CMPE 352
Signal Processing & Algorithms

Spring 2019

Sedat Ölçer

April 22, 2019

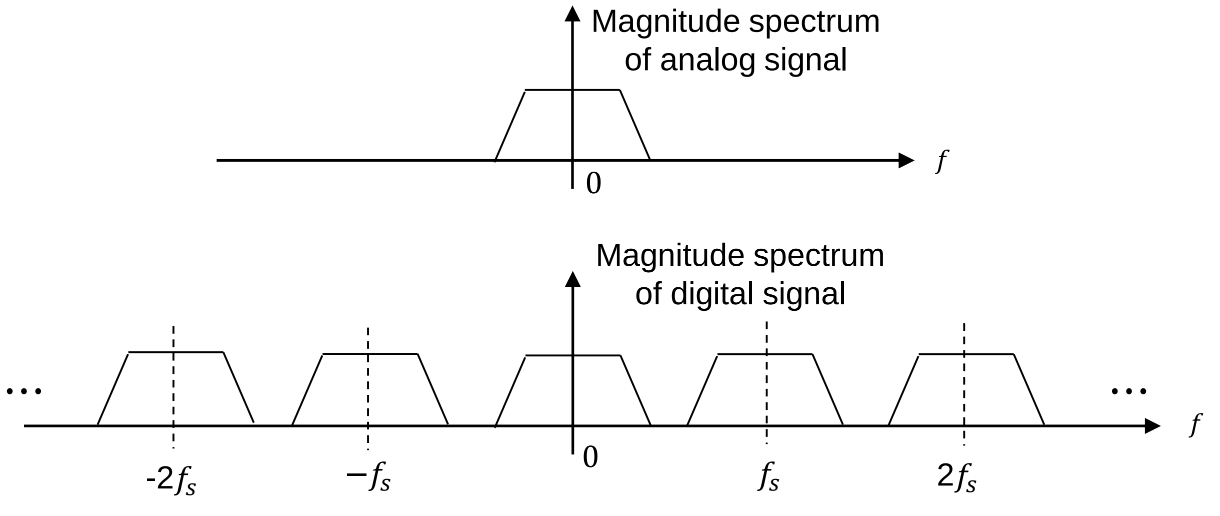
Review Questions (1)

- What is ″Spectrum Replication″?

  When an analog signal is sampled, the spectrum of the discrete-time signal is obtained as the periodic repetition of the analog continuous-time signal (the analog spectrum is "replicated")

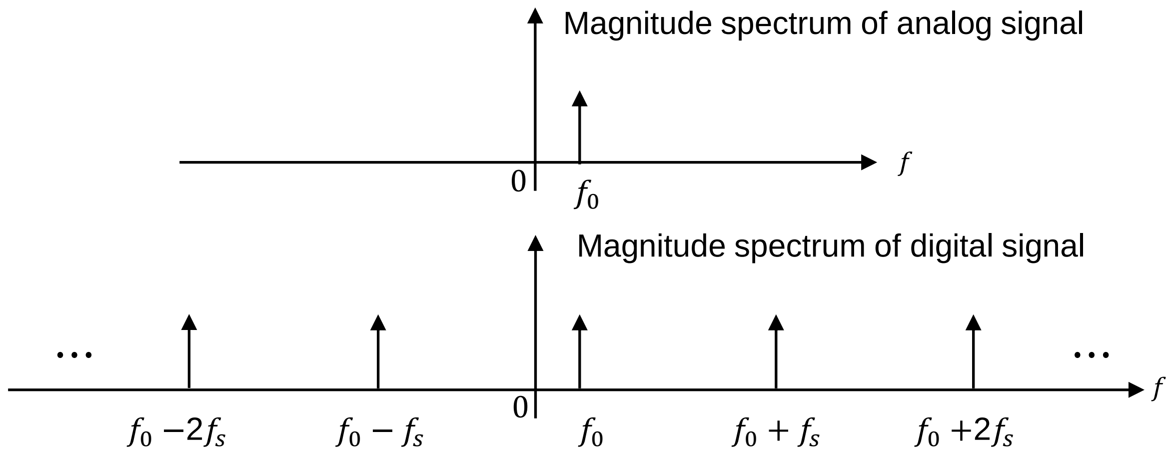
- What is the repetition period of the spectrum of the discrete-time signal?

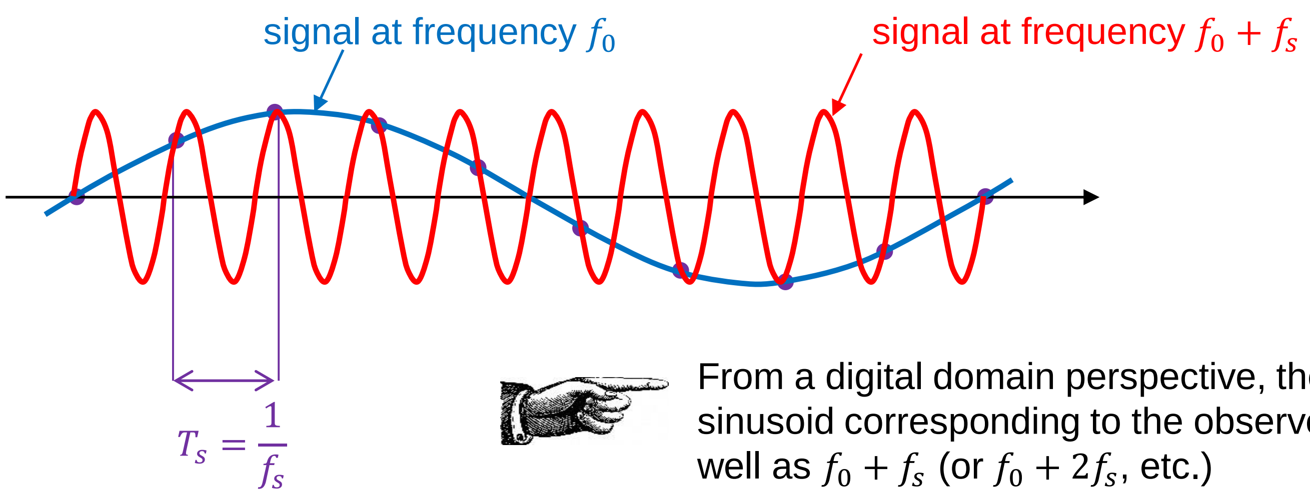
  The repetition period is f_s, the signal sampling frequency.



- How does this drawing look in the case of a single sinusoidal signal?

