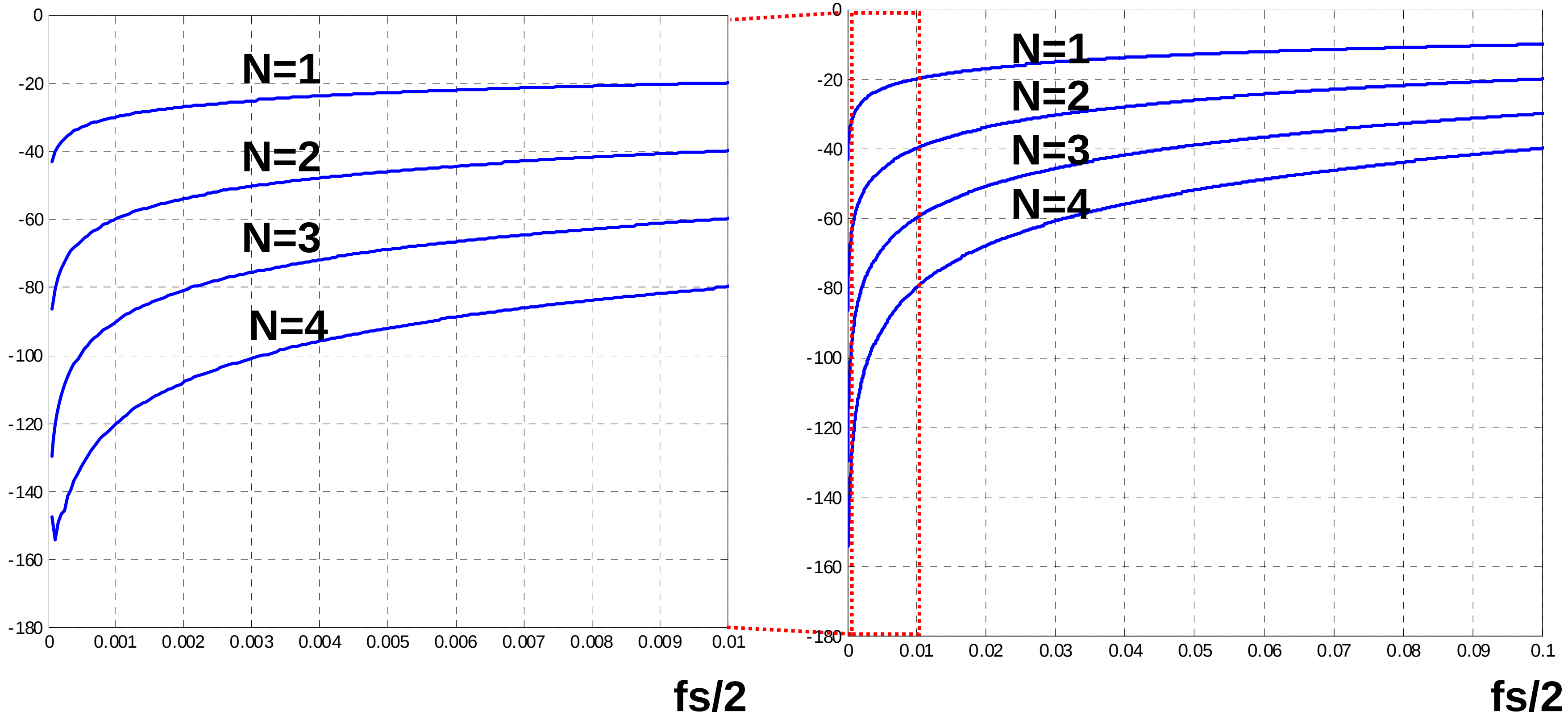
OSR versus for Single-Bit ΣΔ Modulators



OSR versus for 3-Bit ΣΔ Modulators



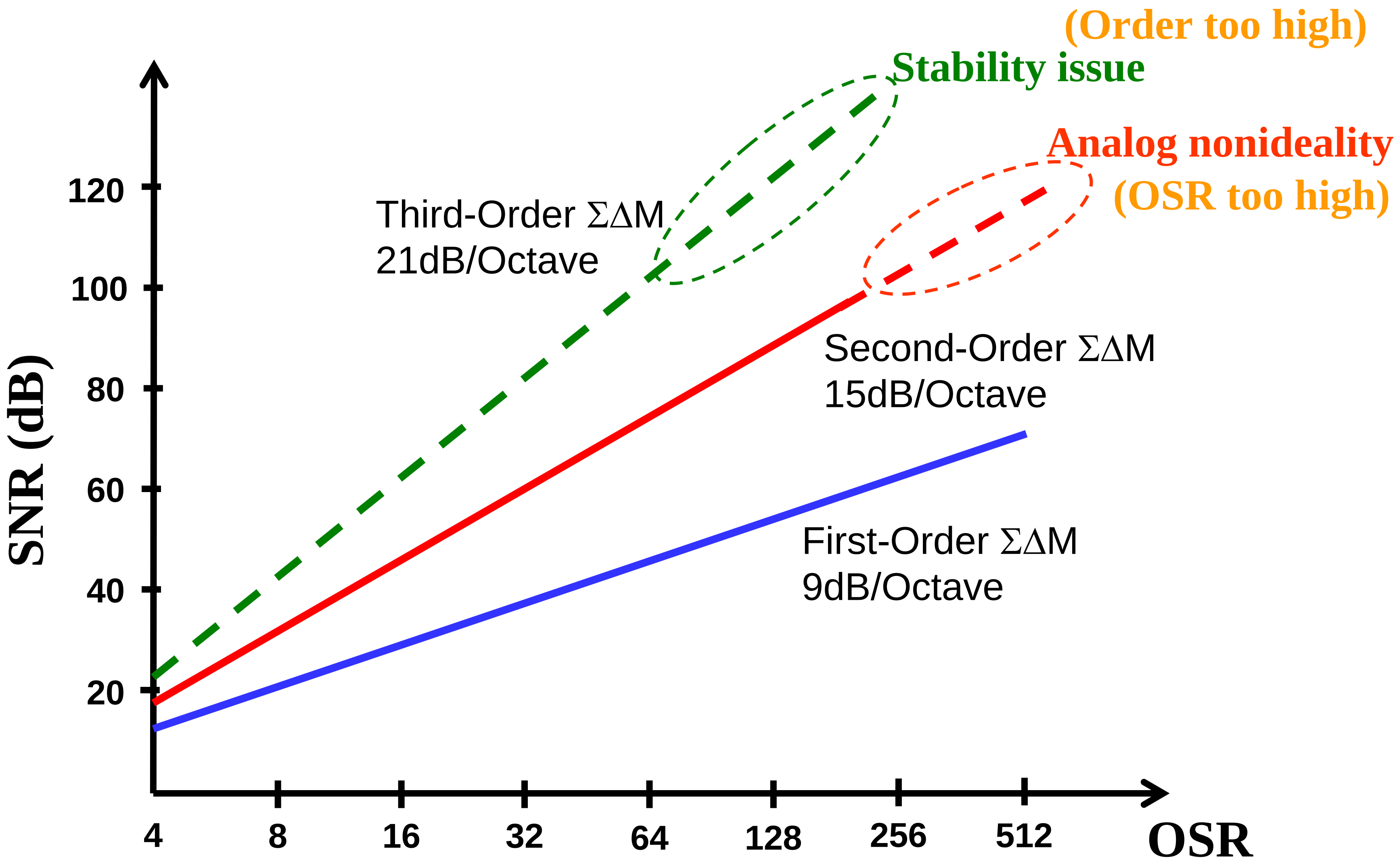
NTFs vs. Modulator Order



$\Rightarrow$ **High-Order Modulators achieve better SNR.**

Can this derivation hold valid up to *L*th-Order?

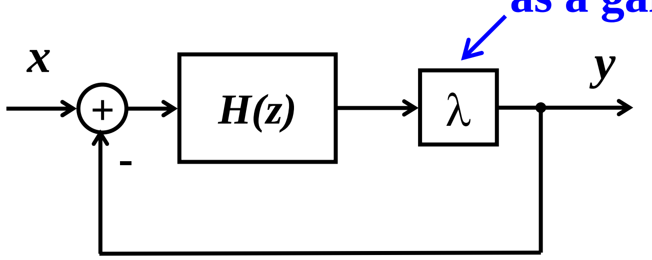
Oversampling Ratio vs. Modulator Order



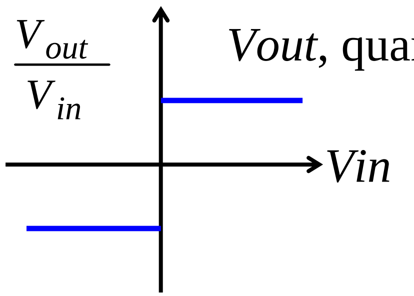
Variable Gain Model

Stability Analysis

[Fiez, 1994]



$$\lambda = \frac{V_{out}}{V_{in}}$$

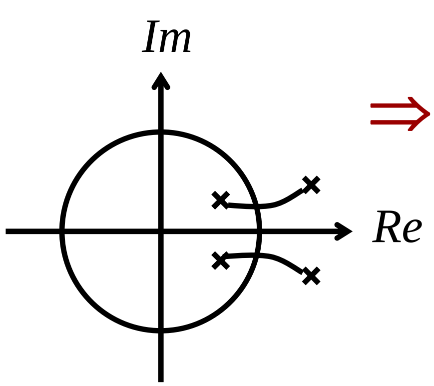


$$\frac{y}{x} = \frac{\lambda H}{1 + \lambda H}$$

characteristics equation

$\lambda \to 0$ **Large input, small output**

$\lambda \to \infty$ **Small input, Large output**

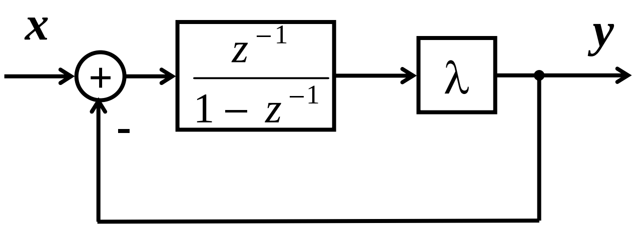


⇒ Stability criterion: all roots inside unity circle.

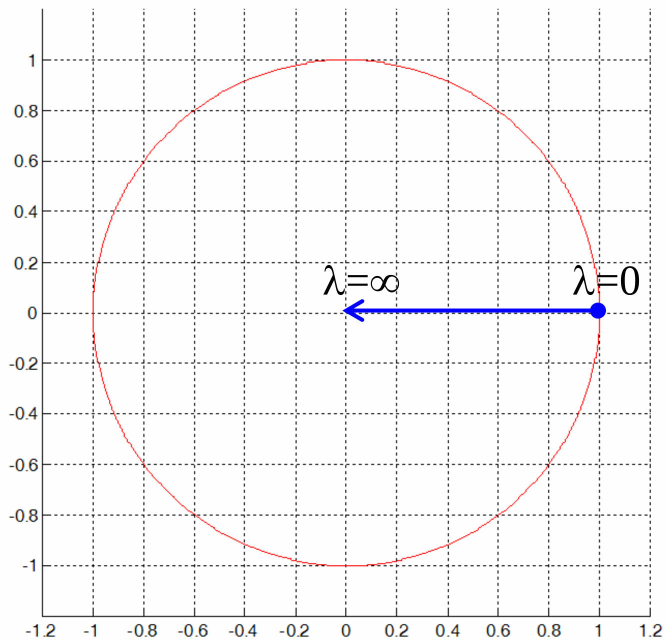
⇒ Varying λ and inspect root locus:
They should be inside unit circle.

Stability Analysis of ΣΔ Modulators (I)

(1) First-order ΣΔM

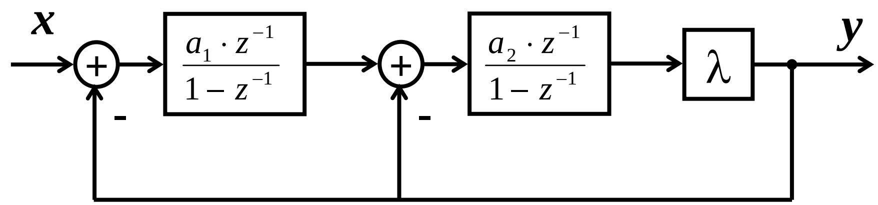


$$\frac{y}{x} = \frac{\lambda a_1 \cdot z^{-1}}{1 + (\lambda a_1 - 1)\cdot z^{-1}} \qquad a_1 = 1$$



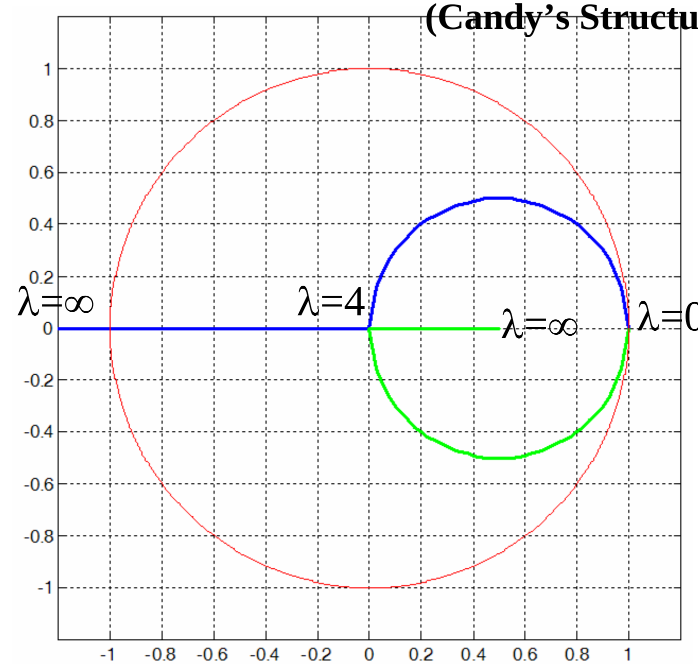
⇒ Absolutely Stable

(2) Second-order ΣΔM



$$\frac{y}{x} = \frac{\lambda a_1 a_2 \cdot z^{-2}}{1 + (\lambda a_2 - 2)\cdot z^{-1} + (\lambda a_1 a_2 - \lambda a_2 + 1)\cdot z^{-2}}$$

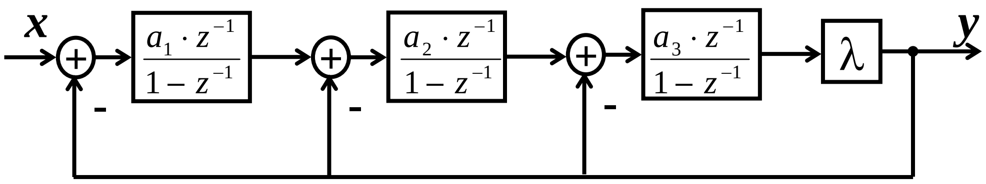
(Candy's Structure: ½ , ½)



⇒ Stable

Stability Analysis of ΣΔ Modulators (II)

