Digital Signal Processing
ADC and DAC

Moslem Amiri, Václav Přenosil
Masaryk University

Resource: “The Scientist and Engineer's Guide to Digital Signal Processing”
(www.dspguide.com)
By Steven W. Smith

Quantization

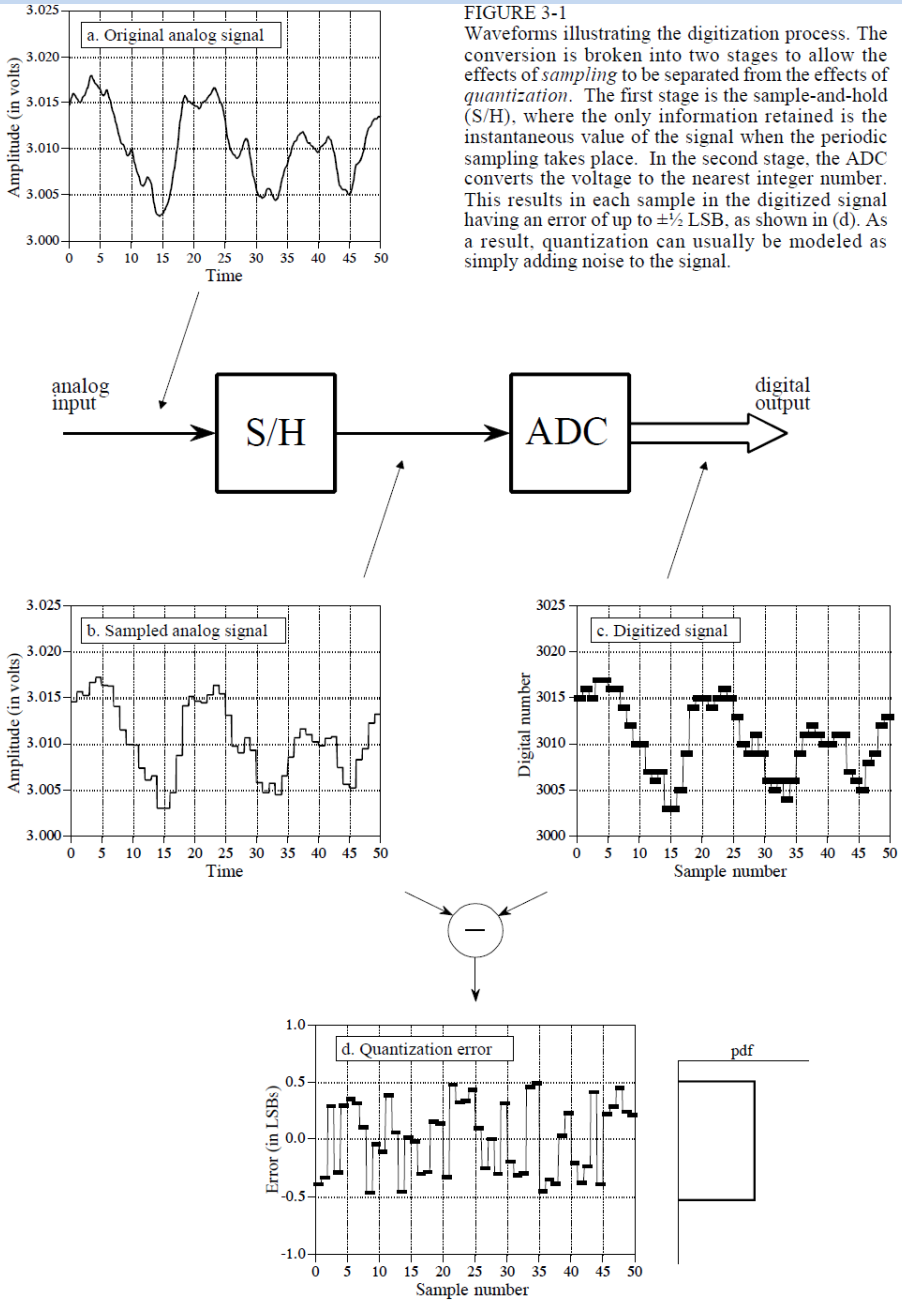


FIGURE 3-1
Waveforms illustrating the digitization process. The conversion is broken into two stages to allow the effects of *sampling* to be separated from the effects of *quantization*. The first stage is the sample-and-hold (S/H), where the only information retained is the instantaneous value of the signal when the periodic sampling takes place. In the second stage, the ADC converts the voltage to the nearest integer number. This results in each sample in the digitized signal having an error of up to ±½ LSB, as shown in (d). As a result, quantization can usually be modeled as simply adding noise to the signal.

Quantization

- In previous figure
  - Block diagram broken into 2 sections (theoretical model)
    - Sample-and-hold (S/H) → sampling converts independent variable (time) from continuous to discrete
    - Analog-to-digital converter (ADC) → quantization converts dependent variable (voltage) from continuous to discrete
  - Effects of quantization
    - LSB (Least Significant Bit): distance between adjacent quantization levels
    - Any sample in digitized signal has a maximum error of ±1/2 LSB
    - Quantization error appears very much like random noise

Quantization

- Model of quantization error
  - Quantization results in addition of a specific amount of random noise to signal
  - Additive noise
    - uniformly distributed between ±1/2 LSB
    - $\mu = 0, \sigma = 1/\sqrt{12} LSB (\sim 0.29 LSB)$
  - E.g., passing an analog signal through
    - 8 bit digitizer → adds an rms noise of 0.29/256
    - 12 bit digitizer → adds a noise of 0.29/4096
    - 16 bit digitizer → adds a noise of 0.29/65536
  - Quantization error is a random noise → number of bits determines precision of data