Review Questions (2)

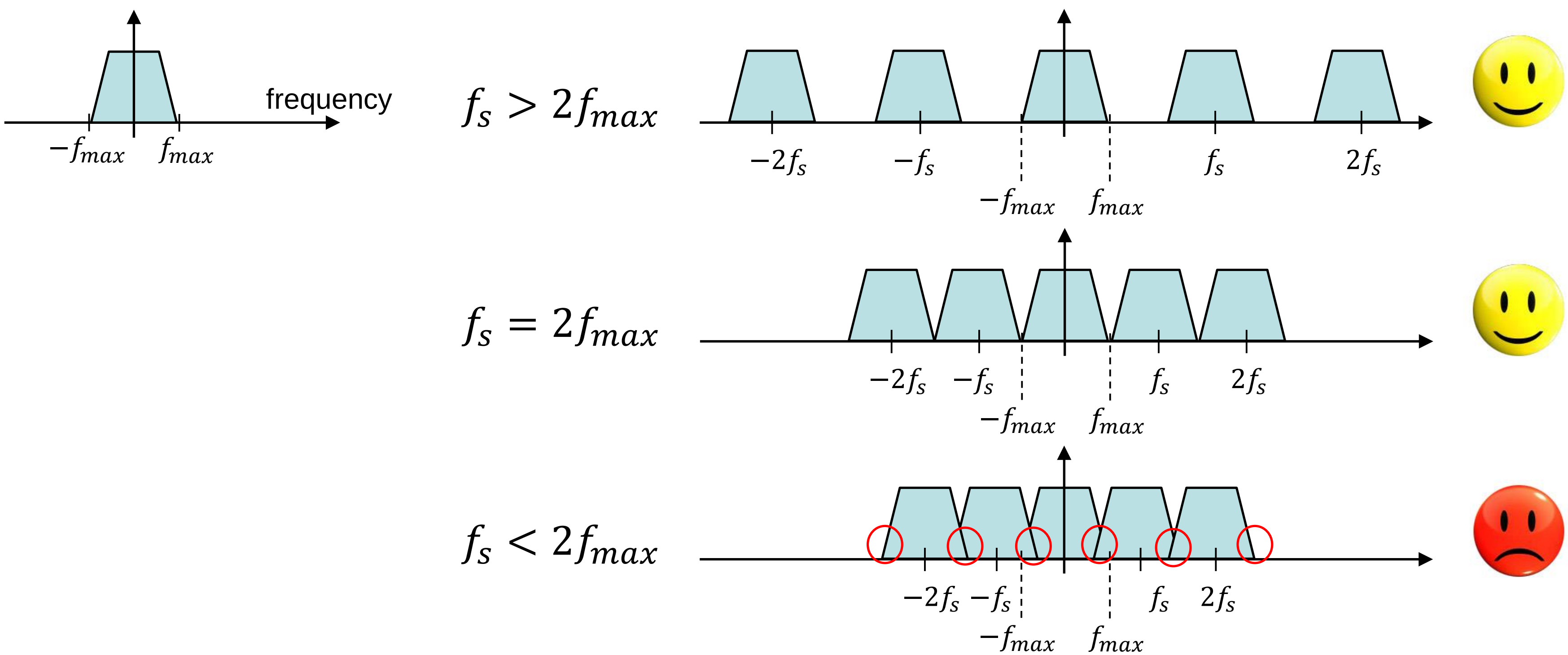




From a digital domain perspective, the frequency of the analog sinusoid corresponding to the observed samples can be f_0, as well as $f_0 + f_s$ (or $f_0 + 2f_s$, etc.)

Review Questions (3)

- A signal with spectrum shown below is sampled at frequency f_s. Sketch the spectrum of the sampled signal.



Aliasing (spectral overlap)

Review Questions (4)

- What function is realized by an analog reconstructor?

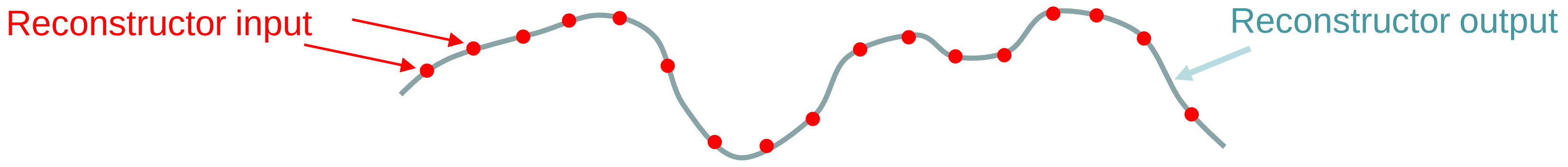
  Creation of an analog continuous-time signal from a set of digital sample values.
- How is an analog reconstructor implemented?

  As a low-pass filter.
- What is the cutoff frequency f_c of this low-pass filter?

  We usually choose $f_c = f_s/2$.
- What does the analog reconstructor do, when viewed in the frequency domain?

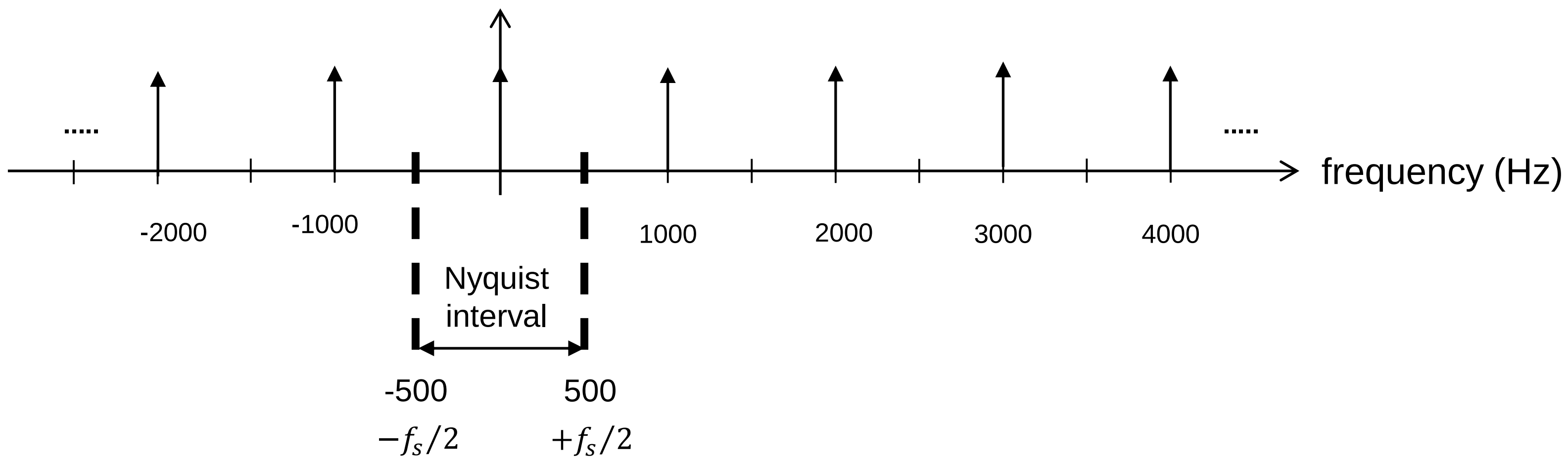
  From a frequency-domain perspective, the reconstructor blocks all (alias) spectral image components outside of the frequency band $|f| \leq f_c = f_s/2$.
- What does the analog reconstructor do, when viewed in the time domain?

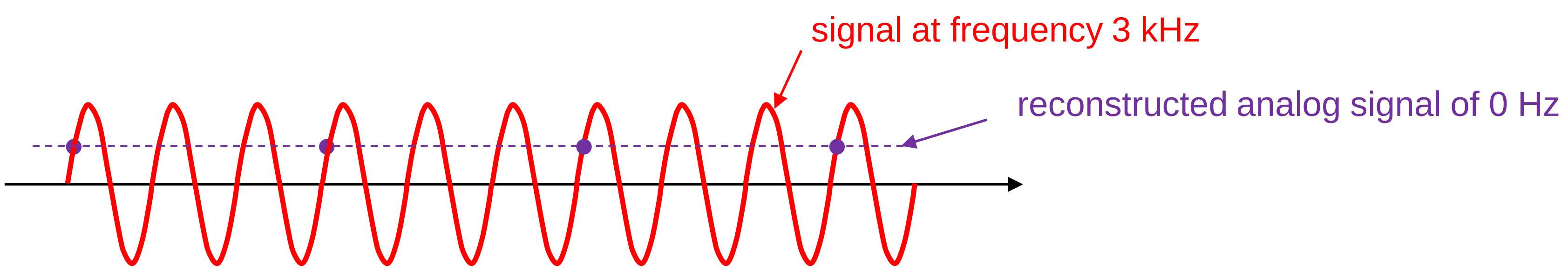
  From a time-domain perspective, the reconstructor ″connects the dots″ between discrete-time samples to yield a continuous-time signal.