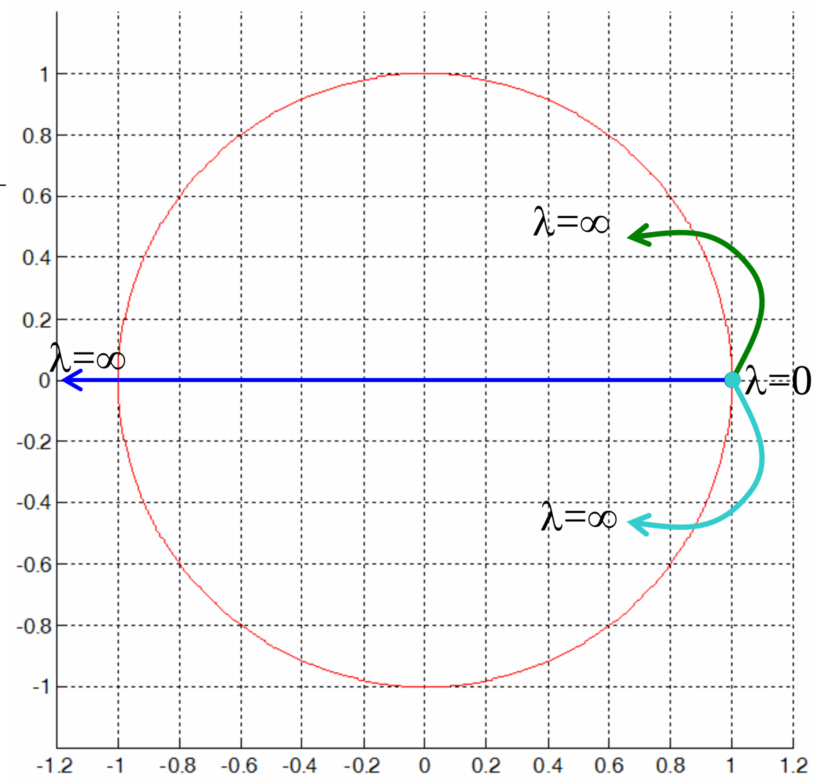
(3) Third-order ΣΔM



$$\frac{y}{x} = \frac{\lambda a_1 a_2 a_3 \cdot z^{-1}}{1 + (\lambda a_3 - 3)\cdot z^{-1} + \left[\lambda a_3 (a_2 - 2) + 3\right]\cdot z^{-2} + \left[\lambda a_3 (a_1 a_2 - a_2 + 1) - 1\right]\cdot z^{-3}}$$

⇒ Conditionally Stable

$a_1 = a_2 = 0.5 \qquad a_3 = 0.25$



Conditionally stable ΣΔM is not acceptable: when it is unstable, it diverges and cannot return to normal operation.

Stability is a function of
Input amplitude,
Input frequency, and
Loop coefficients.

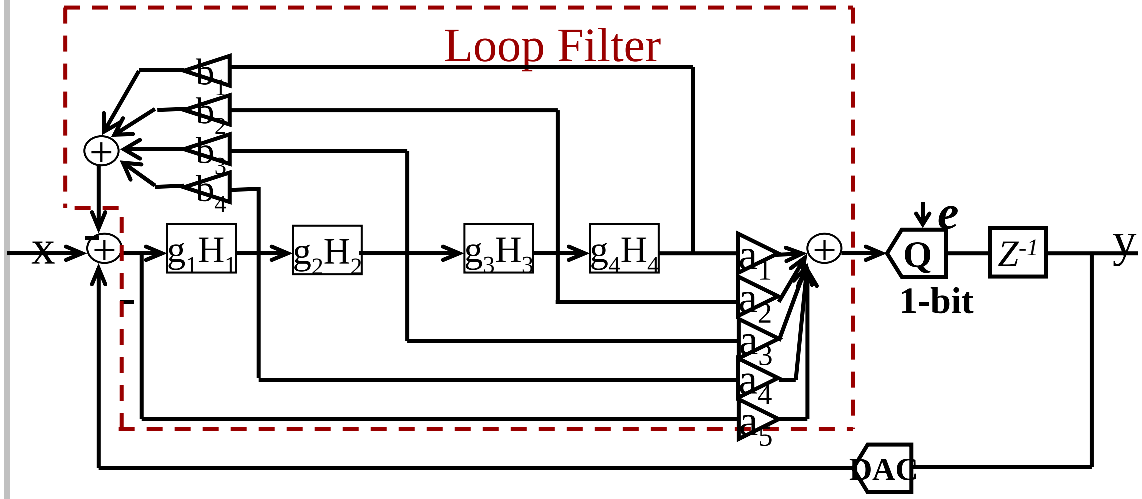
⇒ Reduced input range for high-order ΣΔM.

Stability Analysis of ΣΔ Modulators (III)

(4). High-Order Single-Loop Modulator (Interpolative Structure)

[Sodini 1990]

Filter Synthesis Method

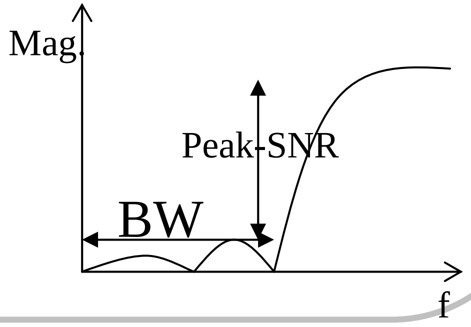


Loop Coefficients

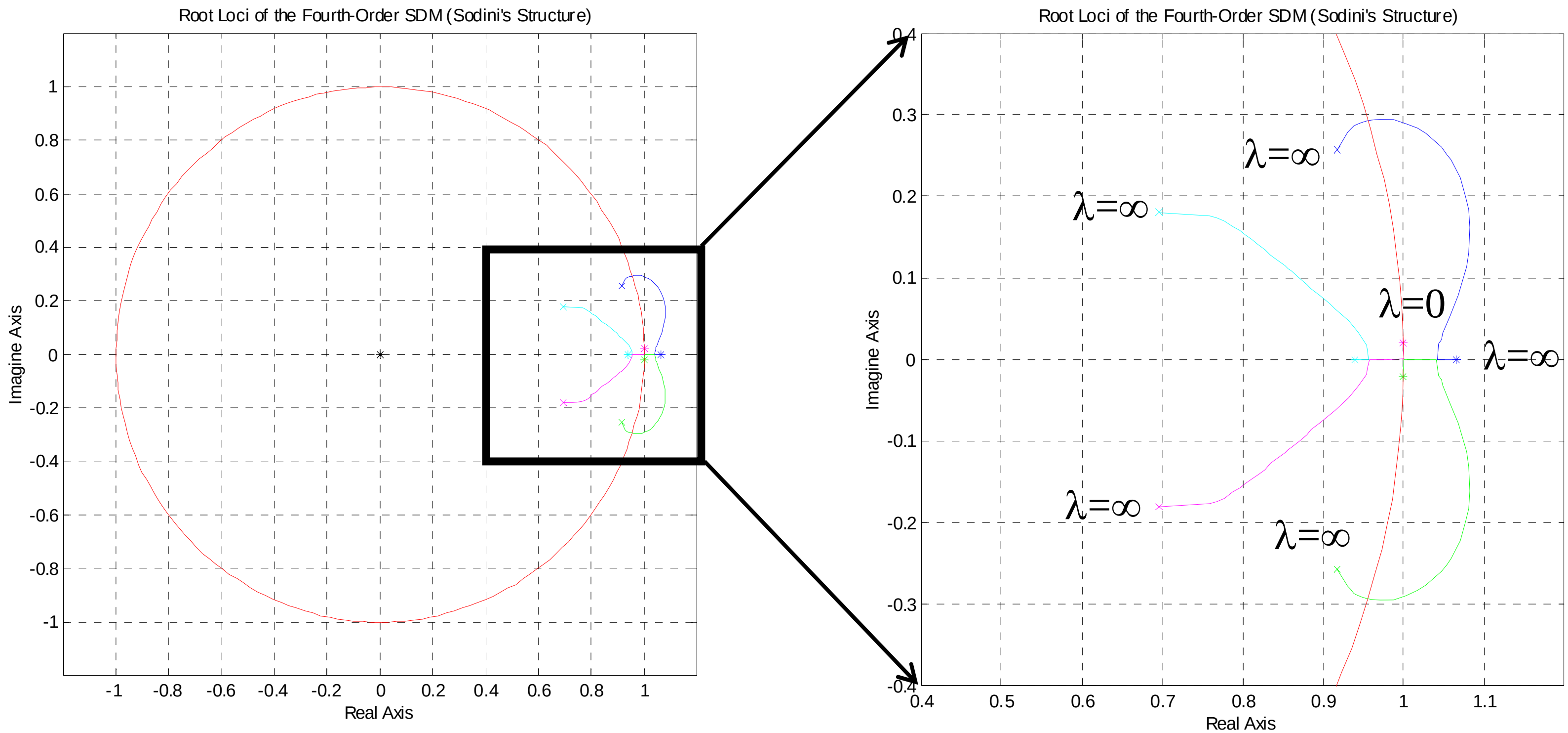
Feedforward		Feedback	
Coeff.	**Values**	**Coeff.**	**Values**
a_1	0.00912	b_1	0.00000161
a_2	0.07398	b_2	0.00000323
a_3	0.36610	b_3	0.003559
a_4	1.07610	b_4	0.003558
a_5	0.77486		

(1). Loop coefficients are synthesized from elliptic functions with f_s=2.1MHz, f_b=20kHz, bandwidth, and SNR wanted.

(2). Max/Min coefficient ratio is greater than 100,000, which leads to difficulty in circuit implementation.



Stability Analysis of Sodini's Structure

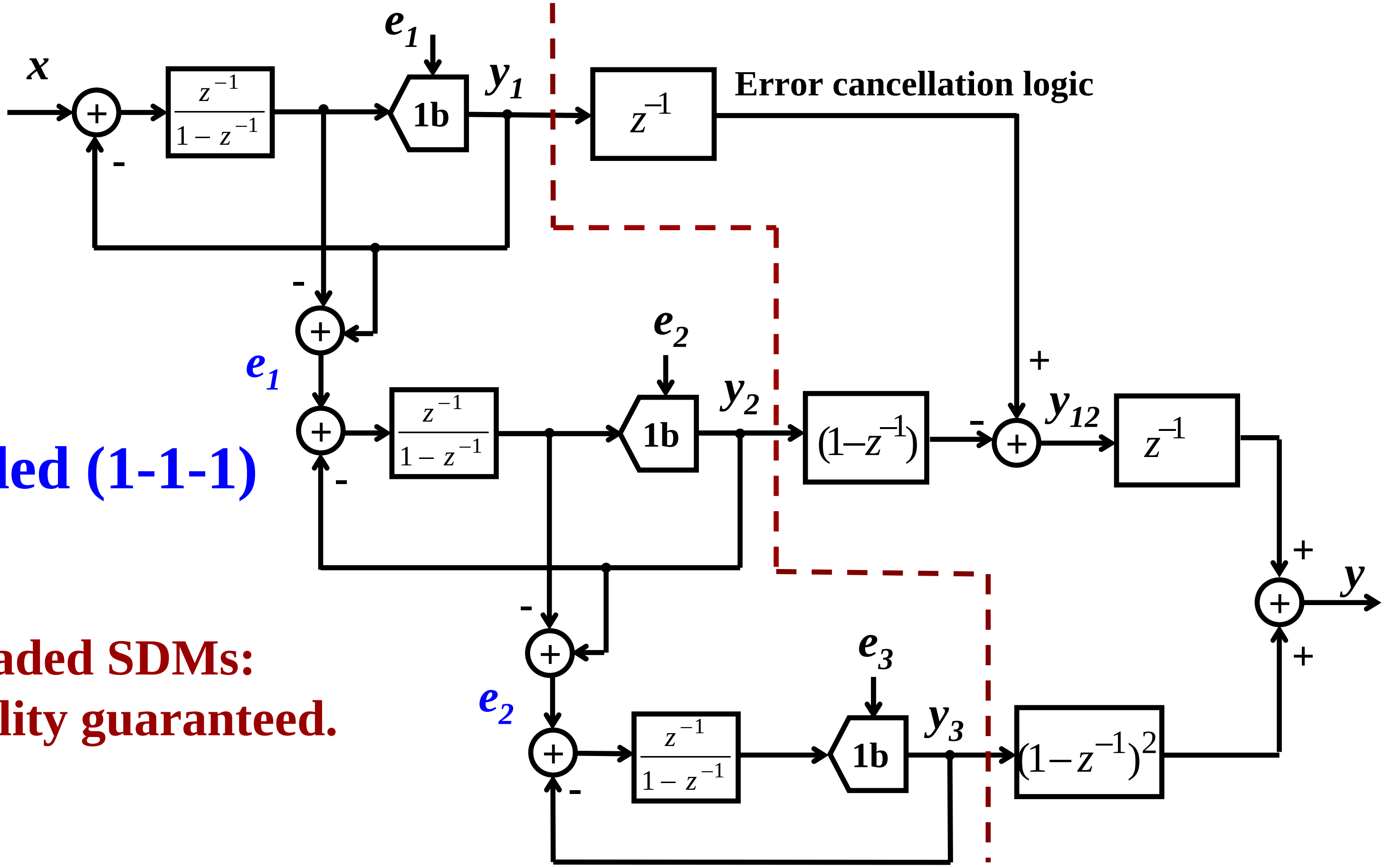


⇒ Conditionally Stable

⇒ Stability protection mechanism is needed. It includes overload detector and integrator reset circuits.

Cascaded ΣΔ Modulators

MASH: **M**ulti-St**A**ge Noise **SH**aping *[IEEE, 1992]*



Cascaded (1-1-1)

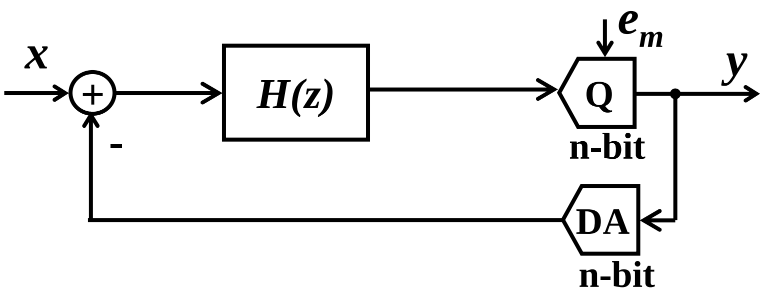
Cascaded SDMs:
Stability guaranteed.

$$Y(z) = z^{-3} \cdot X(z) + \underline{(1 - z^{-1})^3} \cdot E_3(z)$$

(e1,e2 are cancelled out.)

Multibit ΣΔ Modulators

Advantages:



1. Increases modulator's SNR by 6dB per extra bit.
2. Reduces low frequency oscillation associated with the single-bit ΣΔ modulators.
3. Reduces the effect of jitters with single-bit modulation.
4. Helps to stabilize the high-order ΣΔ Systems.

Disadvantages:

The **nonlinearity (NL) problem** caused by component mismatch in a multibit DAC. Errors caused by DAC NL **remain unshaped** at the modulator output, deteriorating modulator performance.