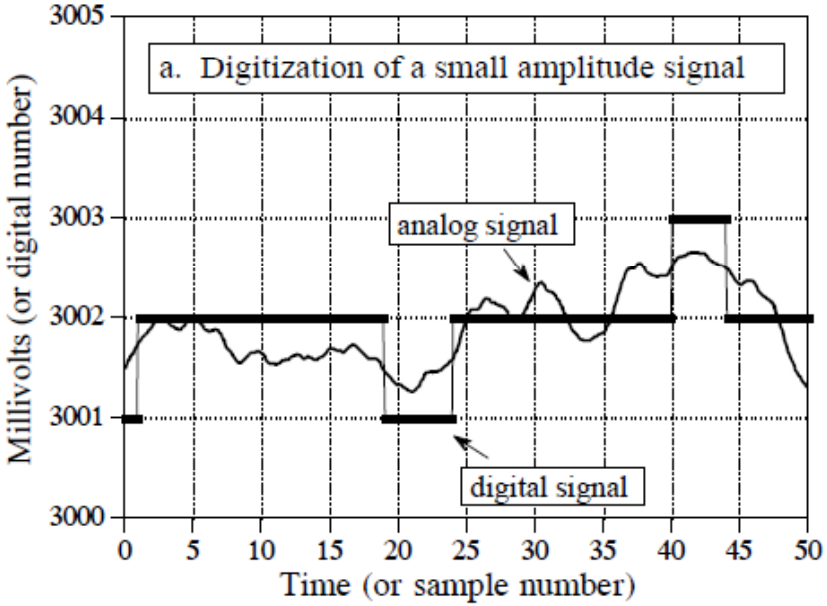
Quantization

- Example(quantization error + noise in analog signal)
  - Analog signal – maximum amplitude of 1.0 volt – random noise of 1.0 millivolt rms
  - 8 bit digitizer
    - 1.0 volt → 0.29 LSB
    - 1.0 millivolt = 0.255 * 1/255 → 0.255 LSB
    - Total noise = $\sqrt{0.29^2 + 0.255^2} = 0.386$ LSB ≈ 0.0015
    - An increase of about 50% over noise already in analog signal
  - 12 bit digitizer
    - No increase in noise, nothing would be lost due to quantization
  - Two questions when deciding how many bits needed
    - How much noise is already present in analog signal
    - How much noise can be tolerated in digital signal

Quantization

- Dithering
  - Quantization error not random
    - Occurs when analog signal remains at about same value for many consecutive samples
    - Output remains stuck on same digital number for many samples in a row
    - Quantization error model not valid
    - Dithering improves digitization of slowly varying signals
  - Small amount of random noise is added to analog signal

Quantization



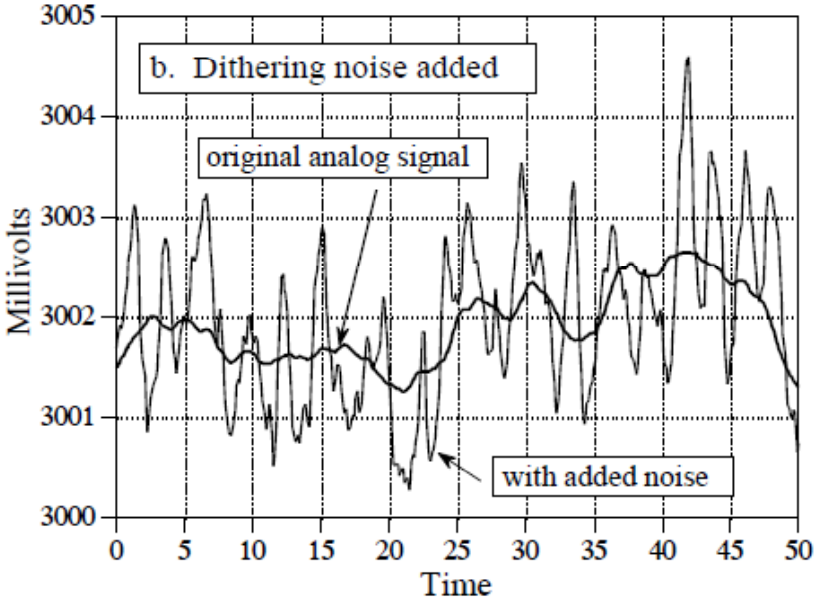
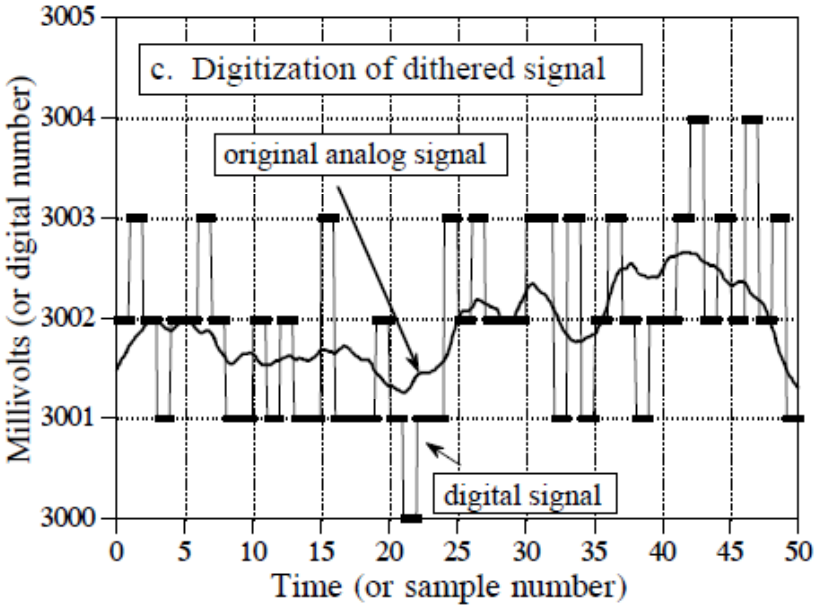


FIGURE 3-2
Illustration of dithering. Figure (a) shows how an analog signal that varies less than ±½ LSB can become *stuck* on the same quantization level during digitization. Dithering improves this situation by adding a small amount of random noise to the analog signal, such as shown in (b). In this example, the added noise is normally distributed with a standard deviation of 2/3 LSB. As shown in (c), the added noise causes the digitized signal to toggle between adjacent quantization levels, providing more information about the original signal.