Review Questions (5)

Problem: What is the alias frequency (in the Nyquist interval) of frequency 3000 Hz if the sampling frequency is 1000 Hz?



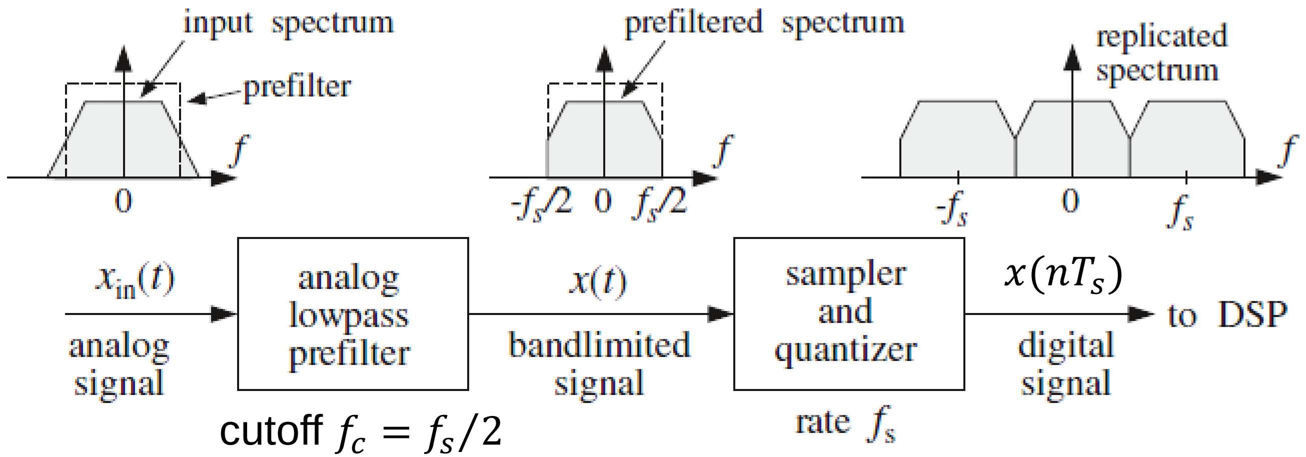


Review Questions (6)

- What is the function of an anti-aliasing filter? To avoid aliasing effects due to sampling.
- How is an anti-aliasing filter implemented? As a low-pass filter.
- How is the cutoff frequency f_c of this low-pass filter chosen?

  It is chosen to bandlimit the analog signal to be sampled to frequencies below $f_s/2$.

  It is must also be chosen so that it does not significantly alter the spectral characteristic of the analog signal.



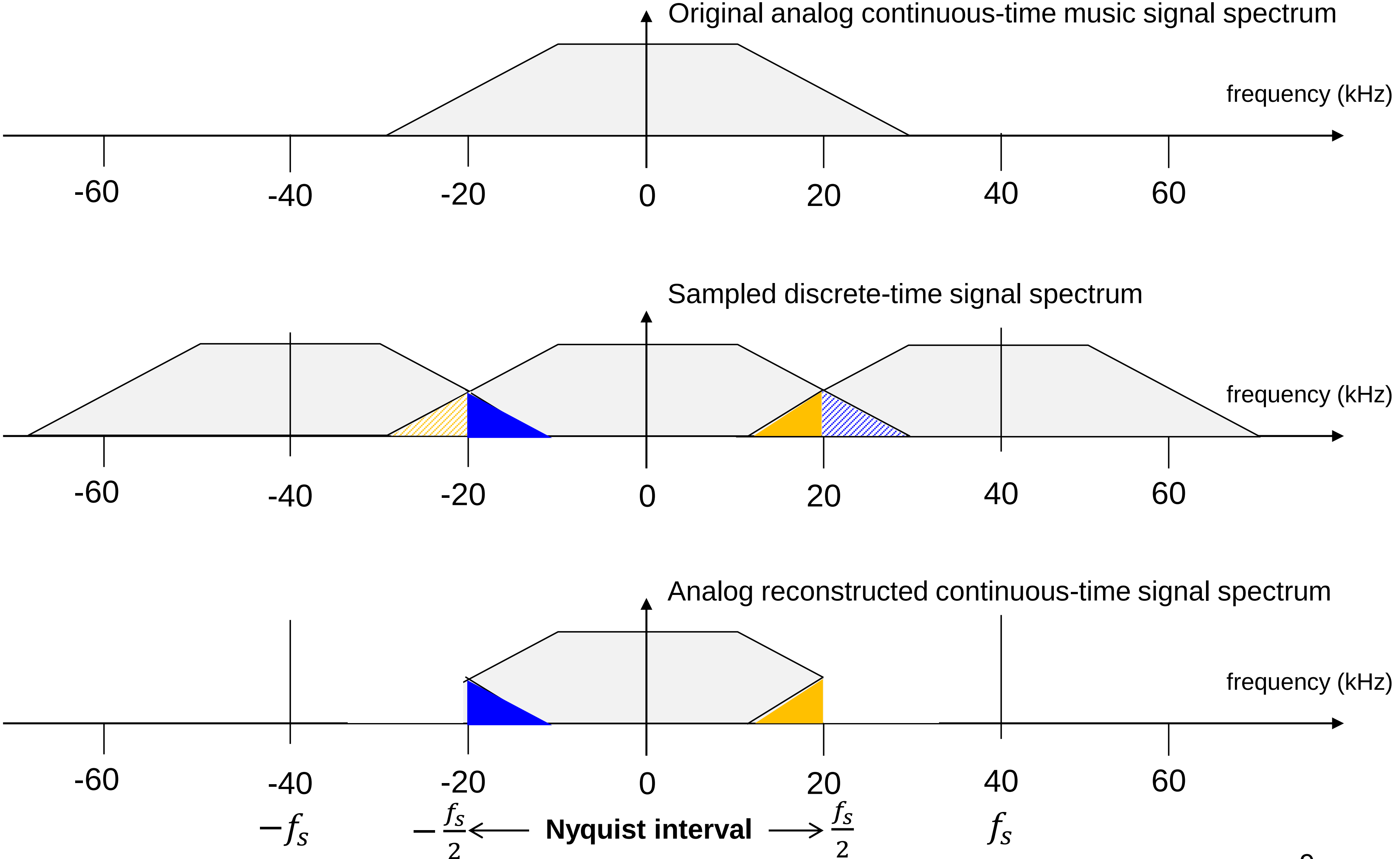
Example

Suppose a piece of music is sampled at a rate of 40 kHz without using an anti-aliasing prefilter. Then, what will happen?

Inaudible components having frequencies greater than 20 kHz will be aliased into the Nyquist interval [−20 kHz, 20 kHz] (and become audible!) and distort the true frequency components in that interval.

When the digital signal is reconverted into analog form, the aliased frequency components will alter and distort the original music sound.