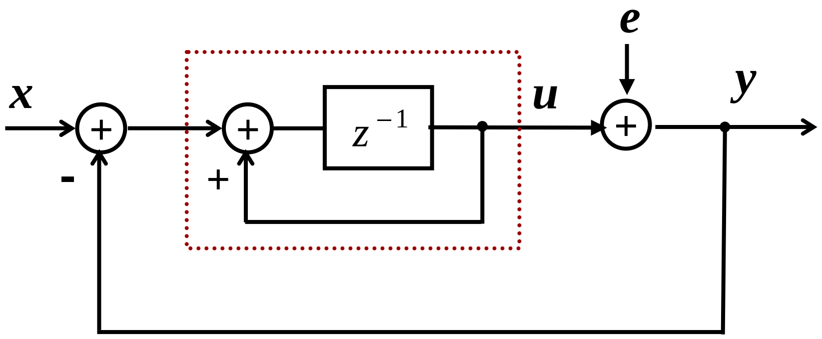
Comparison Table for Modulator Design

[IEEE, 1996]

Modulator type	Advantages	Disadvantages
Low-order single-bit (1^{st} & 2^{nd})	. Guaranteed stability . Simple loop filter design . Simple circuit design . Wider input range	. Low SNR . More prone to having idle tones (needs dithering)
High-order single-bit (Filter Synth.)	. High SNR for modest OSR . Less prone to having idle tones . Simple circuit design	. Stability is signal dependent . Difficult loop filter design . Maximum Input range is limited . Overload and scaling problem . coefficient implementation problem
Cascaded (MASH)	. High SNR for modest OSR . Stability guaranteed . Wider input range	. Request near-perfect matching between analog integrator and digital differentiator . Imperfert matching may cause leakage of tones into baseband
Multibit	. High SNR for fairly low OSR . Helps to stabilize high-order loop . Reduces idle tones	. DAC nonlinearity problem. . More complex of the followed digital filters . Large area, complex circuit design

DC Input Behavioral of the ΣΔ Modulator

First-Order Modulator



$$u[n] = x[n-1] - y[n-1] + u[n-1]$$

$$\begin{cases} if\ u[n] \geq 0, y[n] = 1 \\ if\ u[n] < 0, y[n] = -1 \end{cases}$$

$$e[n] = y[n] - u[n]$$

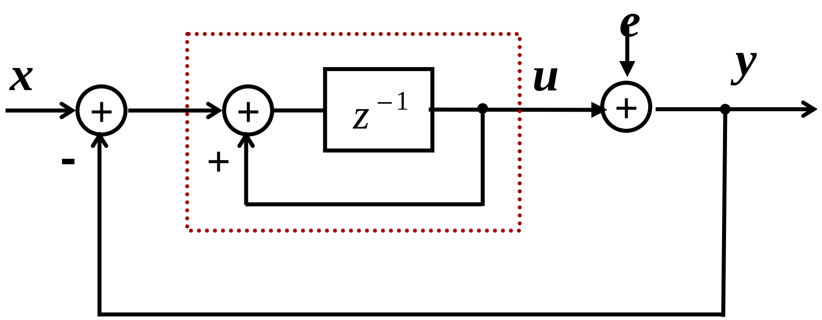
n	x[n]	u[n]	y[n]	e[n]	u[n+1]
0	1/3	0.1	+1	0.9	-0.5667
1	1/3	-0.5667	-1	-0.4333	0.7667
2	1/3	0.7667	+1	0.2333	0.1
3	1/3	0.1	+1	0.9	-0.5667
4	1/3	-0.5667	-1	-0.4333	0.7667
5	1/3	0.7667	+1	0.2333	0.1
6	……	……	……	……	……

⇒ Avergae y(n)=1/3

⇒ Output pattern is periodic, which implies that quantization noise is not random.

DC Input Behavioral of the ΣΔ Modulator

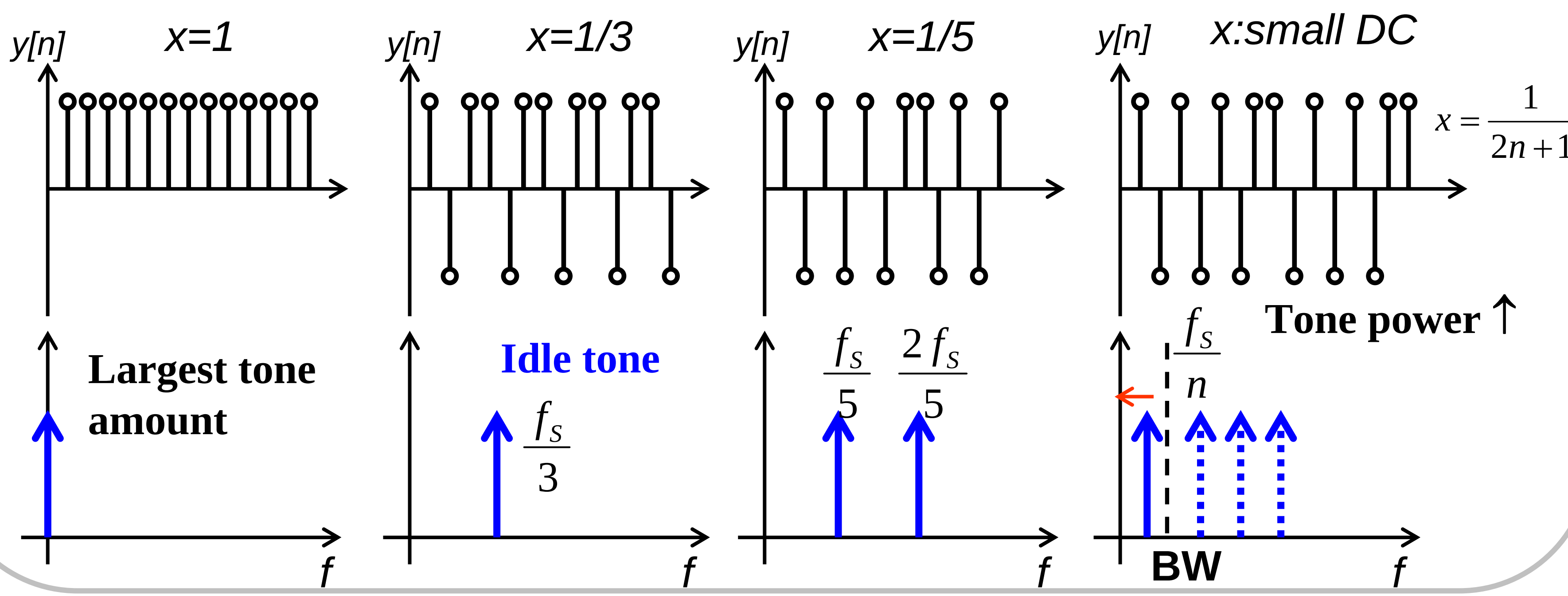
Periodic Output Pattern Causes Idle Tones – (I)



$$u[n] = x[n-1] - y[n-1] + u[n-1]$$

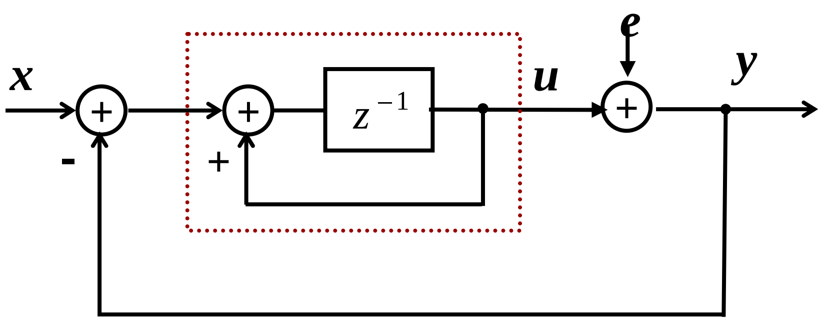
$$\begin{cases} if\ u[n] \geq 0, y[n] = 1 \\ if\ u[n] < 0, y[n] = -1 \end{cases}$$

$$e[n] = y[n] - u[n]$$



DC Input Behavioral of the ΣΔ Modulator

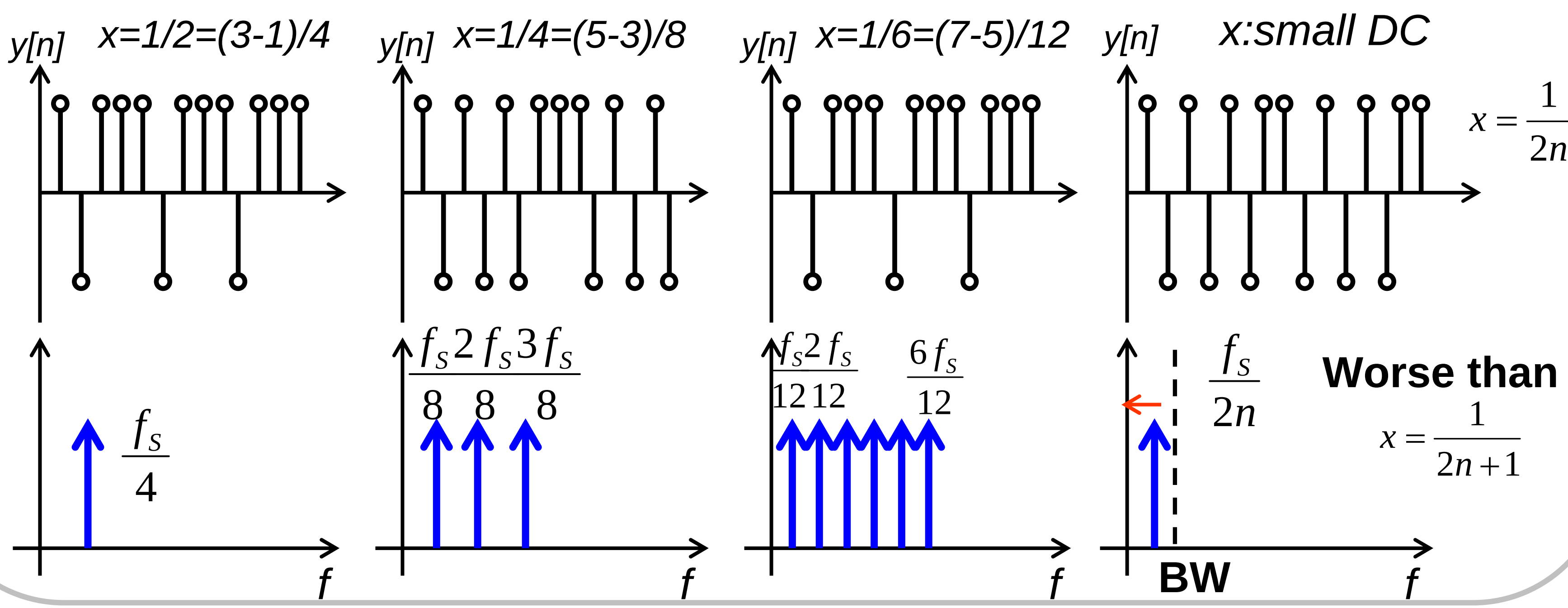
Periodic Output Pattern Causes Idle Tones – (II)



$$u[n] = x[n-1] - y[n-1] + u[n-1]$$

$$\begin{cases} if\ u[n] \geq 0, y[n] = 1 \\ if\ u[n] < 0, y[n] = -1 \end{cases}$$