The Sampling Theorem

- Proper sampling definition
  - If analog signal can be reconstructed from samples

The Sampling Theorem

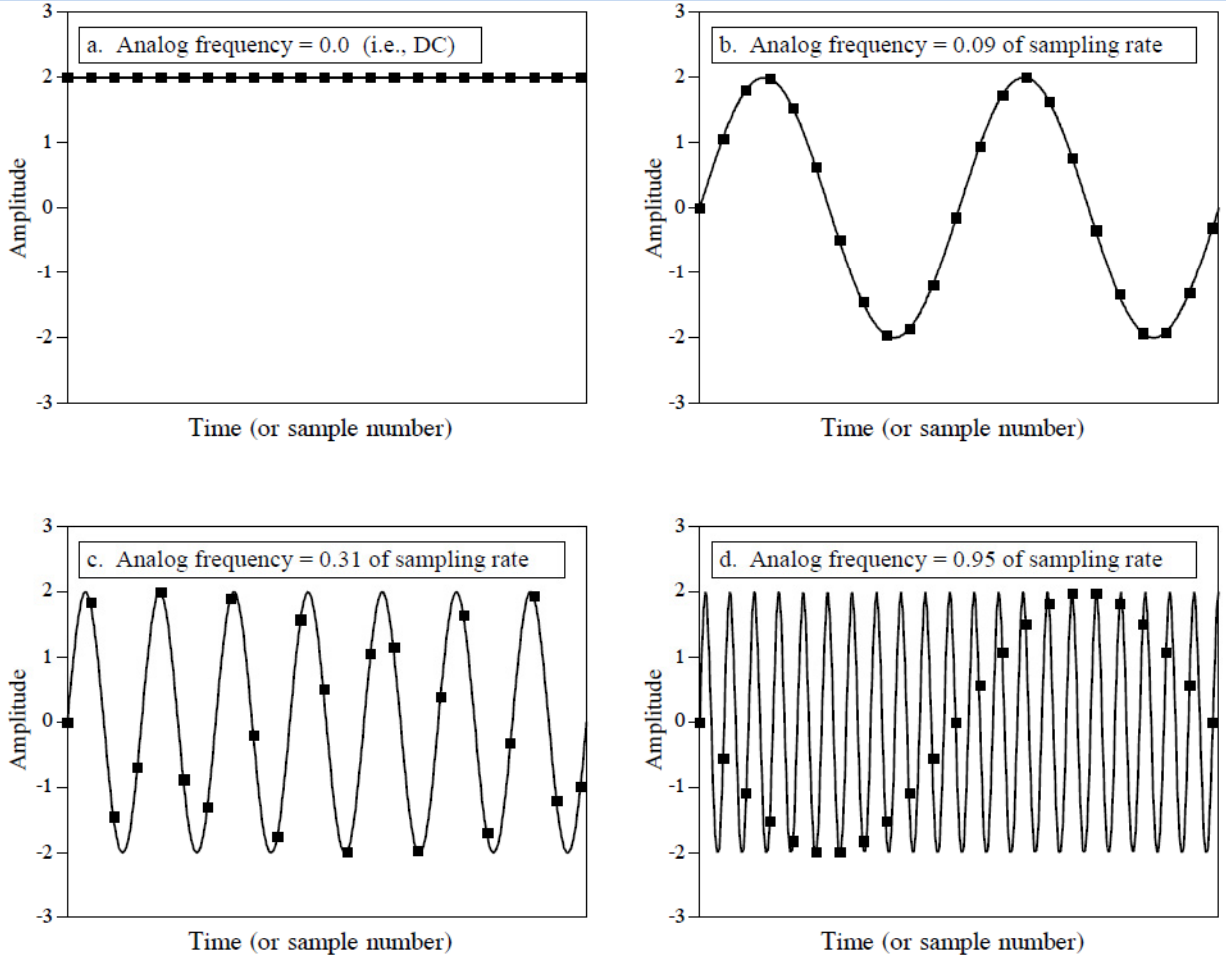


FIGURE 3-3
Illustration of proper and improper sampling. A continuous signal is sampled *properly* if the samples contain all the information needed to recreate the original waveform. Figures (a), (b), and (c) illustrate *proper sampling* of three sinusoidal waves. This is certainly not obvious, since the samples in (c) do not even appear to capture the shape of the waveform. Nevertheless, each of these continuous signals forms a unique one-to-one pair with its pattern of samples. This guarantees that reconstruction can take place. In (d), the frequency of the analog sine wave is greater than the Nyquist frequency (one-half of the sampling rate). This results in *aliasing*, where the frequency of the sampled data is different from the frequency of the continuous signal. Since aliasing has corrupted the information, the original signal cannot be reconstructed from the samples.