Example -- Without Anti-Aliasing Filter



Example

Suppose a piece of music is sampled at a rate of 40 kHz without using an anti-aliasing prefilter. Then, what will happen?

Inaudible components having frequencies greater than 20 kHz will be aliased into the Nyquist interval [−20 kHz, 20 kHz] (and become audible) and distort the true frequency components in that interval.

When the digital signal is reconverted into analog form, the aliased frequency components will affect and distort the original music sound.

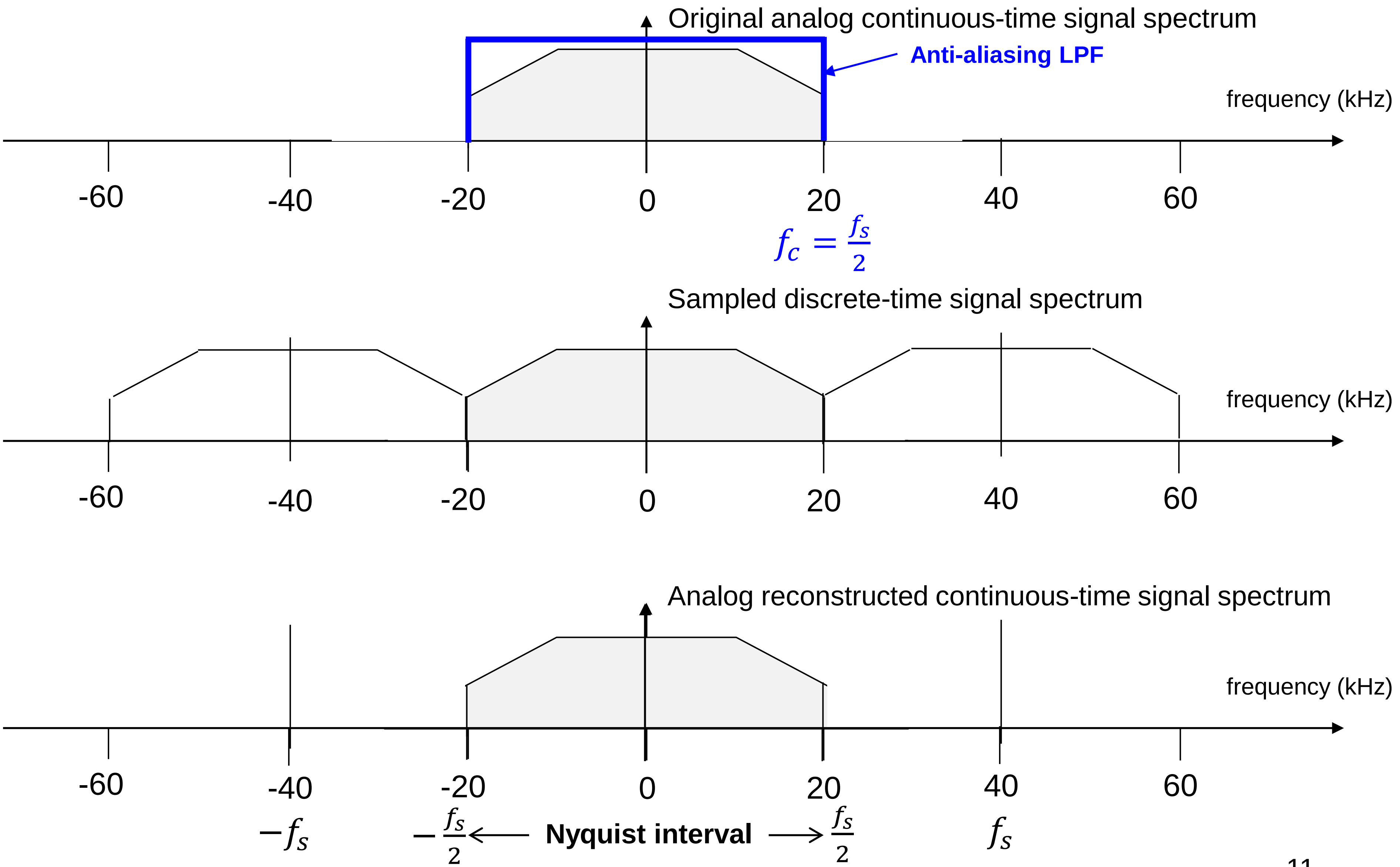
What would you recommend to do in order to avoid this problem?

Use an anti-aliasing (pre)filter before sampling the signal.

What type of filter should this be? A low-pass filter.

What should the cutoff frequency be? For example, 20 kHz.

Example -- With Anti-Aliasing Filter



Choice of the Sampling Frequency

We have seen that we need to constrain the choice of the sampling rate f_s in order to avoid loss of information on the signal during the sampling process. Could there be some other constraints for the choice of f_s?

The hardware also imposes restrictions on the choice of the sampling rate f_s.

> In real-time applications, each input sample must be acquired, quantized, and processed digitally, and digital values possibly converted back into analog format.

How does this affect the choice of the sampling rate f_s?

If the total processing or computation time required for each sample is T_{proc}, in order for the processor to be able to keep up with the incoming samples, we need the condition:

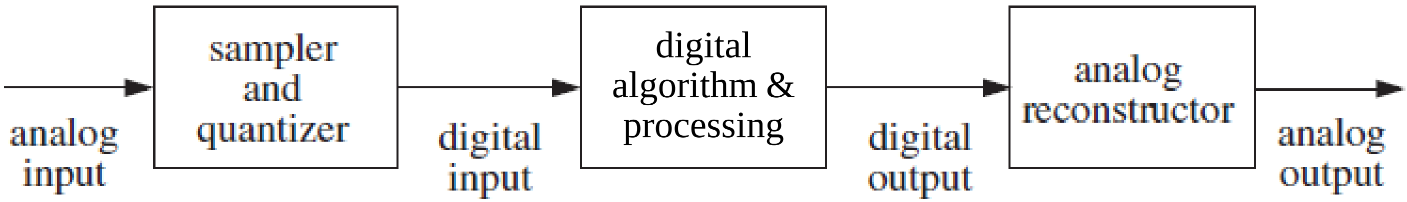
$\boxed{T_s \geq T_{proc}}$ or, defining $f_{proc} = 1/T_{proc}$: $\boxed{2f_{max} \leq f_s \leq f_{proc}}$

Note: Digital operations can be pipelined to reduce the total processing time.

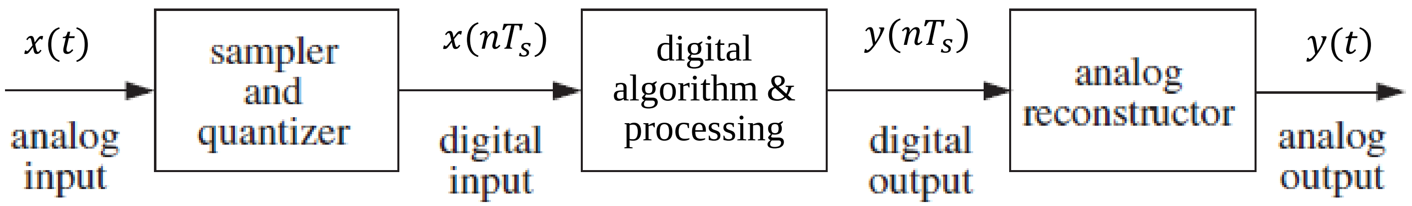
> *Pipelining is a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel fashion. Some amount of buffer storage is often inserted between elements.*

Example: Differentiation Algorithm (1)

Problem: Design a discrete-time system, like the one shown below, to differentiate continuous-time signals. Assume that the differentiator is used in an audio system having an input signal bandwidth of below 20 kHz.