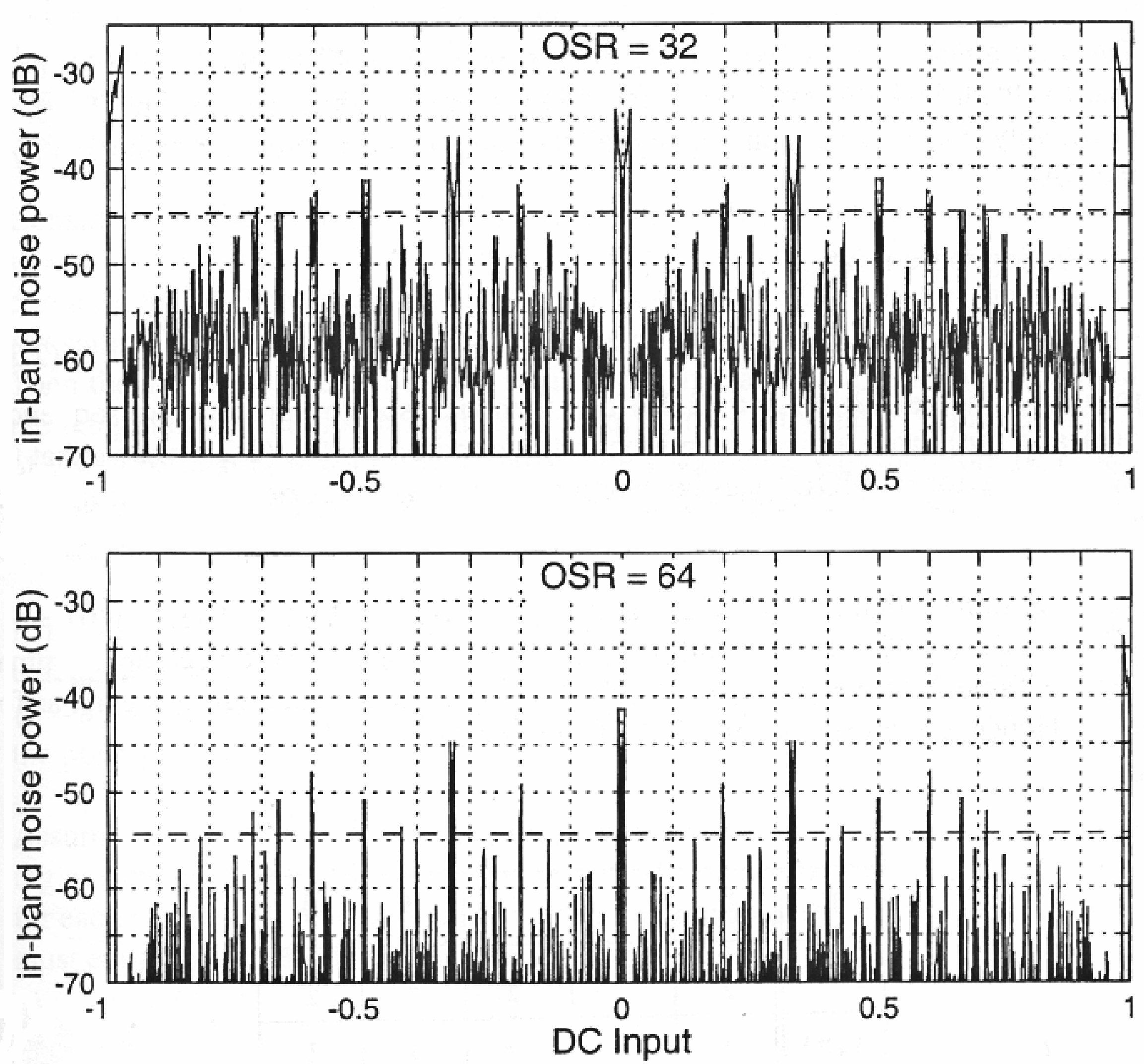
$$e[n] = y[n] - u[n]$$



Inband Quantization Noise Power vs. DC Input Level

For a First-Order Modulator

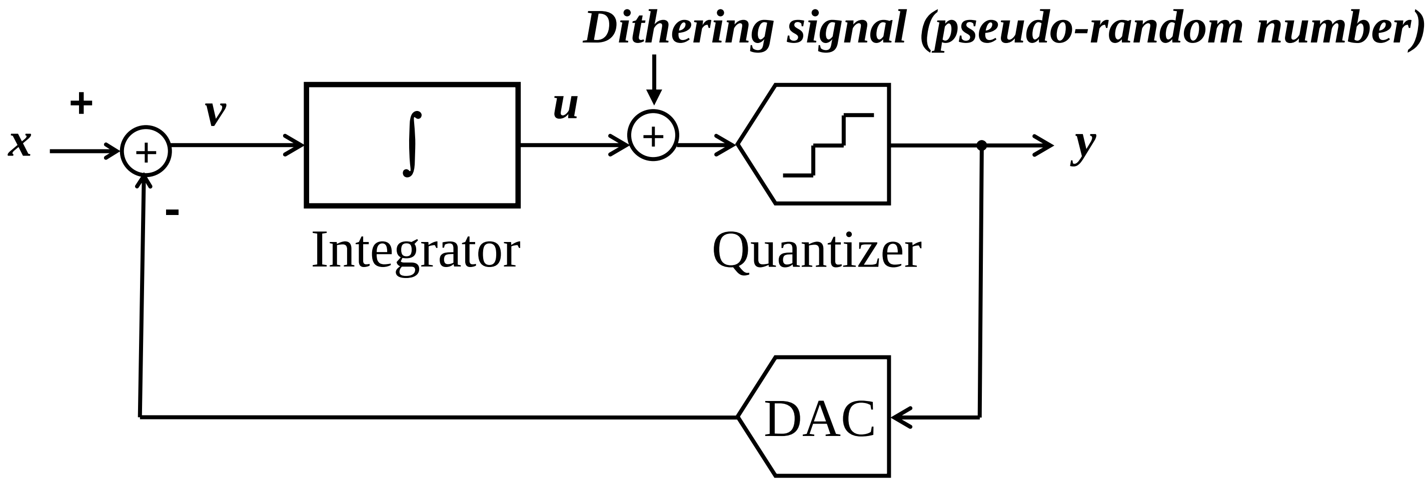
[Schreier, 2005]



- ⇒ **For a smaller dc input, modulator has a longer period, causing the idle tone shifting to the inband.**
- ⇒ **A rational dc input has a large quant. noise power.**
- ⇒ **The idle tone pattern duplicates itself in an endless recursion.**
- ⇒ **High-order ΣΔM has less idle tones.**

Idle Tones Reduction

Dithering



⇒ **Add a small dithering signal before the quantizer to break up the patterned output sequence so that the idle tone will not occur.**

⇒ **Since the dithering signal is placed before the quantizer, it will be spectrally shaped.**

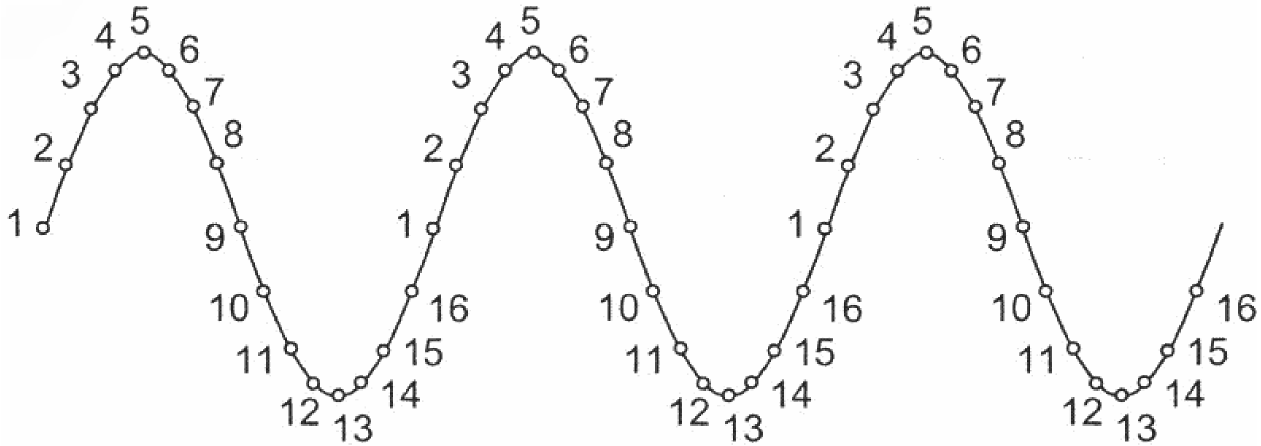
⇒ **No need for high-order modulators.**

Behavioral Simulations Considerations (I)

Coherent Sampling Principle:

– **In the DSP based testing , signals are created using a finite sample set of a few hundred to thousand data points.**

– **In order to repeat signal without loss its information, it is critically important that the last data point flow smoothly into the first data point of the repeating sequence-called coherence.**



One cycle by 16 samples

⇒ **Sampling frequency:** F_s

⇒ **Fundamental (primitive) frequency:** $F_f=F_s/16$

Behavioral Simulations Considerations (II)

Coherent Sampling:

- In general, the fundamental frequency F_f of N samples collected at a sampling rate of F_s is

$$F_f = \frac{F_s}{N}$$ **(frequency resolution)**

- **Unit test period (UTP) (or primitive period):** $UTP = \frac{1}{F_f}$
- **The amount of time required to collect a set of N samples at a rate of F_s is one UTP** $\Rightarrow$ $UTP = \frac{N}{F_s}$
- **The coherent frequencies F_c that can be produced with a repeating sample set are those frequencies that are integer multiples of F_f**

$$F_c = MF_f = M\frac{F_s}{N} \quad (M = 0,1,2,........, N)$$ **(M,N:mutual prime numbers)**

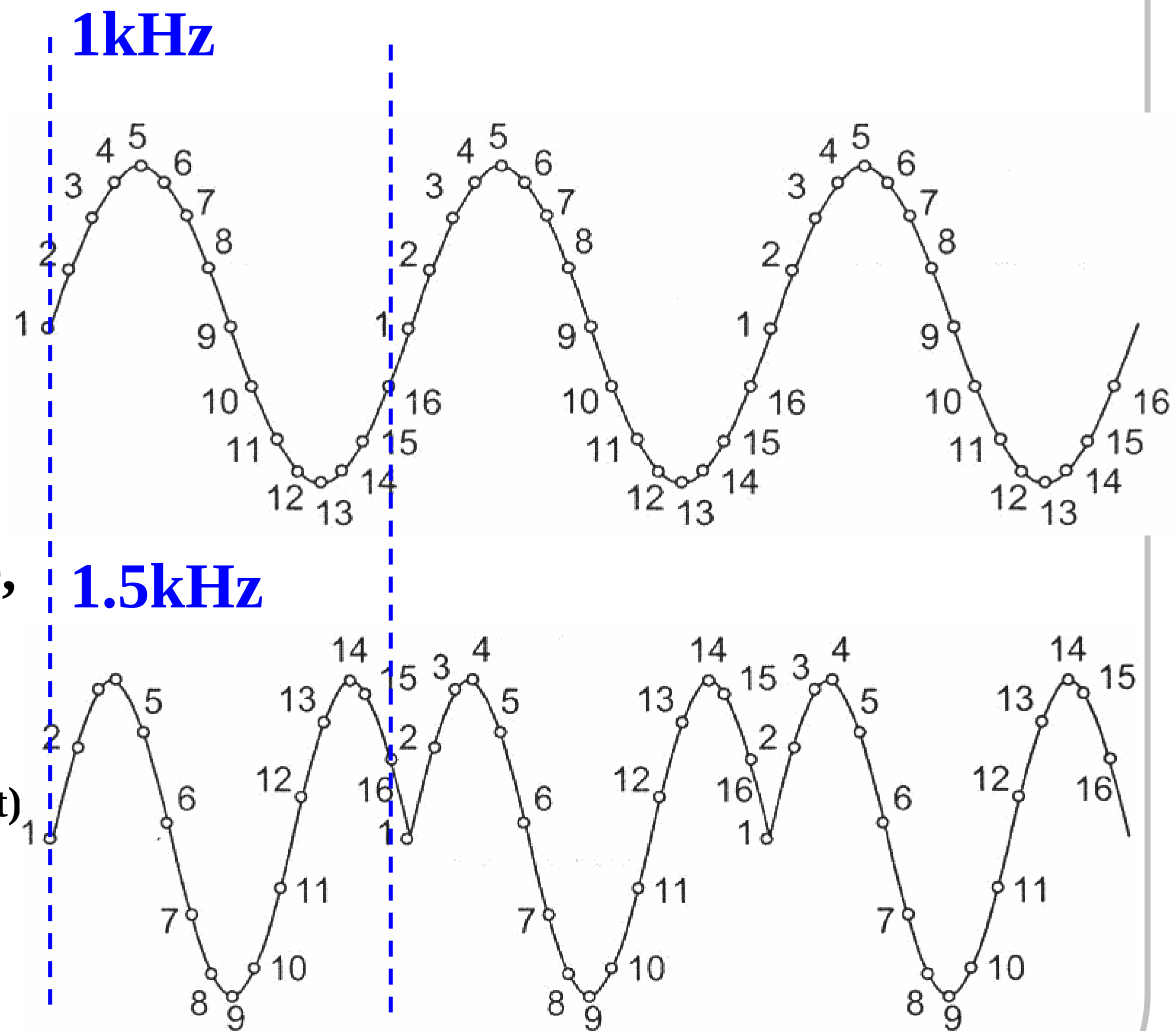
Coherent Sampling Example

- **If F_s =16 kHz**

$$\Rightarrow F_f = \frac{16kHz}{16} = 1kHz$$

- **The next highest frequency for coherent sampling is 2 kHz.**

- **If we wanted to produce a 1.5kHz sinusoidal wave, then we would have a noncoherent sample set.**



Matlab routine (1.5kHz coherent)

```
pi=3.14159265359
for k=1:32;
sinewave(k)=sin(2*pi*3/32*(k-1));
end
```

$$\Rightarrow F_f = \frac{16kHz}{32} = 500Hz$$

Calculations for Coherent Sampling

FFT Pts=8,192

$$f_{in} = M \times \frac{f_s}{N} \Rightarrow f_s = 3.2MHz \Rightarrow N = 8192 \Rightarrow M = 3$$

$$\Rightarrow f_{in} = 3 \times 390.625Hz = \underline{1171.875Hz}$$

(Not exact 1kHz)

$$\Rightarrow \text{Transient step} = \frac{1}{200} \times \frac{1}{f_s} = \underline{1.5625ns}$$

$$\text{Transient time} = 2.56ms + 0.44ms = \underline{3ms}$$

(For HSPICE)

FFT Pts=16,384

$BW = 24kHz$

$OSR = 64$

$f_s = 3.072MHz$

$f_{in} = 1.5kHz$

$pts = 65536$

$$f_{in} = M \times \frac{f_s}{N} \Rightarrow f_s = 3.2MHz \Rightarrow N = 16384 \Rightarrow M = 5$$

$$\Rightarrow f_{in} = 5 \times 195.3125Hz = \underline{976.5625Hz}$$

$$\Rightarrow \text{Transient step} = \frac{1}{200} \times \frac{1}{f_s} = \underline{1.5625ns}$$

$$\text{Transient time} = 5.12ms + 0.88ms = \underline{6ms}$$

Behavioral Simulation Files (1)

MATLAB – sdm1s.m (Part 1)

```
%  *************************************************************
%  * First Order Sigma-Delta Modulator                         *
%  * By Chun-Hsien Su 2006.07.01                               *
%  *************************************************************

          clear;
            clf;
    format long;

% -------- Define Parameters ----------

     tr = 1024*4;          % points for transient
     pt = 1024*16;         % FFT points
     tt = tr+pt;           % total points
    osr = 64;              % osr
     fc = 24000;           % signal bandwidth
     fs = 3200000;         % sampling frequency
    fin = 976.5625;        % input frequency

    wsf = sqrt(sum((min4win(2^14).^2)/2^14));
                           % windown shap factor
```

Behavioral Simulation Files (2)

MATLAB – sdm1s.m (Part 2)

```
% -------- Input Signals -------

     t = 0:1/fs:tt/fs;
    tn = t';

inputdb = [-3];
%inputdb = [ 0 -3 -6 -10 -20 -30 -40 -50 -60 -70 -80 -90 -100];
%For input vs. SNR

for k = 1:1:length(inputdb)

  amp = 10^(inputdb(k)/20);

    x = amp*sin(2*pi*fin*t);
   xn = x';

% -------- By Simulink --------

  sim('sdm1a',[0 tt/fs]);
```

Behavioral Simulation Files (3)

MATLAB – sdm1s.m (Part 3)

```
% -------- Calculating FFT of Y[n] --------

   Y = fft(y(tr+1:tt));
 % Y = fft(y(tr+1:tt).*min4win(pt)/wsf); % With Window Function
   L = length(Y);
   f = fs*(0:L-1)/(L);
  Py = Y .*conj(Y)/((L/2)^2);

  figure(1);
  semilogx(f(1:L/2),10*log10(Py(1:L/2)),'b');
  xlabel('Hz');
  ylabel('dB');
   title('PSD of the First-order modulator');
   grid;
   axis([10^1 10^6 -140 0]);
```

Behavioral Simulation Files (4)

MATLAB – sdm1s.m (Part 4)

```
% -------- Calculating  Dynamic Range --------

    ws = (fin*L/fs)+1;
    wn = (fc*L/fs)+1;
    signal = sum(Py(ws-3:ws+3));
    noise  = sum(Py(5:wn))-signal;
    snr = 10*log10(signal/noise);
    fprintf('InputdB= %g.\n',inputdb(k));
    fprintf('Max u %g.\n',max(u(tr+1/4:L)));
    fprintf('The SNR is %g.\n',snr);

  uu(k)=max(u(tr+1:L));
  sn(k)=snr;

  snrtxt=sprintf('SNR=%3.2fdB',snr);
  text(12,-10,snrtxt);

 end;
```