The Sampling Theorem

- In previous figure
  - (a), (b), (c) → proper sampling
  - (d) → improper sampling (aliasing)
- Aliasing
  - Phenomenon of sinusoids changing frequency during sampling
- Shannon (or Nyquist) sampling theorem
  - A continuous signal can be properly sampled, only if it does not contain frequency components above one-half of sampling rate
- Nyquist frequency or Nyquist rate
  - One-half of sampling rate

The Sampling Theorem

- Example
  - Sampling rate = 2000 samples/s
  - Analog signal must be composed of frequencies below 1000 cycles/s
  - Frequencies above 1000 cycles/s present in signal
    - Will be aliased to frequencies between 0 and 1000 cycles/s
    - Combines with legitimate information

The Sampling Theorem

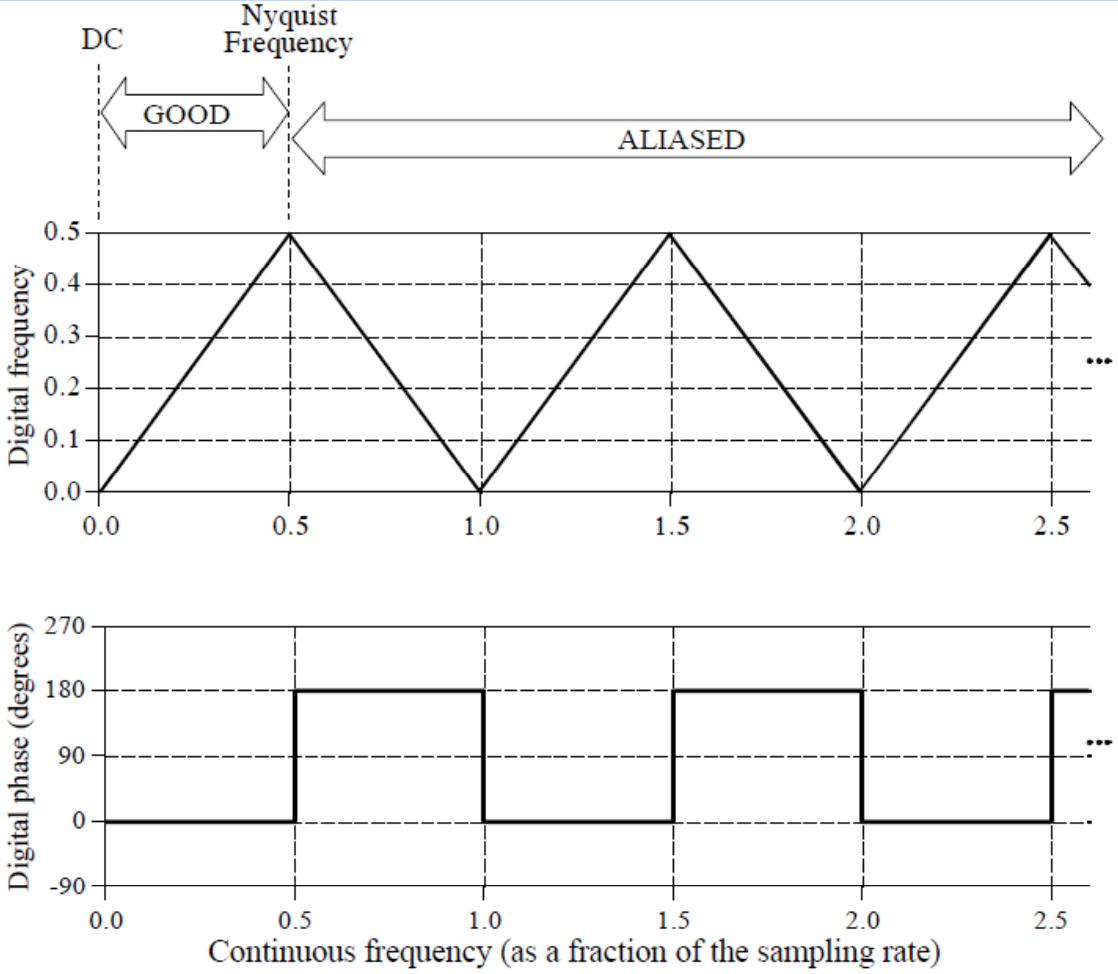


FIGURE 3-4
Conversion of analog frequency into digital frequency during sampling. Continuous signals with a frequency less than one-half of the sampling rate are directly converted into the corresponding digital frequency. Above one-half of the sampling rate, aliasing takes place, resulting in the frequency being misrepresented in the digital data. Aliasing always changes a higher frequency into a lower frequency between 0 and 0.5. In addition, aliasing may also change the phase of the signal by 180 degrees.

The Sampling Theorem

- In previous figure
  - Continuous signal's frequency above Nyquist rate
    - Continuous frequency corresponds to digital frequency between zero and one-half sampling rate
    - If a sinusoid already at this lower frequency → aliased signal will add to it → loss of information
  - Aliasing can change phase too
    - Only two phase shifts possible: 0˚ (no phase shift) and 180˚ (inversion)
- Aliasing is a double curse
  - Information can be lost about higher and lower frequency