Example: Differentiation Algorithm (2)



The output $y(t)$ is required to be the derivative of the input $x(t)$.

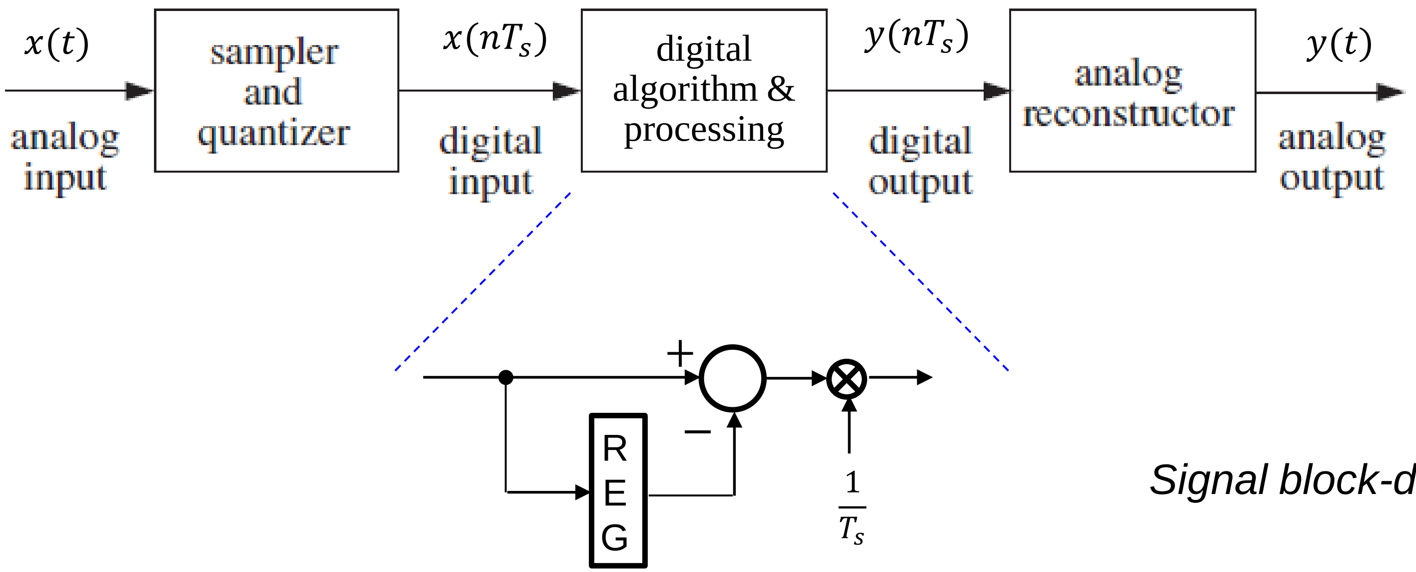
We require that $$y(t) = \frac{dx(t)}{dt}$$

Therefore at $t = nT_s$: $$y(nT_s) = \left.\frac{dx(t)}{dt}\right|_{t=nT_s}$$

$$= \lim_{T_s \to 0} \frac{1}{T_s}\left[x(nT_s) - x[(n-1)T_s]\right]$$

For T_s sufficiently small: $$\boxed{y(nT_s) = \frac{1}{T_s}\left[x(nT_s) - x[(n-1)T_s]\right]}$$ **Differentiation algorithm.**

Example: Differentiation Algorithm (3)



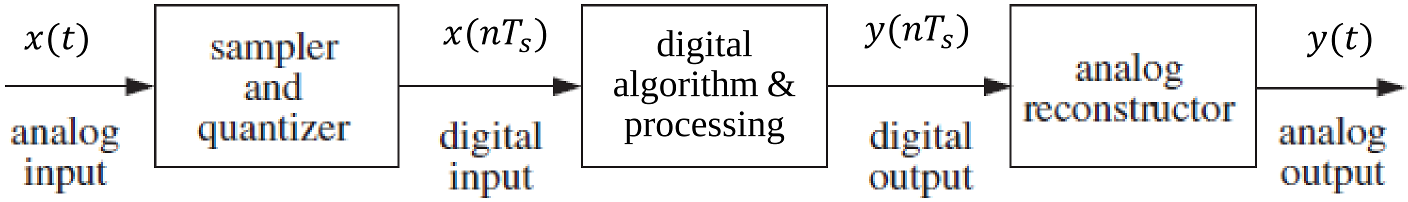
Signal block-diagram

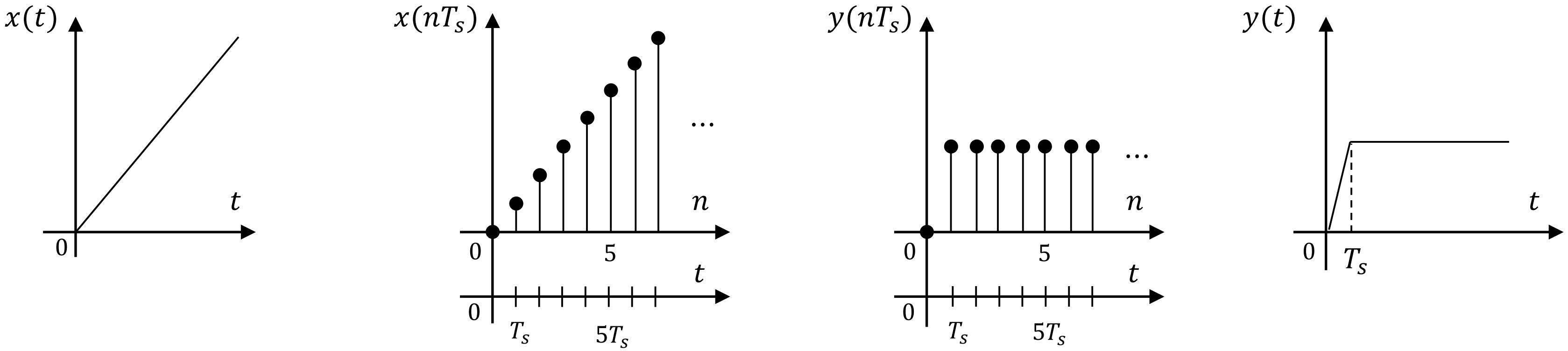
What should the value of T_s be?

To process frequencies below 20 kHz: $T_s \leq \dfrac{1}{2 \times \text{highest frequency}} = \dfrac{1}{40'000} = 25\ \mu s$

To see how well this signal processing algorithm works, consider an input signal in the form of a ramp: $x(t) = t, \quad t \geq 0$. Sketch the signals $x(t), x(nT_s), y(nT_s), y(t)$.

Example: Differentiation Algorithm (4)





Example: Averaging Algorithm (1)

Problem: Design a discrete-time system to compute the cumulative average of continuous-time signals.

To simplify the notation, let $x(nT_s) = x_n$.