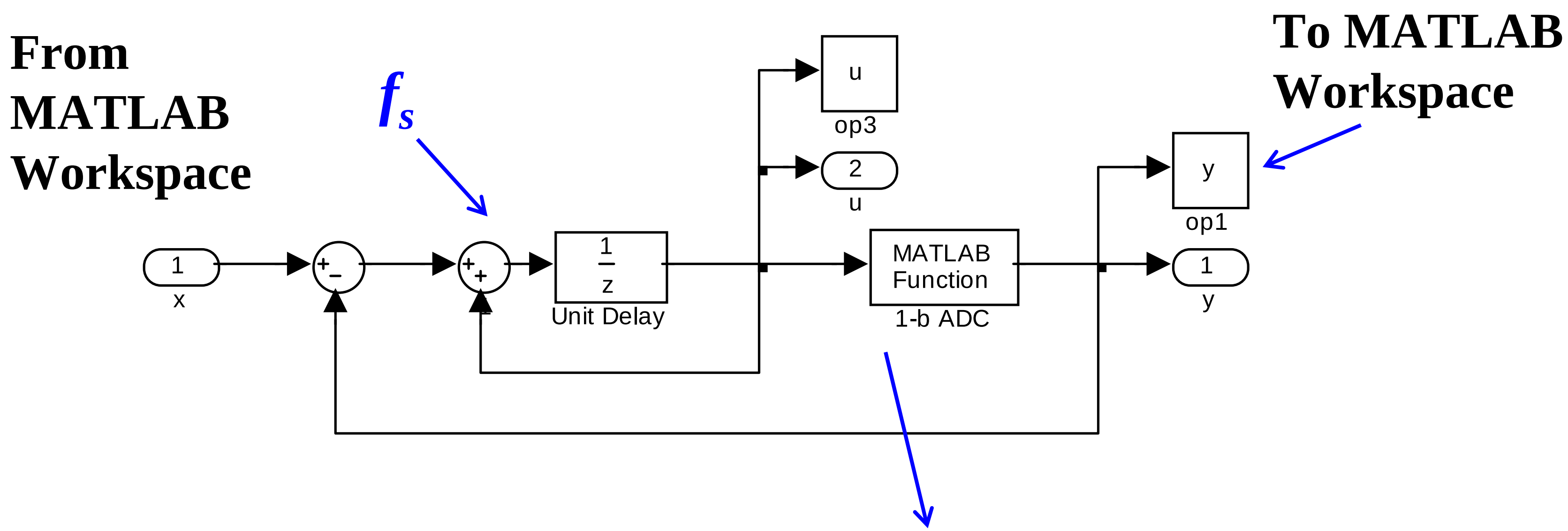
Behavioral Simulation Files (5)

MATLAB – sdm1s.m (Part 5)

```
% -------- Plot Figures --------

   figure(2)
   plot(inputdb,uu,'b');
   grid;
   xlabel('Input Levels in dB');
   ylabel('V');
   title('Integrator''s Outputs');

   figure(3)
   plot(inputdb,sn,'b');
   grid;
   xlabel('Input Levels in dB');
   ylabel('dB');
   title('Input level vs SNR');
```

Behavioral Simulation Files (6)

Simulink – sdm1a.m



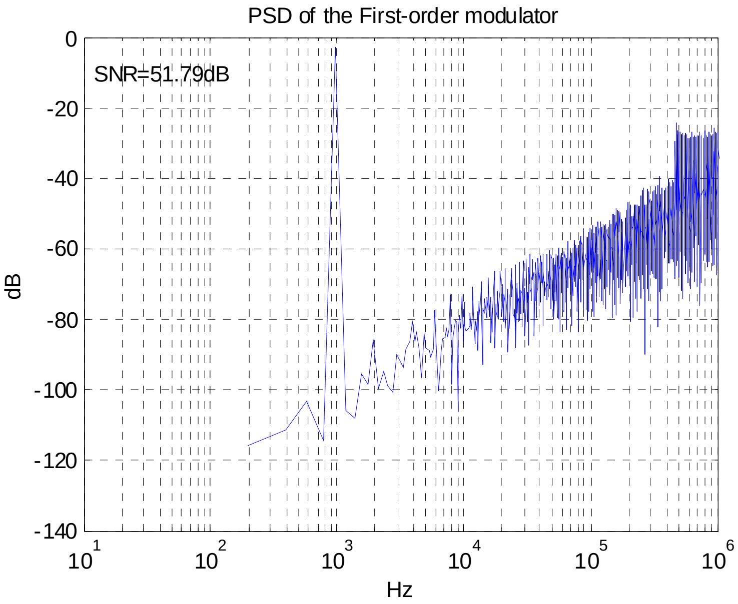
Min4win function

```
function w = min4win(n);

w = blackmanharris(n);
```

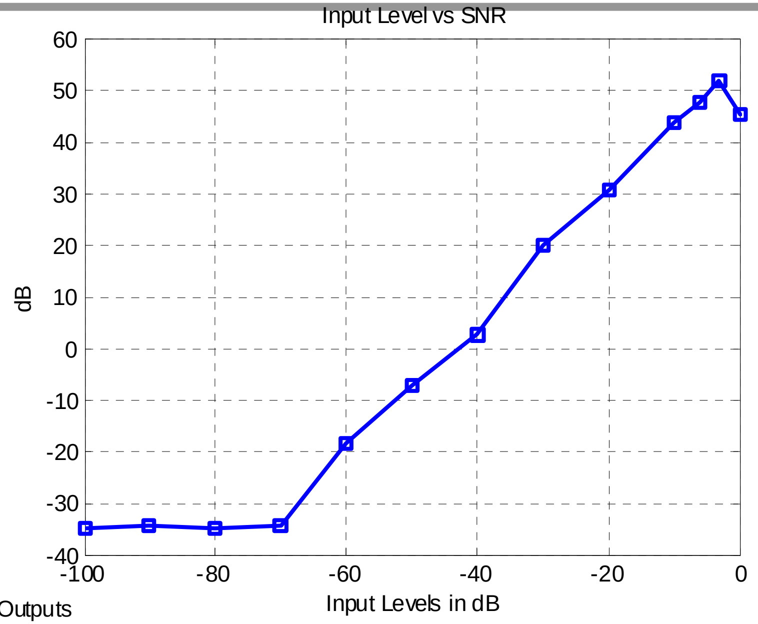
1-b ADC function

```
function dout = adc21(ain);

if ain < -0
   dout = -1;
else dout = 1;
   end;
```

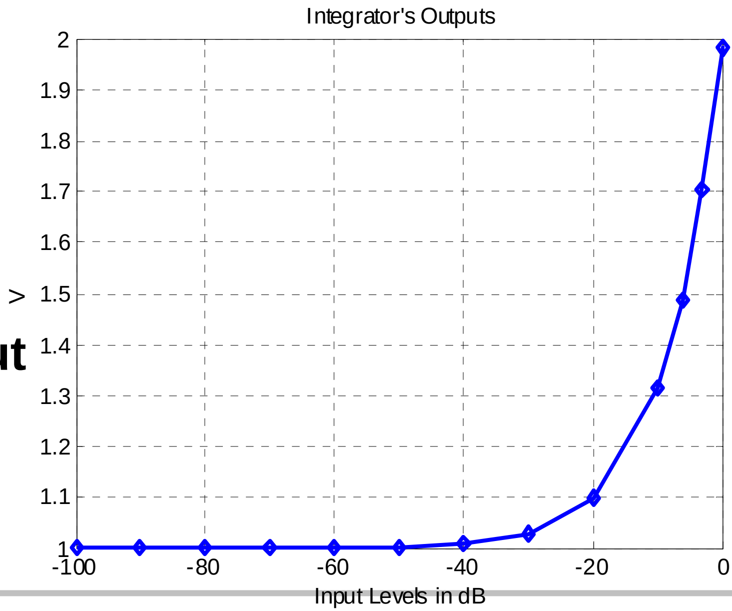
Simulation Results



FFT of y



Input vs SNR



Integrator Output

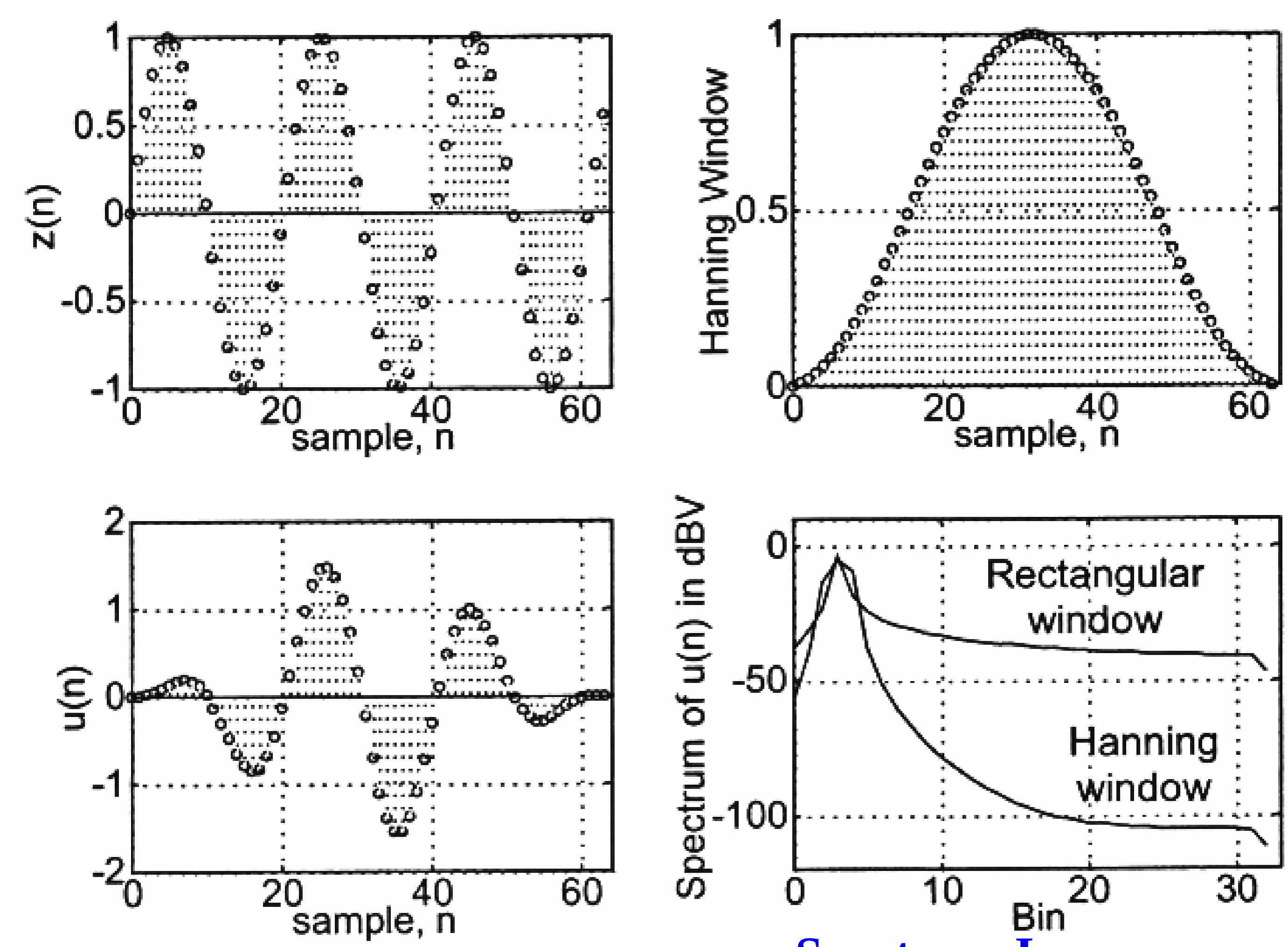
Windowing Functions (I)

Prevent Spectrum Leakage

Hanning window

Original noncoherent waveform

Windowed date



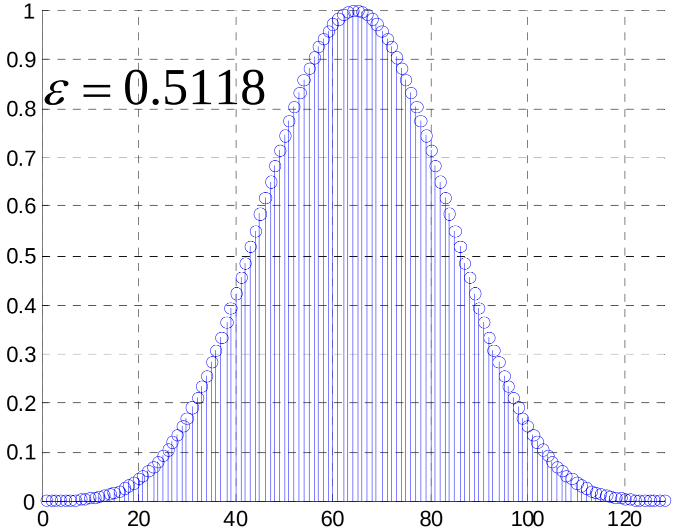
Spectrum Improvement

Windowing Functions (II)

Window Shape Factor

```
% noncoherent signal definition - z -
NOI=64;
N=64; M=pi; A=1; P=0;
        for n=1:NOI,
                z(n)=A*sin(2*pi*M/N*(n-1)+P);
        end;
% windowing operation
        w= hanning(NOI)';
        epsilon=sqrt(sum(w.*w)/NOI);
        u= 1/epsilon * z .* w;
```

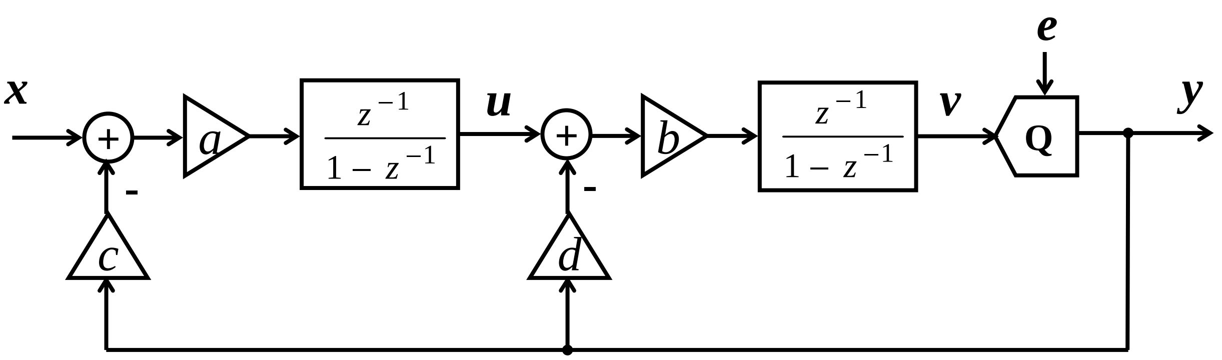
Min4win



⇒ **The windowed data are scaled by the window shape factor denoted by epsilon, ε (=0.612 for N=128)**