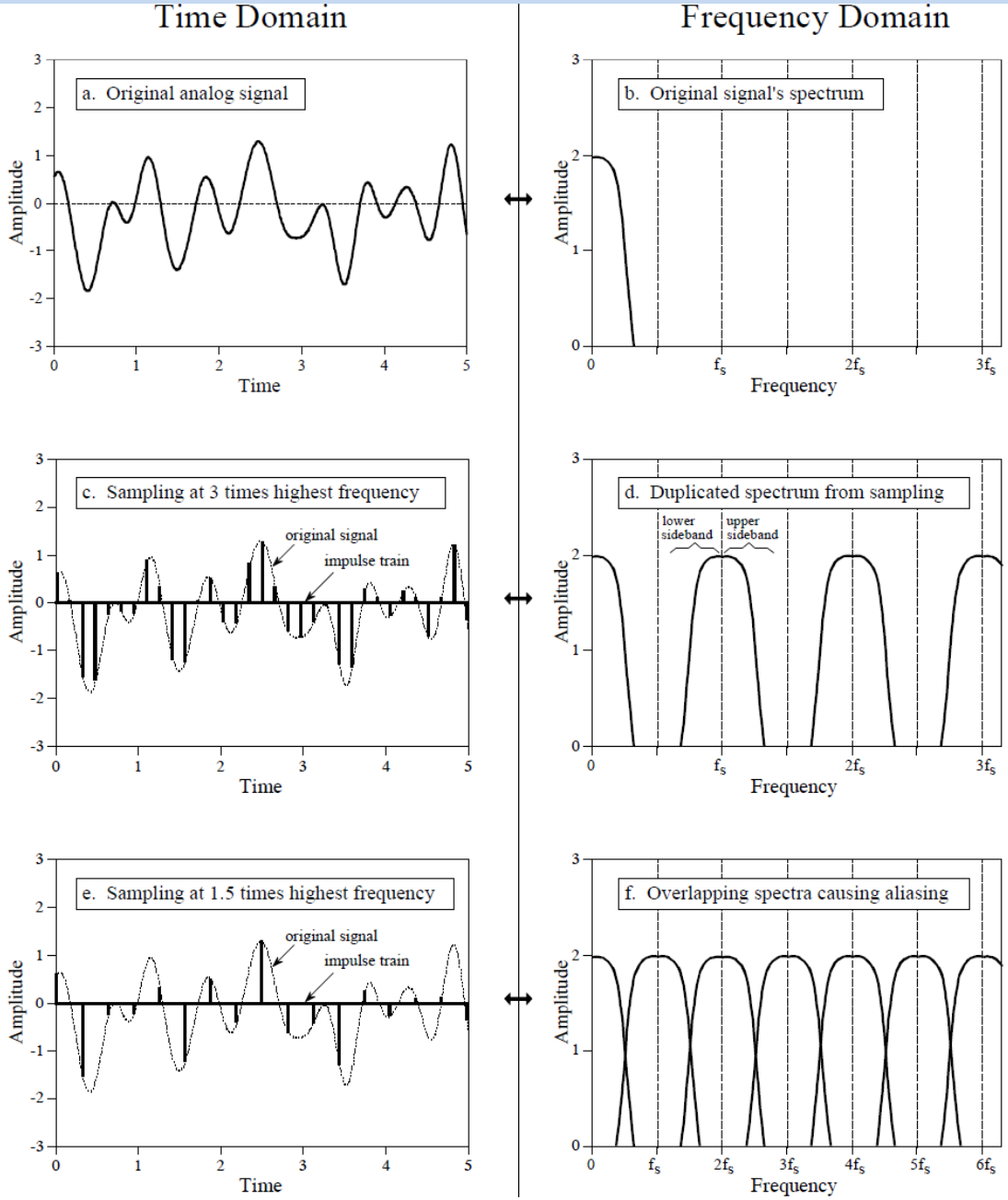
The Sampling Theorem

- Example
  - Given a digital signal containing a frequency of 0.2 of sampling rate
  - If signal obtained by proper sampling
    - Frequency of original analog signal = 0.2
  - If aliasing took place
    - Frequency of original analog signal could be any of: 0.2, 0.8, 1.2, 1.8, 2.2, …

The Sampling Theorem

- Impulse train (theoretical concept)
  - A continuous signal consisting of a series of narrow spikes (impulses) that match original signal at sampling instants
  - Can be compared with original analog signal since both are continuous