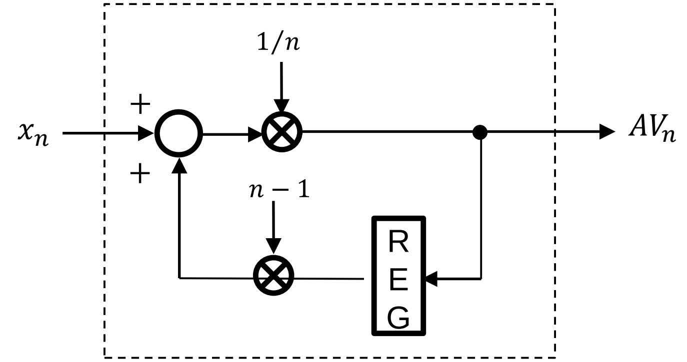
The cumulative average at time nT_s is obtained as: $AV_n = \frac{x_1 + \cdots + x_n}{n}$

Note that $x_{n+1} = (x_1 + \cdots + x_{n+1}) - (x_1 + \cdots + x_n)$

$$x_{n+1} = (n+1)\ AV_{n+1} - n\ AV_n$$

$$\Rightarrow AV_{n+1} = \frac{x_{n+1} + n\ AV_n}{n+1}$$

$$\Rightarrow \boxed{AV_n = \frac{x_n + (n-1)\ AV_{n-1}}{n}}$$



Example: Averaging Algorithm (2)

Different approach:

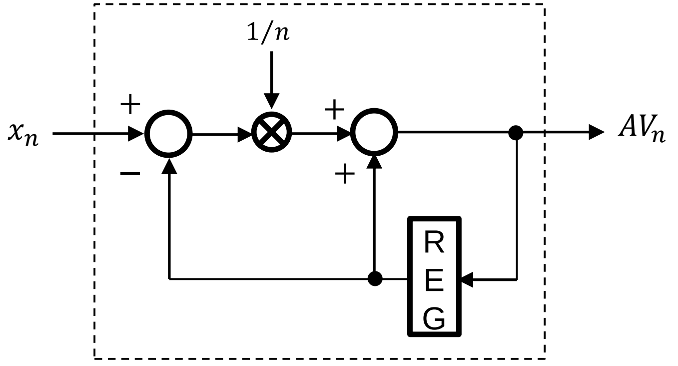
$$x_{n+1} = (n+1)\ AV_{n+1} - n\ AV_n$$

$$\Rightarrow AV_{n+1} = \frac{x_{n+1} + n\ AV_n}{n+1}$$

$$= \frac{x_{n+1} + (n+1-1)\ AV_n}{n+1}$$

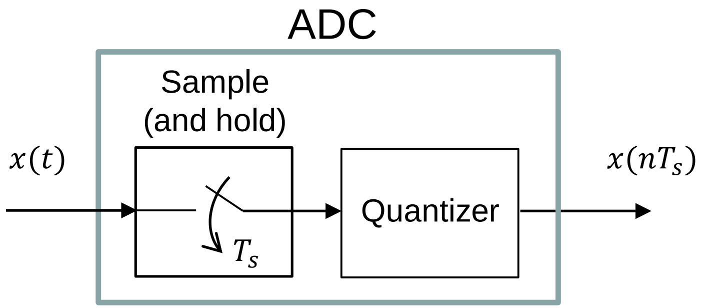
$$= \frac{(n+1)AV_n + x_{n+1} - AV_n}{n+1}$$

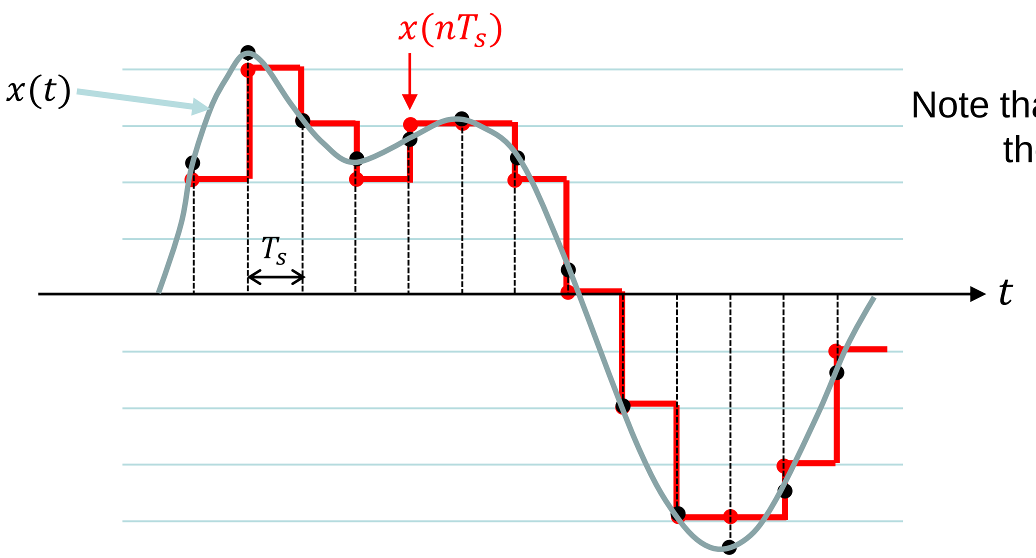
$$\Rightarrow \boxed{AV_{n+1} = AV_n + \frac{x_{n+1} - AV_n}{n+1}}$$



Analog-to-Digital Conversion

- Objective: Take samples of a continuous-amplitude waveform and map them to a finite set of amplitudes.
- The hardware component that performs this operation is the analog-to-digital converter (ADC)

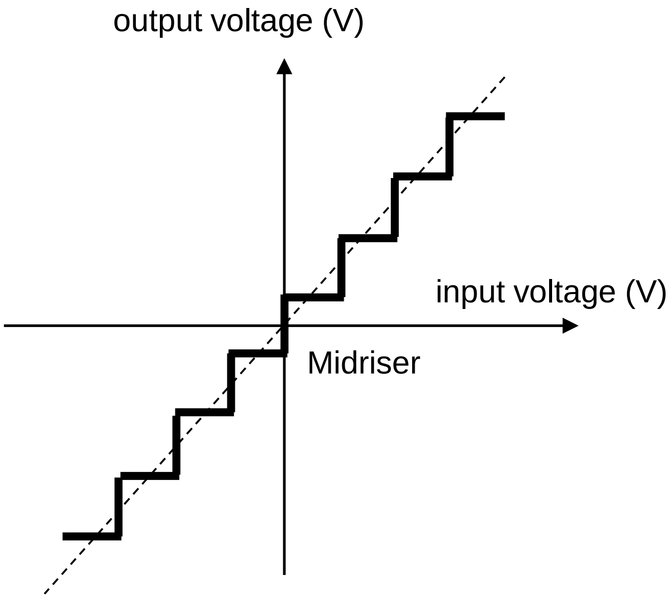


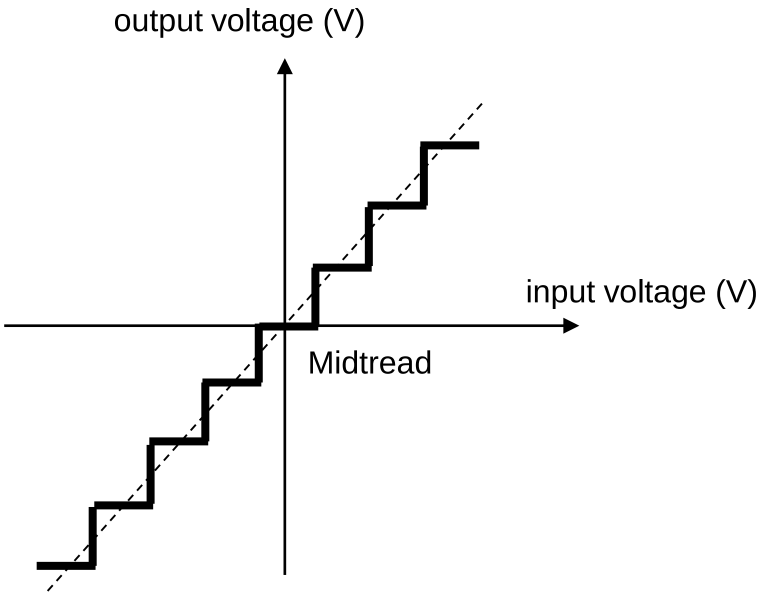


Note that quantization distorts the original signal !