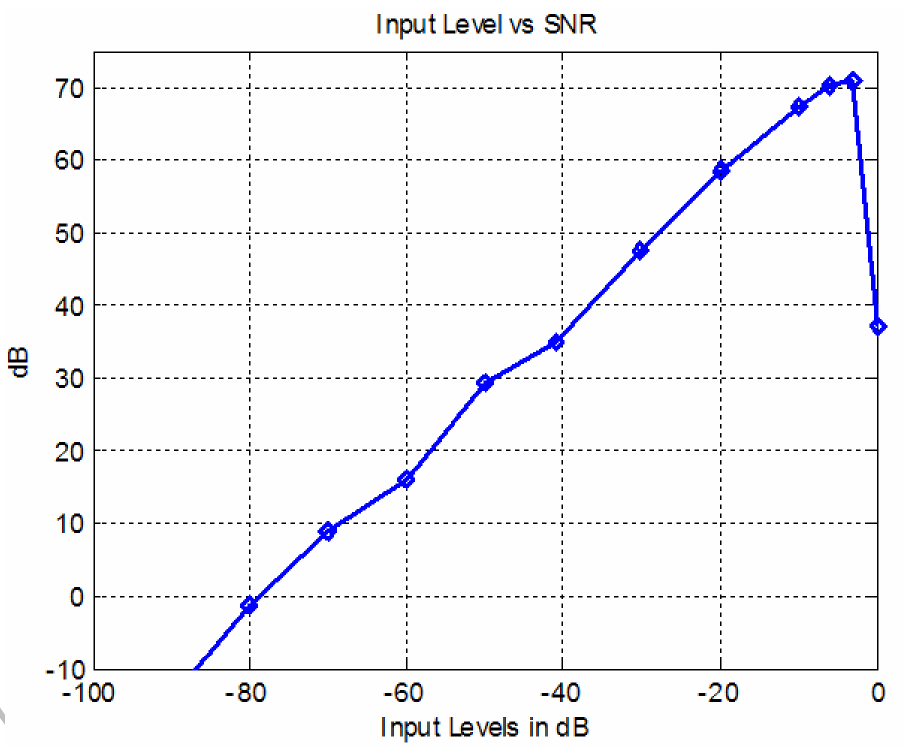
$$\varepsilon = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} w^2(n)}$$

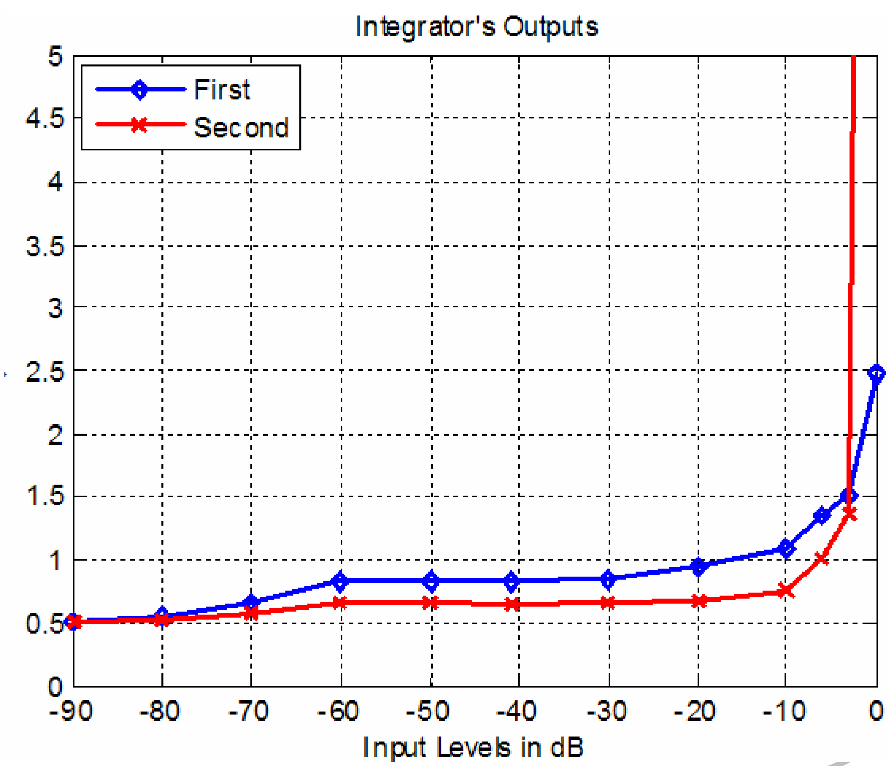
Second-Order ΣΔ Modulator



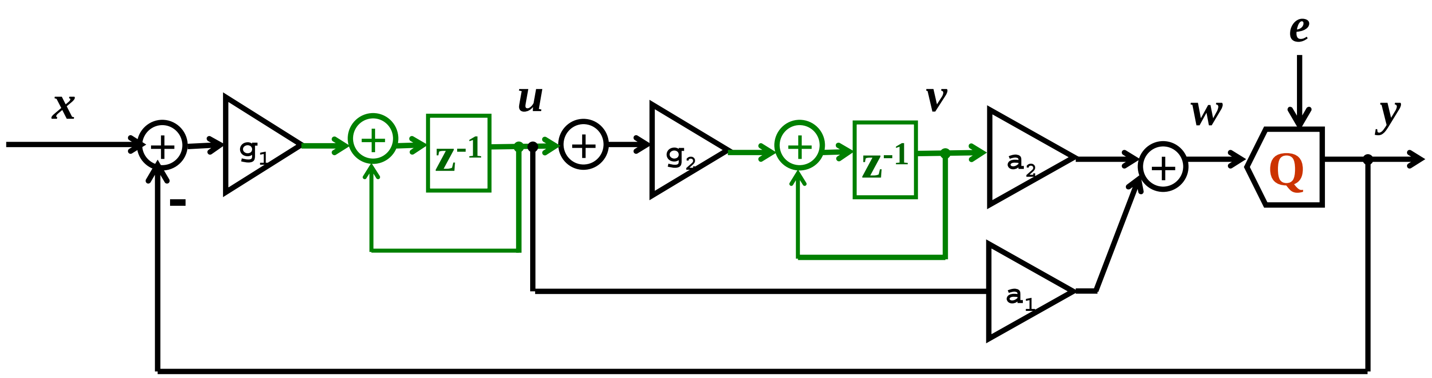
Candy's Structure
a=b=1/2, c=d=1

a, *b*, *c*, and *d* can be optimized.





Feedforward Second-Order Structure



$g_1 = 1/4$ $\quad g_2 = 1/6$ $\quad a_1 = 2$ $\quad a_2 = 3$ $\quad Q:$ 1bit/M-bit

⇒ **NTF can be arranged to being equivalent to multiple feedback structure.**

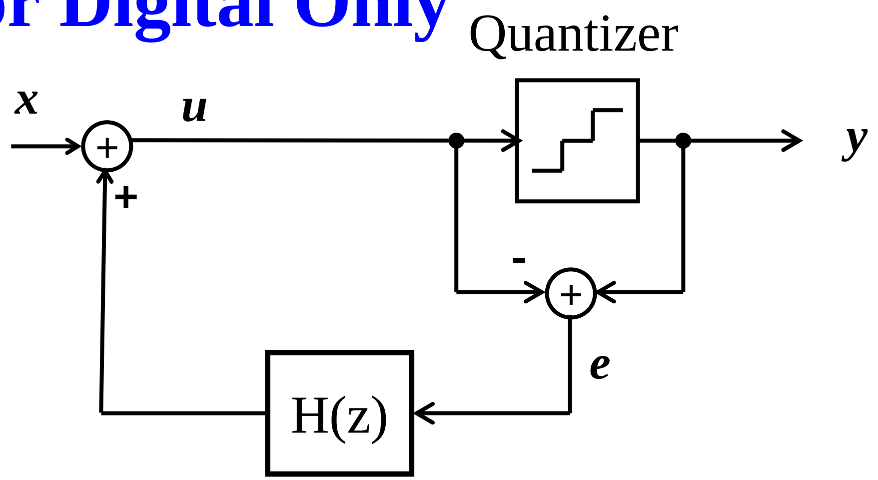
⇒ **Only one feedback path.**

⇒ **Different integrator output ranges.**

Error-Feedback Structure

[Anastassiou, 1989]

For Digital Only

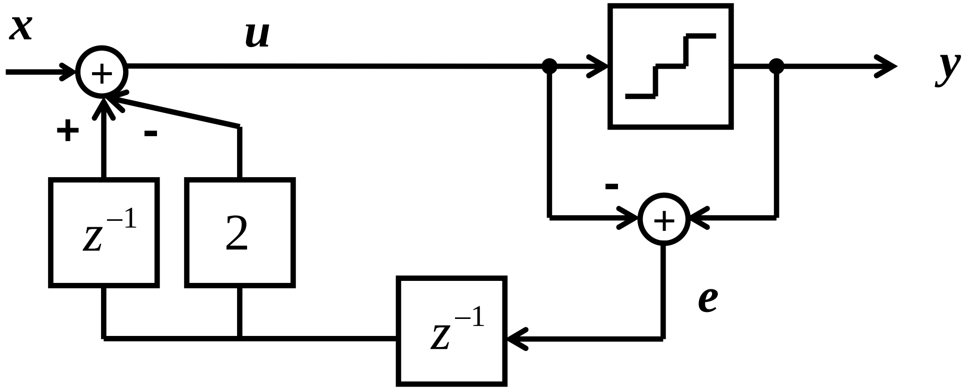


$$\begin{cases} e = y - u \\ u = x + H \cdot e \end{cases}$$

$$\Rightarrow \quad y = x + (1 + H) \cdot e$$

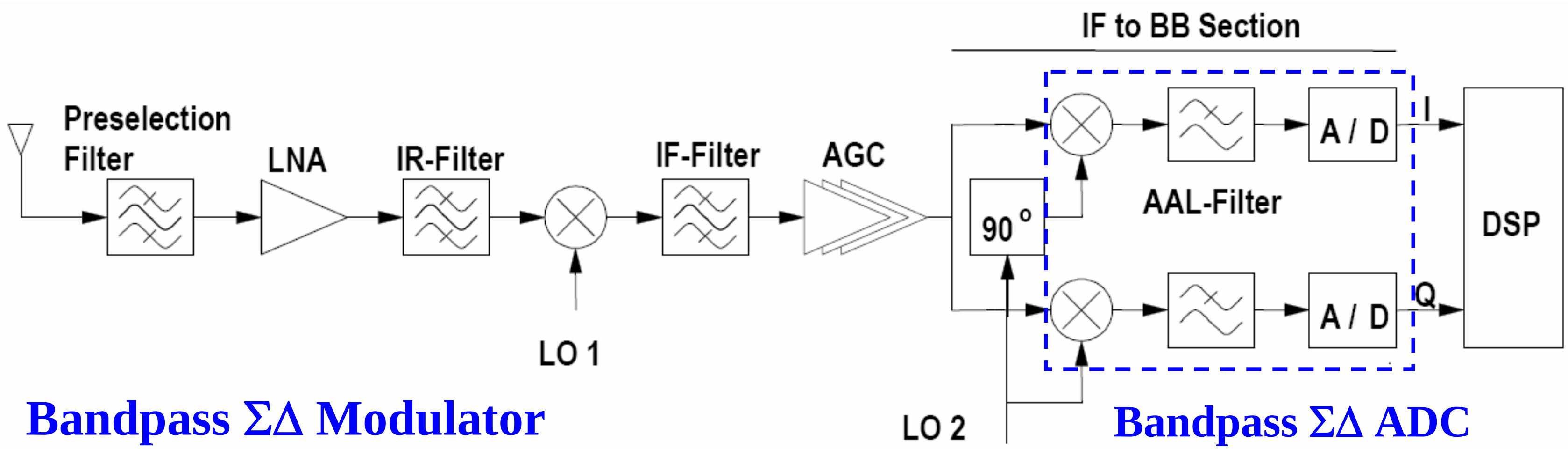
For a second-order function: $y = x + (1 - z^{-1})^2 \cdot e \;\Rightarrow\; 1 + H = (1 - z^{-1})^2$

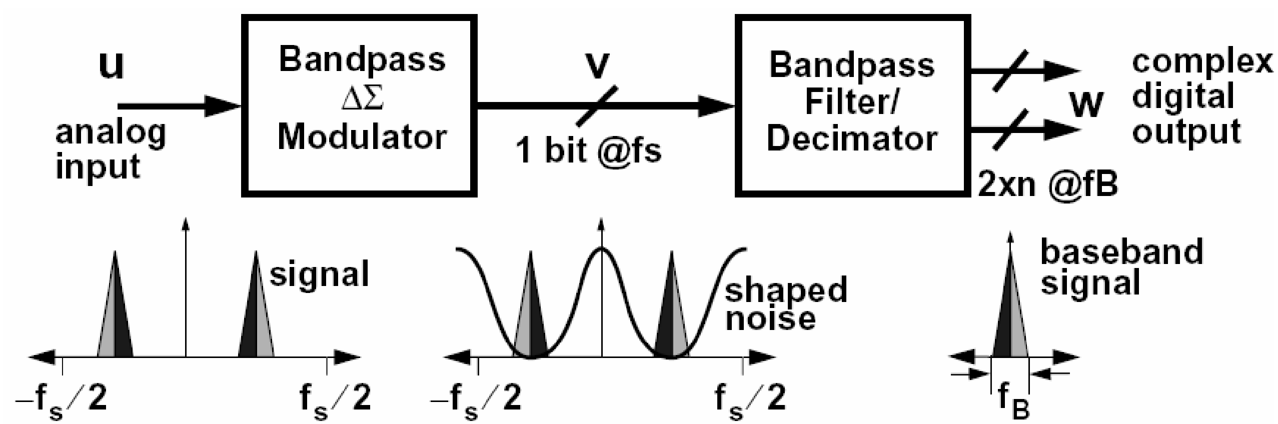
$$\Rightarrow \quad H(z) = (1 - z^{-1})^2 - 1 = -2z^{-1} + z^{-2} = z^{-1} \cdot (-2 + z^{-1})$$



Bandpass ΣΔ ADC

Like a lowpass ΣΔ ADC, a bandpass ΣΔ ADC converts its analog input into a bit-stream output. A digital filter removes out-of-band noise and mixes the signal to baseband.