The Sampling Theorem



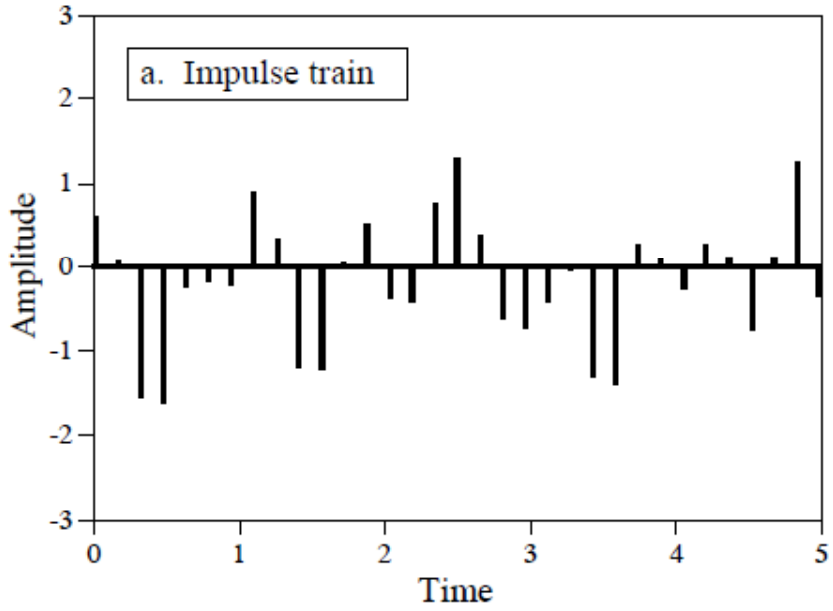
The Sampling Theorem

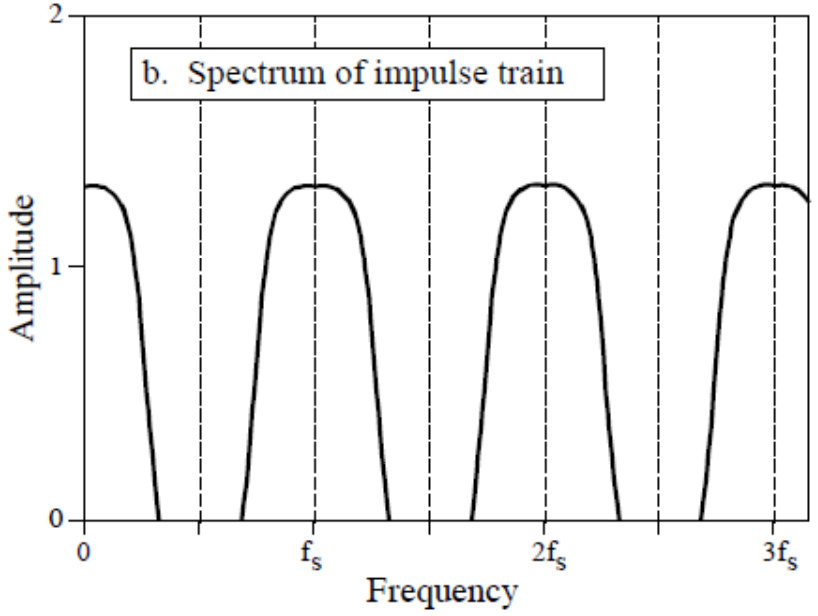
- In previous figure
  - Frequency spectrum in (b) shows (a) is composed only of frequency components between 0 and 0.33 f_s
    - f_s = sampling frequency we intend to use
  - (c) (impulse train) is proper sampling
    - (d) is a duplication of original signal spectrum
    - Upper sideband = copy of original frequency spectrum
    - Lower sideband = flipped copy
    - Transformed back into (a) by eliminating frequencies above ½f_s
  - (e) is improper sampling
    - (f) shows duplicated portions of spectrum invade band between 0 and ½ sampling frequency
    - Overlapping frequencies add together → Information loss

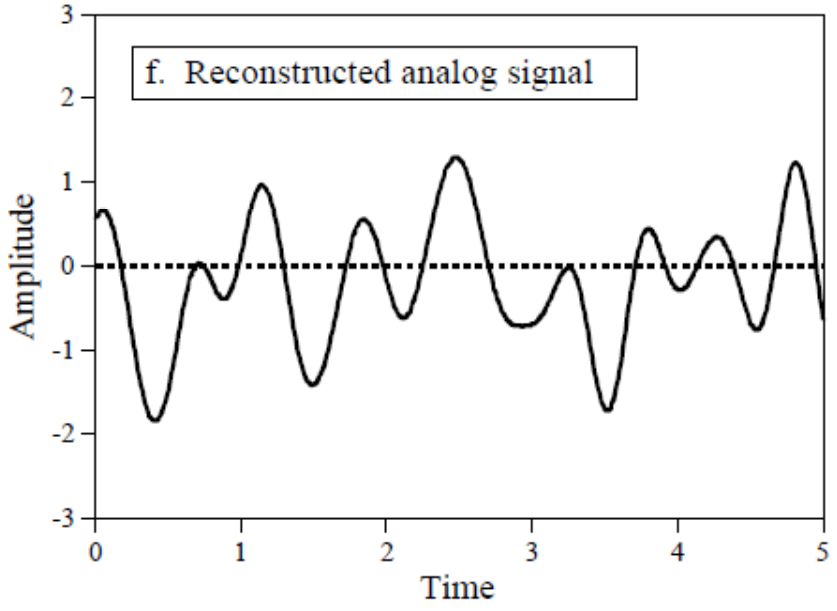
Digital-to-Analog Conversion

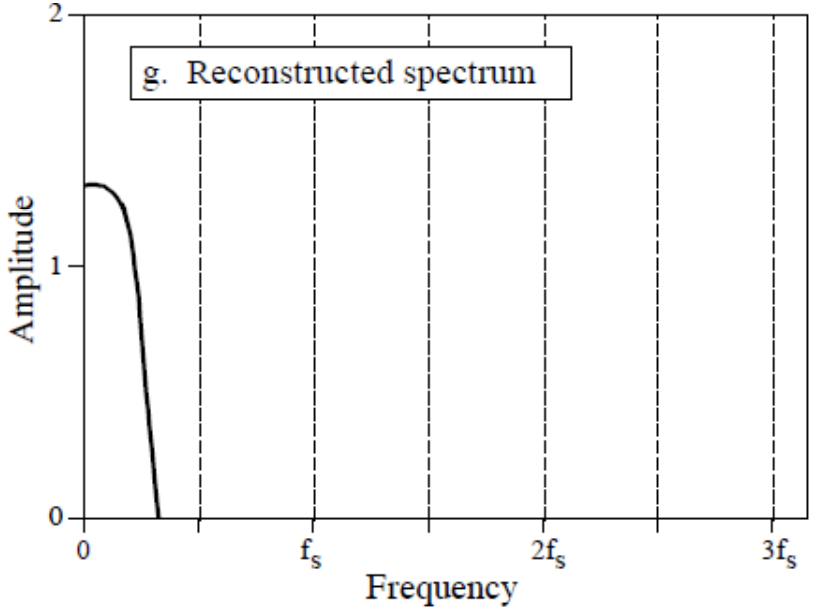
- Simplest method for DAC (in theory)
  - Pull samples from memory
  - Convert them into an impulse train
  - Original analog signal reconstructed by passing impulse train through a low-pass filter
    - Cutoff frequency = ½ sampling rate
  - Problem of this method
    - Difficult to generate required narrow pulses in electronics
  - This method shown in following figure

Digital-to-Analog Conversion









Digital-to-Analog Conversion

- Zeroth-order hold
  - DAC equivalent of sample-and-hold used during ADC
  - DACs operate by holding last value until another sample is received
  - Results in spectrum of impulse train (correct spectrum) being multiplied by sinc function (sinc (x)):