Quantizer

- Uniform quantizers exhibit equally spaced increments between possible quantized output levels

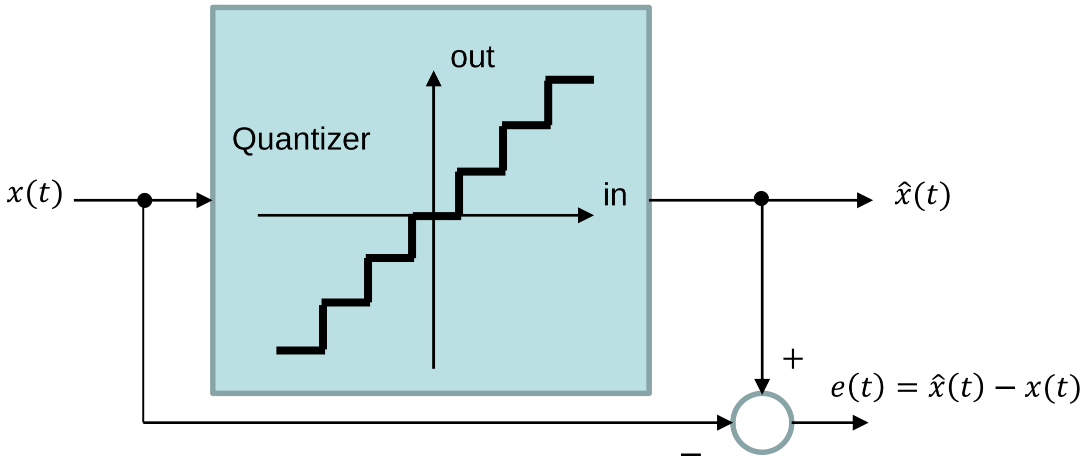




- Midtread quantizer: absence of output level changes when the input to the quantizer is low-level idle noise

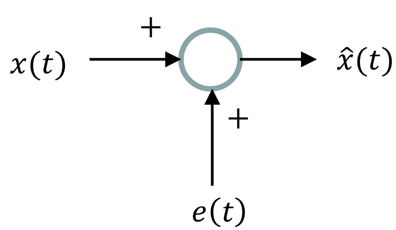
Quantizing Noise

- The difference between the input $x(t)$ and the output $\hat{x}(t)$ of a quantizer is called the <u>quantizing error</u> $e(t)$



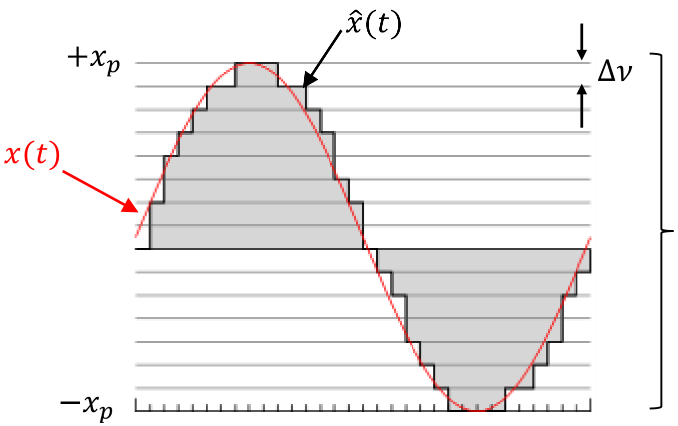
- Quantization affects the quality of the signal! How can we measure this?

- We can visualize forming $\hat{x}(t)$ by adding to each $x(t)$ an error (or noise) signal $e(t)$



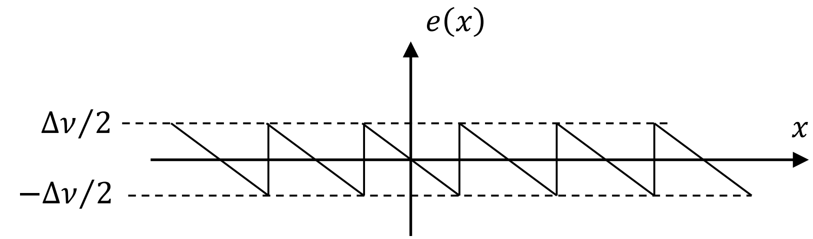
Model of quantizer

Quantizing Noise



$\Delta v = \frac{2x_p}{L}$: quantization step

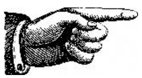
L quantization levels



For a random signal $x(t)$, we can assume that the quantizing error $e(t)$ is a uniformly distributed random variable [probability density function $p(e)$] within the interval $\left[-\frac{\Delta v}{2}, +\frac{\Delta v}{2}\right]$.

$$p(e) = \frac{1}{\Delta v} \qquad -\frac{\Delta v}{2} \le e \le \frac{\Delta v}{2}$$

Power (variance) of quantizing noise: $N_q = \int_{-\infty}^{\infty} e^2\, p(e)\, de = \frac{1}{\Delta v} \int_{-\Delta v/2}^{\Delta v/2} e^2 de = (\Delta v)^2/12 = \frac{x_p^2}{3L^2}$



Therefore, each additional bit in the conversion process reduces the quantization noise power by 6 dB!

Quantizing Noise -- Example

A signal within the range of [-5,+5] is quantized with 5 bits. Assuming the signal average power to be equal to 10, what is the signal to quantization-noise ratio in dB?

$L = 2^5 = 32$ levels.

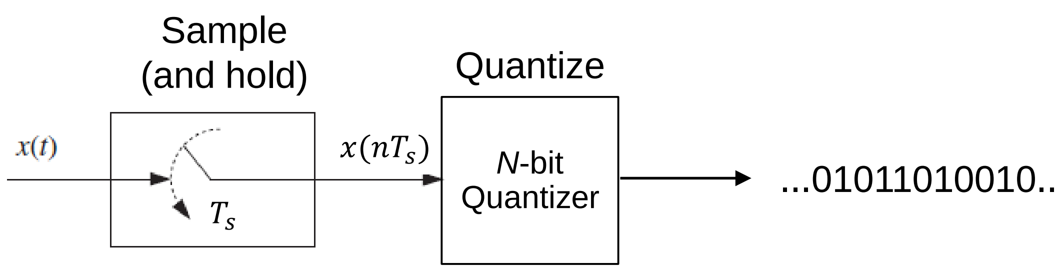
$\Delta v = \frac{10}{32} = 0.313.$

$N_q = \frac{x_p^2}{3L^2} = \frac{25}{3 \times 32^2} = 8.14 \times 10^{-3}.$

Signal to quantization noise ratio: $SNR = 10\ \log_{10} \frac{10}{8.14 \times 10^{-3}} \cong 31$ dB.