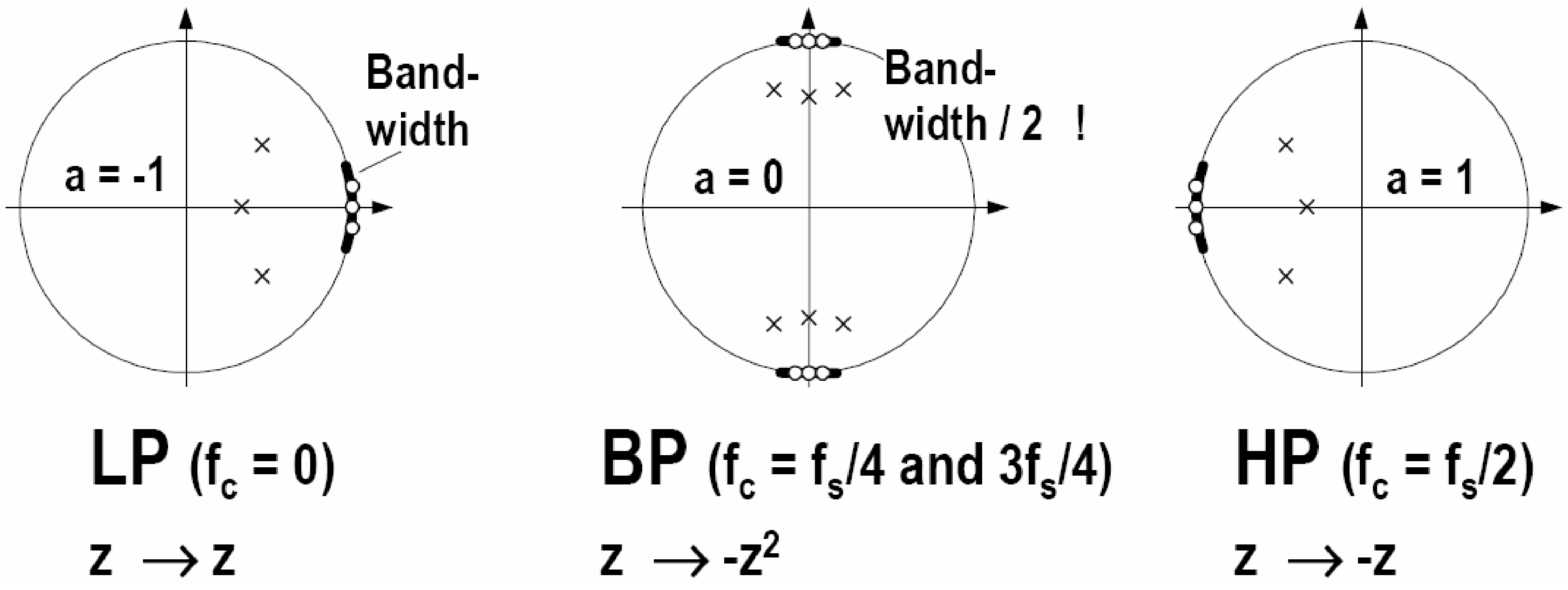
Lowpass to Bandpass Transform *[T. Burger, ETH Zurich]*

General Transform: $z \rightarrow -z\dfrac{z+a}{az+1}, -1 < a < 1$

$\Rightarrow$ **a=0 "Pseudo 2-path transformation"**

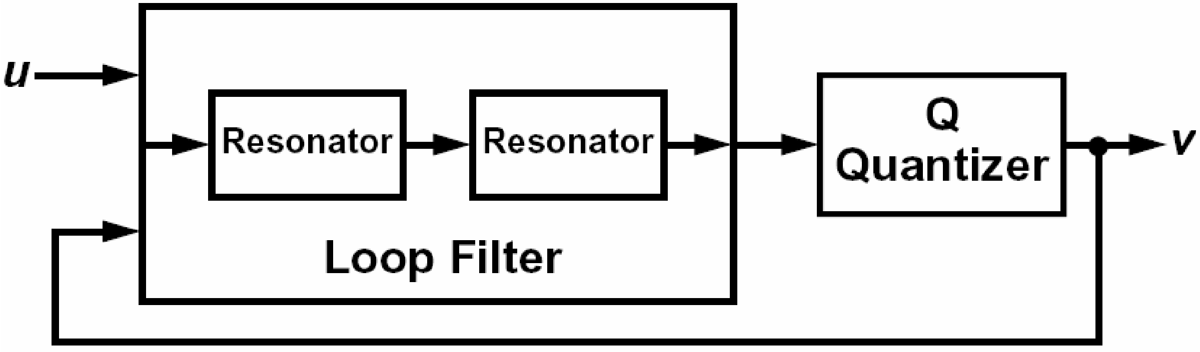
$\Rightarrow$ **$H(z) = 1-z^{-1} \rightarrow H'(z) = 1+z^{-2}$**

Special Cases:

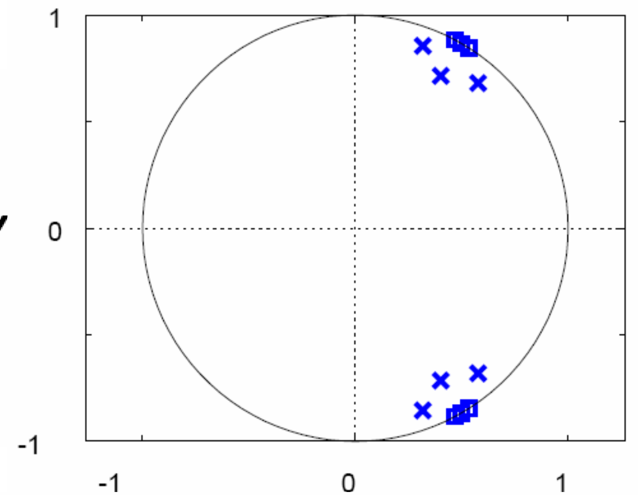


Bandpass ΣΔ Modulator Sturcutre

Bandpass ΣΔ Modulator :



Pole/Zero Diagram:

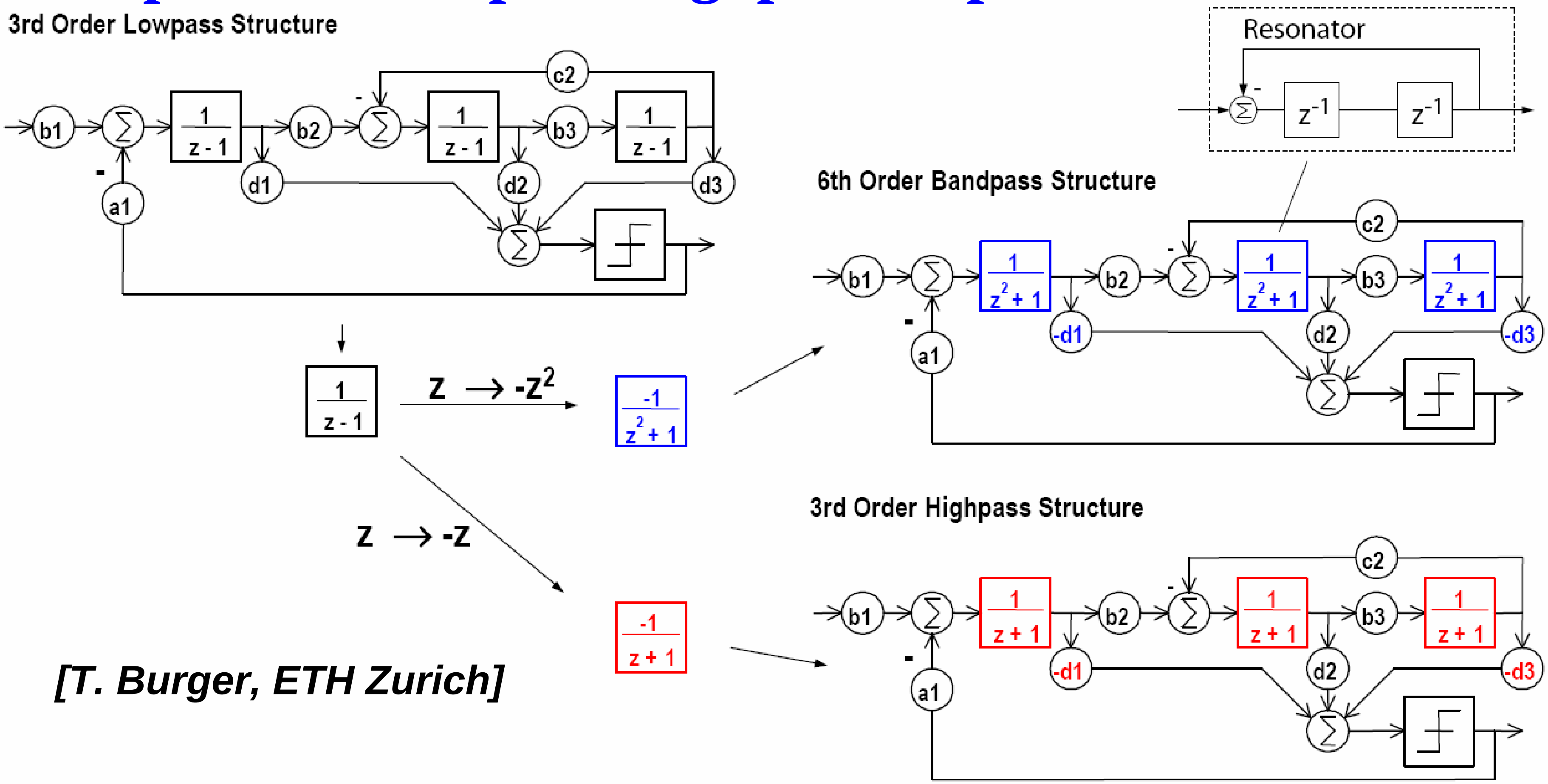



```
OSR = 64;
f0 = 1/6;
H=synthesizeNTF(6,OSR,1,[],f0);...
```

- The loop filter consists of a cascade of resonators.
- The resonance frequencies determine the poles of the loop filter and hence the zeros of the NTF.
- Multibit and multistage variants are also possible

Modulator Architecture Implementation

Lowpass to Bandpass/Highpass Implementation



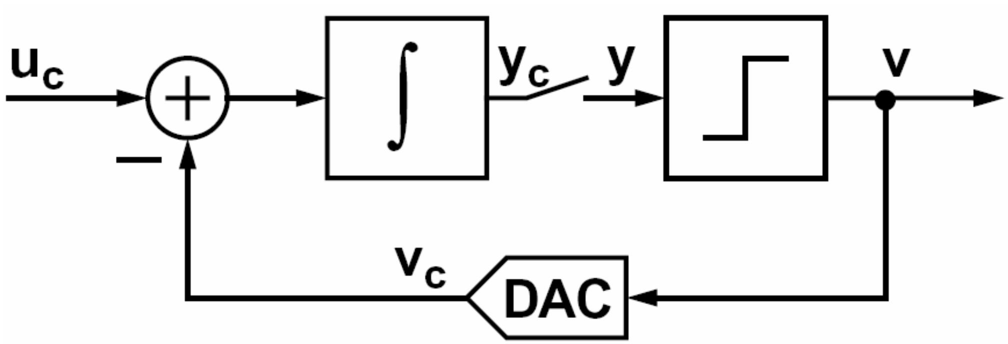
[T. Burger, ETH Zurich]

– **Integrators are replaced by resonators.**

– **Resonators are more difficult to build than integrators .**

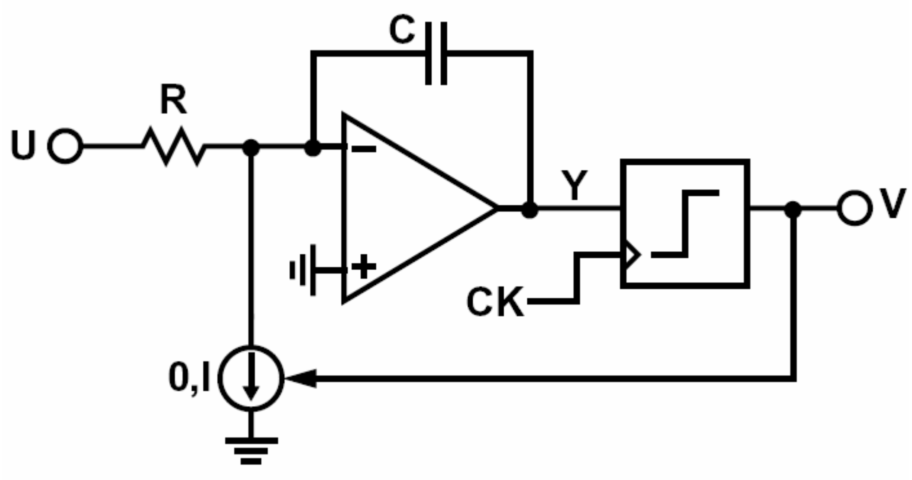
Continuous-Time ΣΔ Modulator *[Schreier, ADI]*

First-Order Model :



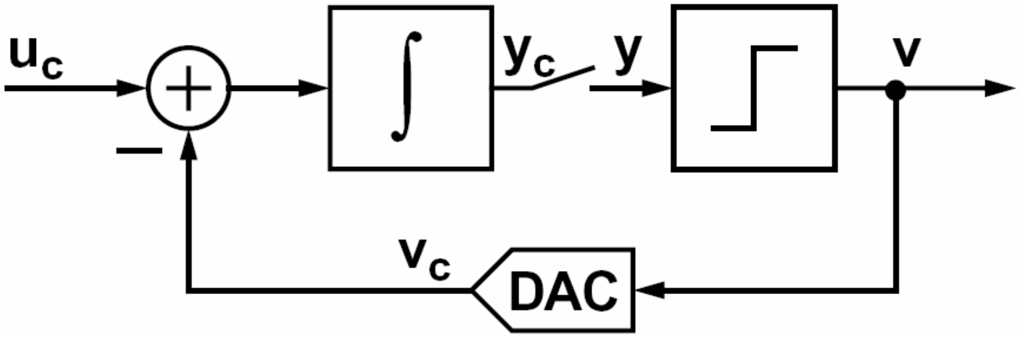
Assume comparator and DAC are delay-free.

Circuit Implementation :



Normalize R=1Ω, C=1F, I=1A, F_s=1Hz
Full-scale range is [0,1]V.

Analysis of First-Order CT ΣΔM



From the diagram:

$$y_c[n] = y_c[n-1] + \int_{n-1}^{n} \big(u_c(\tau) - v_c(\tau)\big) \cdot d\tau$$

1) Sample y_c at integer time and identify $y[n] = y_c[n]$.

2) Observe that $\int_{n-1}^{n} v_c(\tau) \cdot d\tau = v[n-1]$

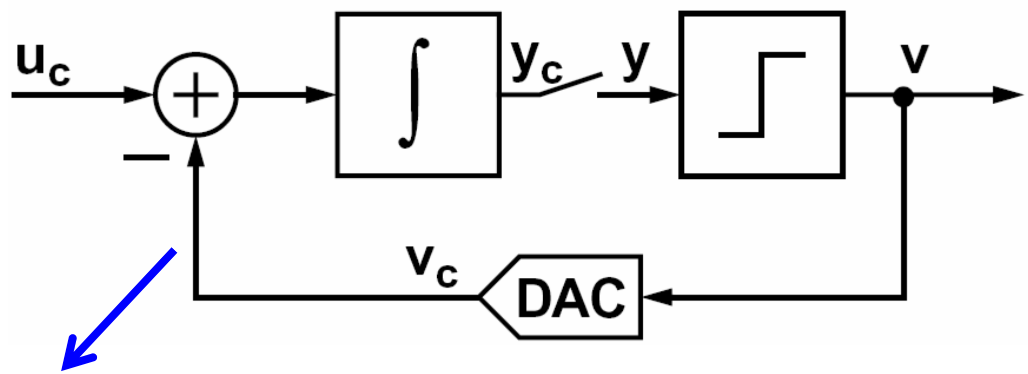
3) Define $u[n] = \int_{n-1}^{n} u_c(\tau) \cdot d\tau$

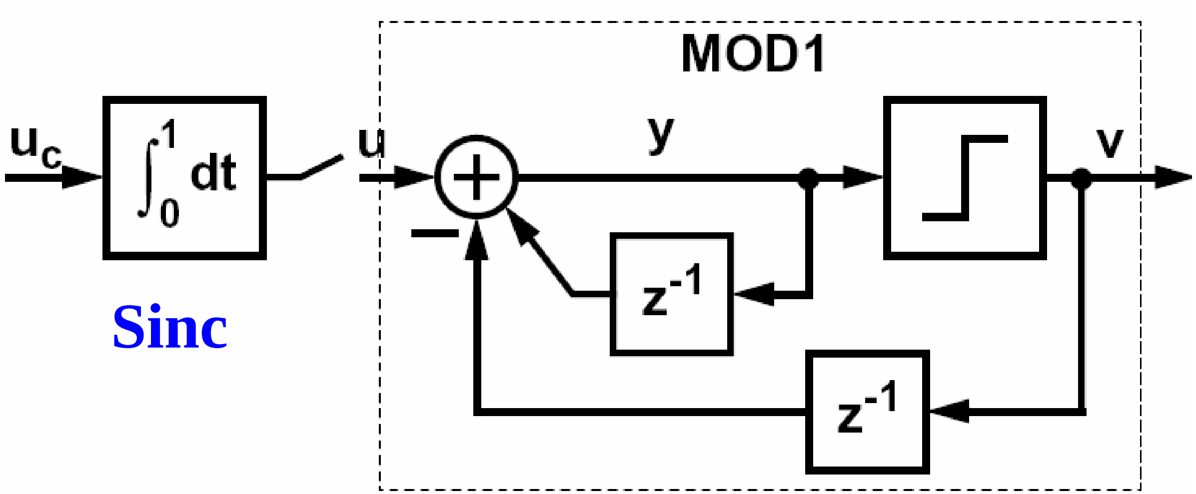
Then $y[n] = y[n-1] + u[n] - v[n-1]$

Also, from the diagram $v[n] = Q\big(y[n]\big)$

CTMOD Equivalency

First-Order Model :



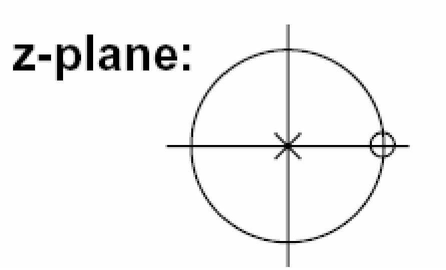


First-order CTMOD is the same as a discrete-time modulator preceded by a sinc filter.