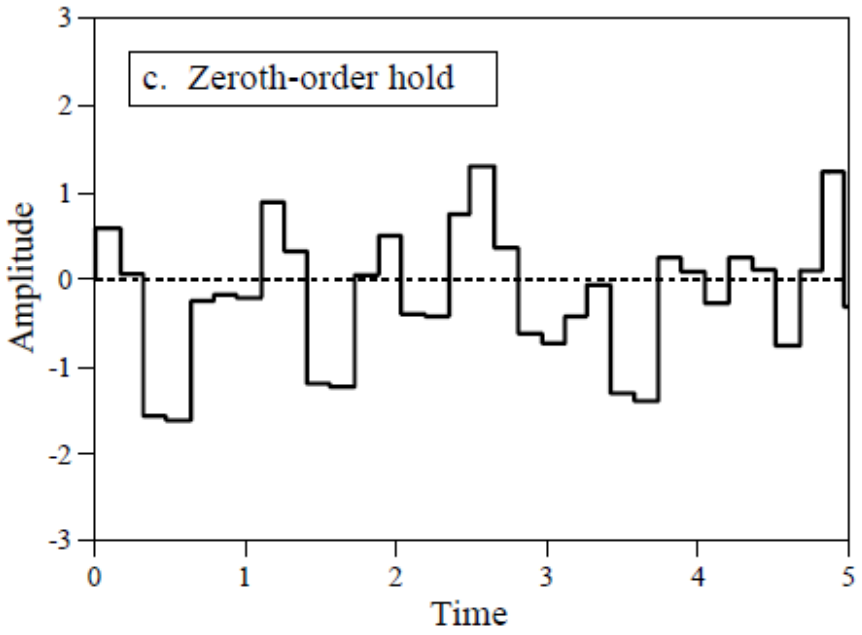
$$H(f) = \left| \frac{\sin(\pi f / f_s)}{\pi f / f_s} \right|$$

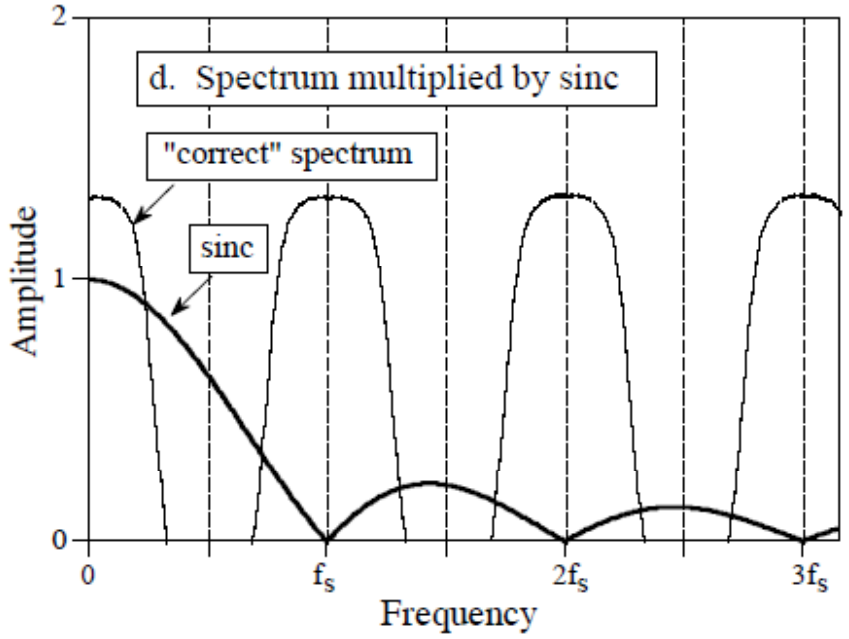
  - Analog filter used to convert zeroth-order hold signal into reconstructed signal needs to
    - Remove all frequencies above ½ sampling rate
    - Boost frequencies by 1/sinc(x)

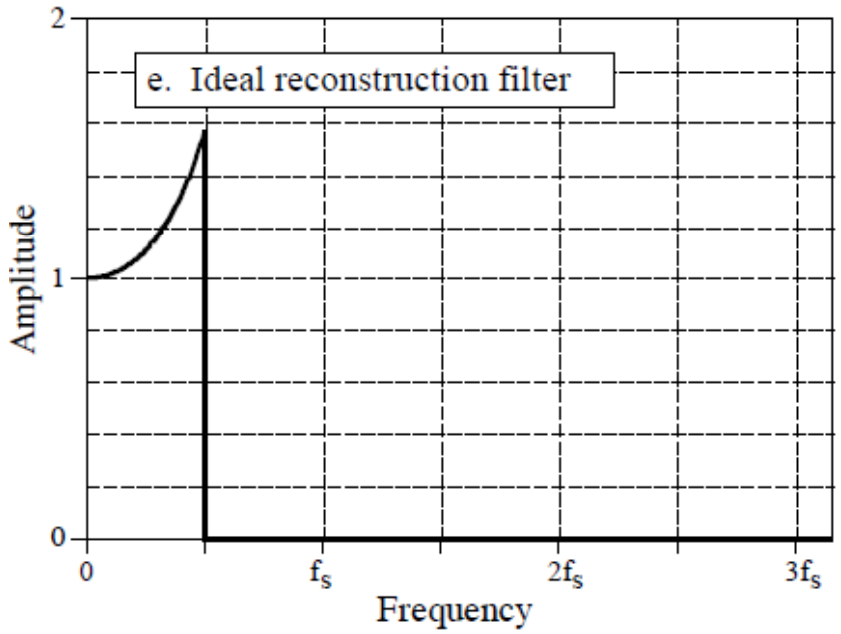
Digital-to-Analog Conversion

- 1/sinc(x) frequency boost can be handled in 4 ways
  - Ignore it – accept consequences
  - Design an analog filter to include 1/sinc(x) response
  - Use a fancy multirate technique
  - Make correction in software before DAC