Pulse Code Modulation (PCM) (1)

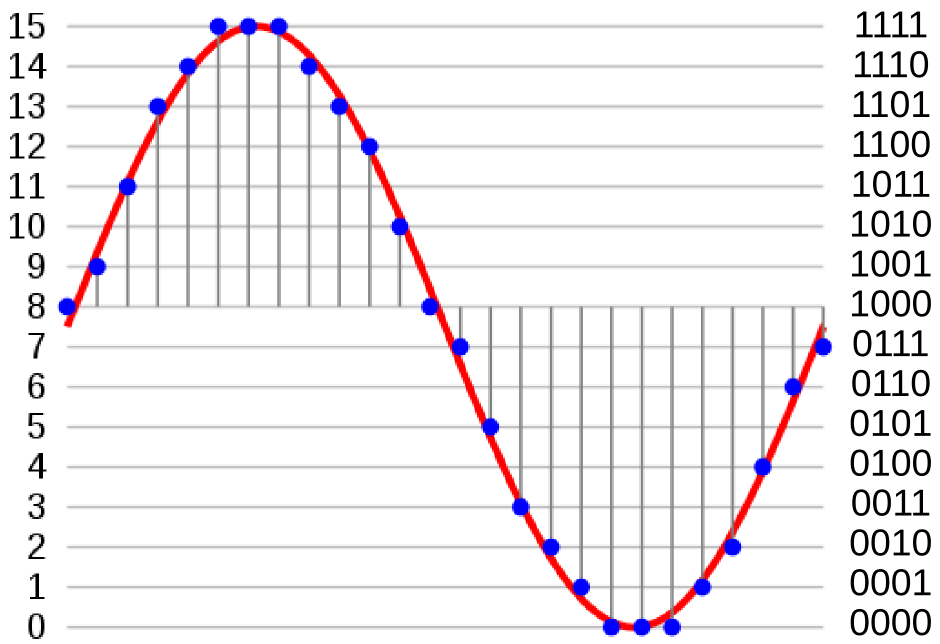
- Once the quantization levels are obtained, they need to be represented in a digital form.
- A common approach is to represent each level by a number of *N* bits
- Then the analog signal is converted to a bit stream – this is called Pulse Code Modulation (PCM)
- PCM is the standard form of digital audio in computers, Compact Discs, DVDs, digital telephony and other digital audio applications



Pulse Code Modulation (PCM) (2)

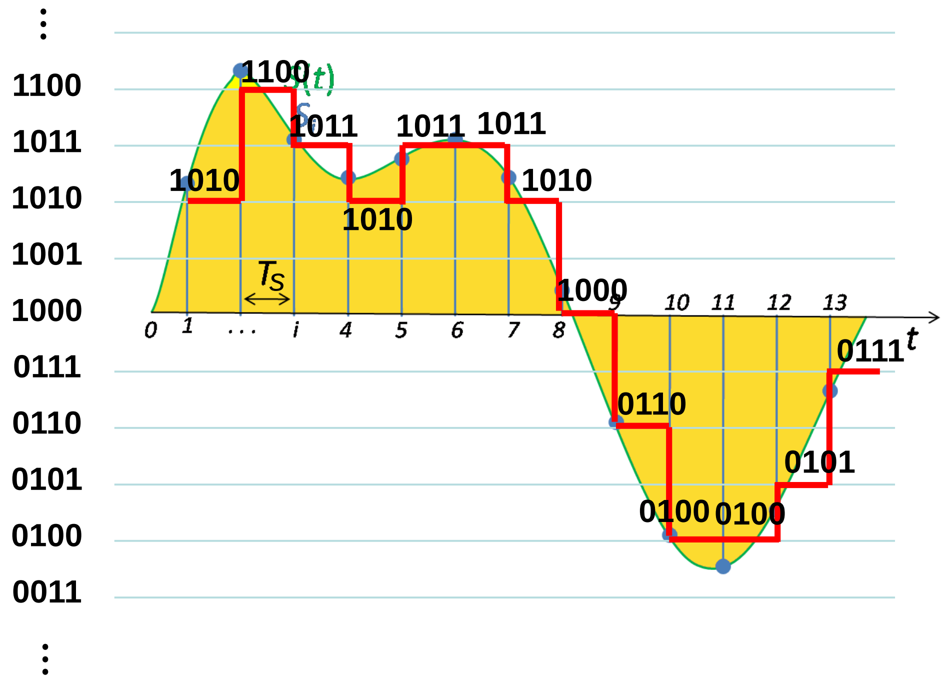
- Hence to obtain a PCM bit stream, the amplitude of the analog signal is sampled regularly at uniform intervals, and each sample is quantized (rounded off) to the nearest value (represented by *N* bits) within a range of digital steps.

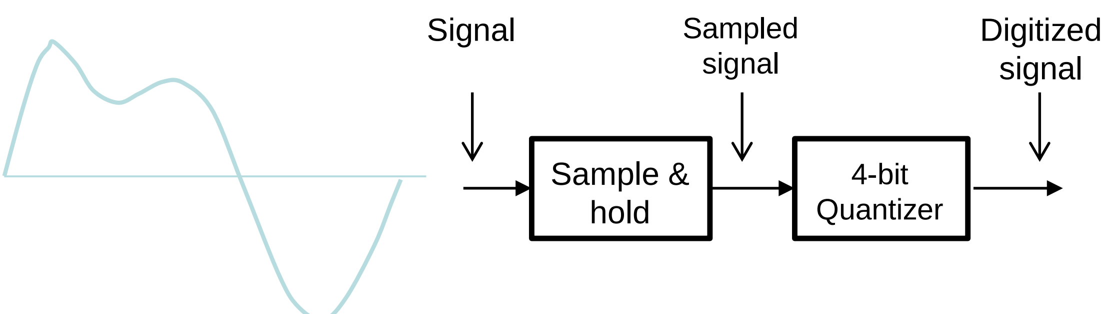
Sampling and quantization of an analog signal (red) for 4-bit PCM



- The pulse code shown on the right is the natural binary code, NBC.
- However, other codes could be used as well (such as, 2’s complement, etc.)

Signal Quantization and PCM





1010110010111010101110111010...

Pulse Code Modulation (PCM) (3)

- A PCM stream has two basic properties that determine the stream's fidelity to the original analog signal:
  - the sampling rate, i.e., the number of times per second that samples are taken
  - the bit depth N, which determines the number of possible digital values that can be used to represent each sample.
- Common sample depths for PCM are N = 8, 16, 20 or 24 bits per sample.
- The sampling theorem shows that PCM devices can operate without introducing distortions within their designed frequency bands if they provide a sampling frequency at least twice that of the highest frequency contained in the input signal.
- For example, DVD-Audio is 24-bit, with a sampling rate of **96 kHz**; in comparison, DVD-Video soundtrack is 16-bit, with a sampling rate of **48 kHz**, and standard audio CD is 16-bit, with a sampling rate of **44.1 kHz**.

PCM Example

- The audio bandwidth is about 15 kHz but, for speech, signal intelligibility is not affected if all the components above 3400 Hz are suppressed
- Since the objective in a telephone communication is intelligibility rather than high fidelity, the components above 3400 Hz are eliminated by a low-pass filter
- The resulting signal is then sampled at a rate of 8000 samples per second (8 kHz) (Why?)
- Each sample is finally quantized into 256 levels ($L = 256$), which requires a group of eight binary pulses to encode each sample
- So what is the bit rate corresponding to a telephone signal?
- The telephone signal requires $8 \times 8000 = 64'000$ bits per second (or binary pulses per second)