STF and NTF of First-Order CTSDM

The NTF is the same as CTSDM:

$$NTF(z) = 1 - z^{-1}$$

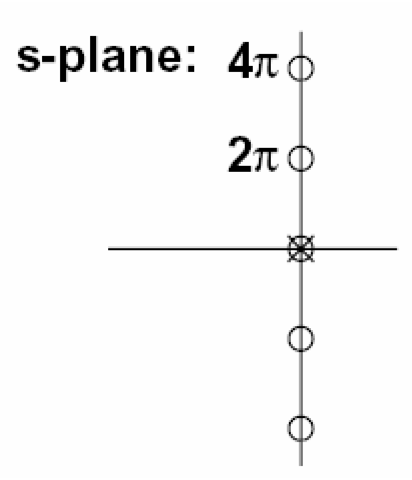


MOD1's STF is 1, (so the overall STF is just the TF of the prefilter):

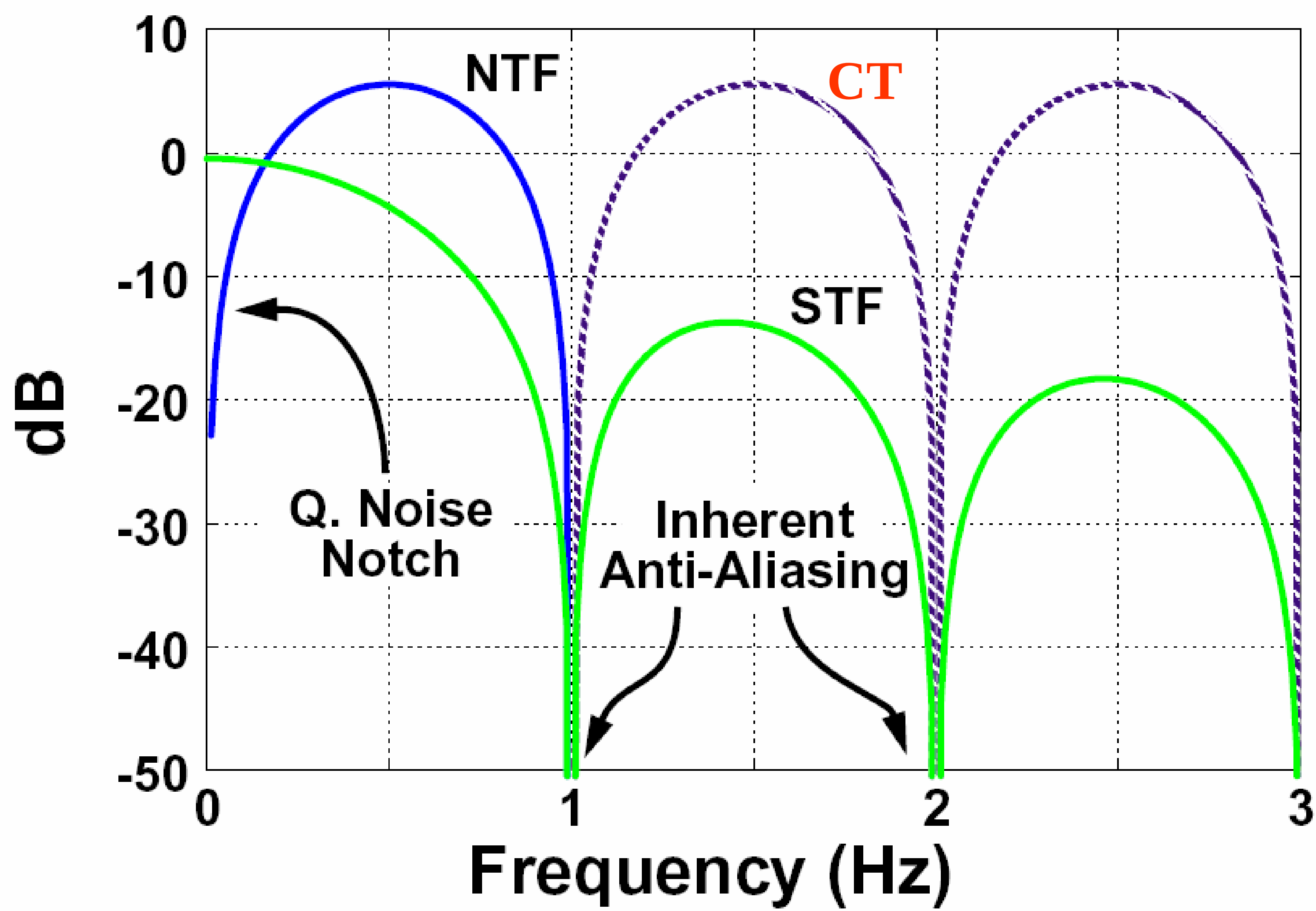
$$STF(z) = \int_0^{\infty} e^{-sT} \cdot 1 \cdot dt$$

$$= \int_0^{1} e^{-sT} \cdot dt = \frac{1 - e^{-s}}{s}$$

$$= \frac{1 - z^{-1}}{s}, \; where\; z^{-1} = e^{-s}$$

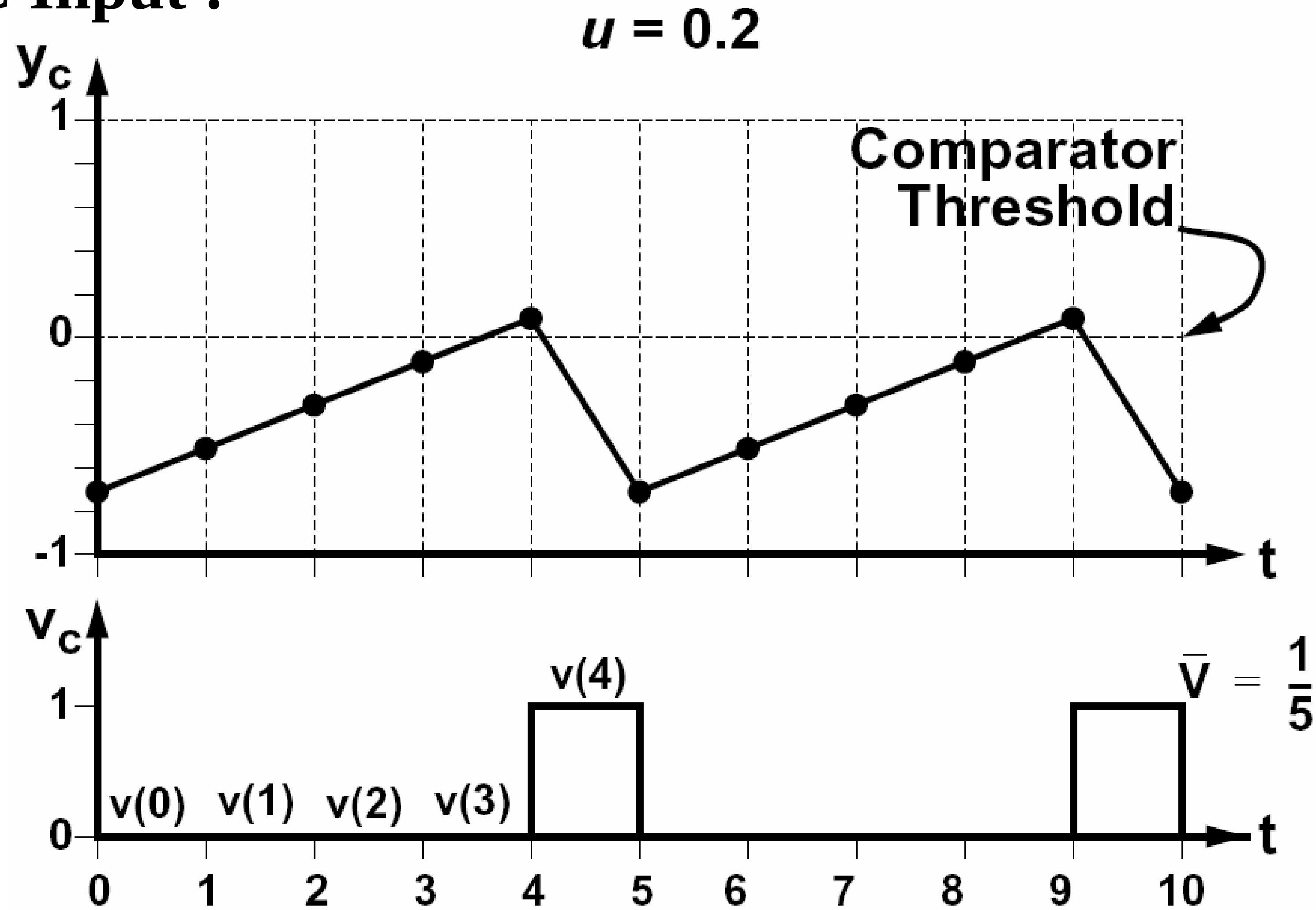


Frequency Response of First-Order CTSDM



Time-Domain Waveform of First-Order CTSDM

DC Input :



Nonidealities in CTSDM (I)

- **Component shifts**

  $\Rightarrow$ **R $\rightarrow$ R+ΔR or I$\rightarrow$ I+ Δ I merely changes full-scale.**

  $\Rightarrow$ **C $\rightarrow$ C+ Δ C scales the output of the integrator, but does not affect the comparator's decisions.**

- **Op-amp offset, input bias current, DAC imbalance**

  $\Rightarrow$ **All translate into a DC offset, which is unimportant in many applications.**

- **Comparator offset & hysteresis**

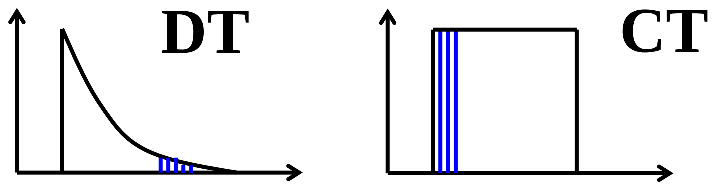
  $\Rightarrow$ **Overcome by integrator.**

- **Finite op-amp gain**

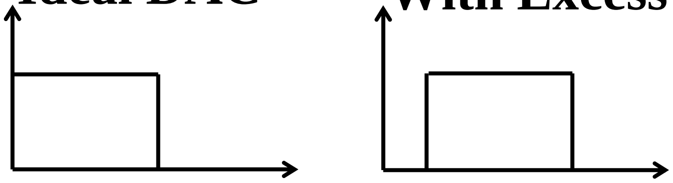
  $\Rightarrow$ **Creates "dead-bands."**

Nonidealities in CTSDM (II)

- **DAC jitter**
  ⇒ **Adds "noise."**



- **Resistor nonlinearity (e.g. due to self-heating)**
  ⇒ **Introduces distortion.**
- **DAC nonlinearity**
  ⇒ **Introduces distortion and inter-modulation of shaped quantization noise.**



- **Capacitor nonlinearity**
  ⇒ **Irrelevant.**
- **Op-amp nonlinearity**
  ⇒ **Same effects as DAC nonlinearity, but less severe.**

References (I)

Sigma-Delta Techniques

[1]. R. Schreier, and G. Temes, Understanding *Delta-Sigma Data Converters*, IEEE Press, New York, NY, 2005.

[2]. S. Norsworthy, R. Schreier, and G. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, New York, NY, 1997.

[3]. J. Candy and G. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, New York, NY, 1992.

[4]. R. Carley, R. Schreier, and G. Temes, “Delta-sigma ADCs with multi-bit internal converters,”: *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, New York, NY, 1997.

[5]. J. Candy, “A use of double integration in sigma-delta modulation,” *IEEE Trans. Commun.,* Vol. COM-33, pp. 249-258, Mar. 1985.

[6]. T. Burger, “Bandpass Sigma-Delta Converters for Communications, ESPRIT - Mixed Signal Design Cluster - Work Shop, Pfäffikon, Switzerland, October, 2001.

[7]. Mark Burns and Gordon W. Roberts, “An Introduction to Mixed-Signal IC Test and Measurement,” Oxford University Press, 2001.

References (II)

Stability Analysis

[1]. M. Aldajani and A. Sayed, “Stability analysis of an adaptive structure for sigma delta modulation,” *Proc. IEEE ISCAS 2000*, vol. 1, pp. 129-132, Dec. 2000.

[2]. S. Ardalan and J. Paulos, “An analysis of nonlinear behavior in delta-sigma modulators,” *IEEE Trans. Circuits Syst*., vol. CAS-34, pp. 593-603, Jun. 1987.

[3]. R. Schreier, “An empirical study of high-order single-bit delta-sigma modulators,” *IEEE Trans. Circuits and Syst. II*, vol. CAS-40, no. 8, pp. 461-466, Aug. 1993.

[4]. R. Baird and T. Fiez, "Stability analysis of high-order delta-sigma modulator for ADC’s," *IEEE Trans. Circuits and Syst. II*, vol. CAS-41, no. 1, pp. 59-62, Jan. 1994.

[5]. S. Hein and A. Zakhor, Sigma Delta Modulators: nonlinear decoding, algorithms, and stability analysis, Chapter 7. *Kluwer Academic Publishers*, Nowell, MA, 1993.

[6]. E. Stikvoort, "Some remarks on the stability and performance of the noise shaper or sigma-delta modulator," *IEEE Trans. Commun.*, vol. COM-36, no. 10, pp. 1157-1162, Oct. 1988.