Digital-to-Analog Conversion







Analog Filters for Data Conversion

- This figure shows a block diagram of a DSP system

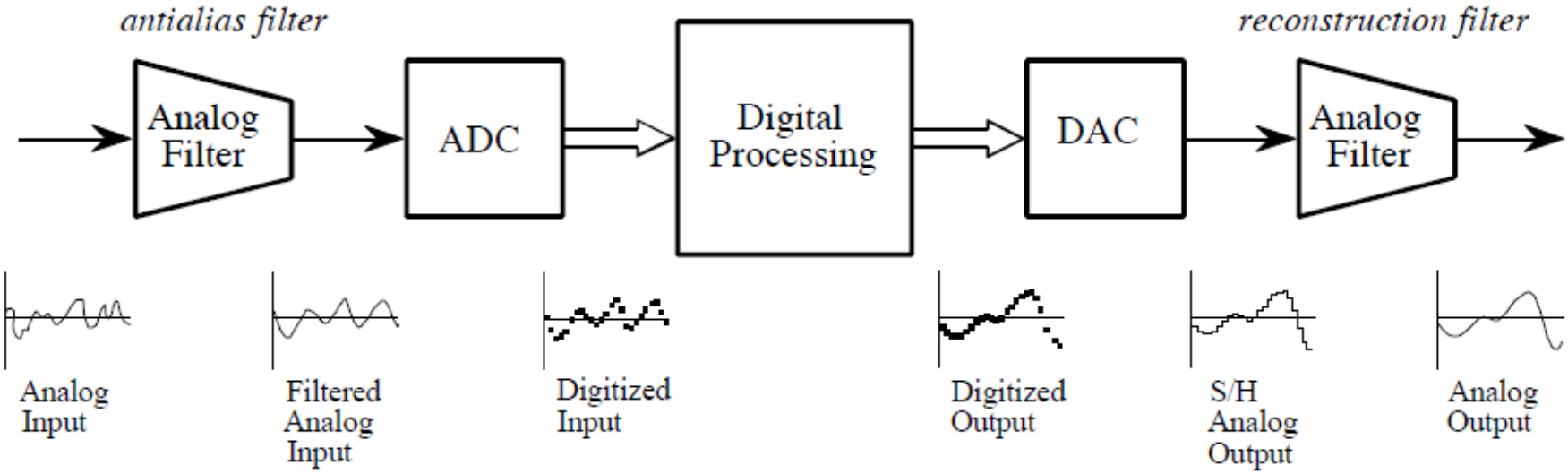


FIGURE 3-7
Analog electronic filters used to comply with the sampling theorem. The electronic filter placed before an ADC is called an *antialias filter*. It is used to remove frequency components above one-half of the sampling rate that would alias during the sampling. The electronic filter placed after a DAC is called a *reconstruction filter*. It also eliminates frequencies above the Nyquist rate, and may include a correction for the zeroth-order hold.

Analog Filters for Data Conversion

- Antialias filter
  - Placed before an ADC
  - Is an electronic low-pass filter
    - Removes all frequencies above Nyquist frequency
  - Prevents aliasing during sampling
- Reconstruction filter
  - Another low-pass filter set to Nyquist frequency
  - Placed after a DAC
  - May include zeroth-order-hold frequency boost

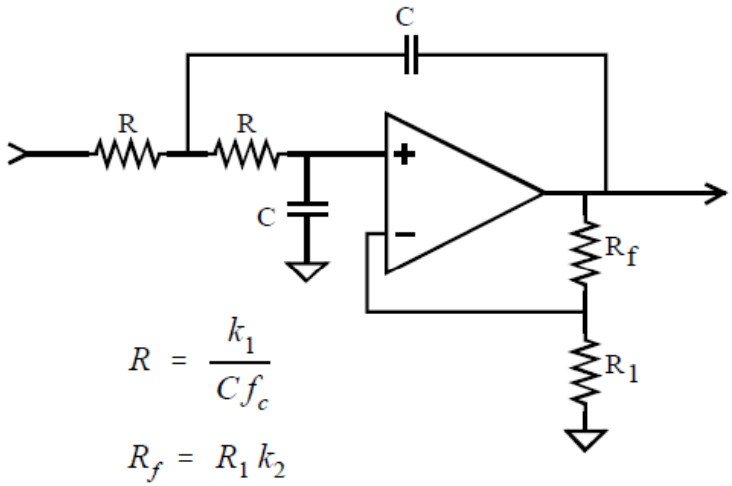
Analog Filters for Data Conversion

- Three types of analog filters
  - Chebyshev
  - Butterworth
  - Bessel (or Thompson)
- Each type of analog filters
  - Designed to optimize a different performance parameter
  - Complexity adjusted by selecting number of poles and zeros
    - E.g., six pole Bessel filter
    - More poles → more electronics → better performance
    - Different numbers describe what filter does, not a particular arrangement of resistors and capacitors

Analog Filters for Data Conversion

- Modified Sallen-Key circuit (2 pole low-pass filter)
  - Common building block for analog filter design

FIGURE 3-8
The modified Sallen-Key circuit, a building block for active filter design. The circuit shown implements a 2 pole low-pass filter. Higher order filters (more poles) can be formed by cascading stages. Find k_1 and k_2 from Table 3-1, arbitrarily select R_1 and C (try 10K and 0.01μF), and then calculate R and R_f from the equations in the figure. The parameter, f_c, is the cutoff frequency of the filter, in hertz.



$$R = \frac{k_1}{C f_c}$$

$$R_f = R_1 k_2$$

TABLE 3-1
Parameters for designing Bessel, Butterworth, and Chebyshev (6% ripple) filters.

# poles		Bessel		Butterworth		Chebyshev	
		k_1	k_2	k_1	k_2	k_1	k_2
2	stage 1	0.1251	0.268	0.1592	0.586	0.1293	0.842
4	stage 1	0.1111	0.084	0.1592	0.152	0.2666	0.582
	stage 2	0.0991	0.759	0.1592	1.235	0.1544	1.660
6	stage 1	0.0990	0.040	0.1592	0.068	0.4019	0.537
	stage 2	0.0941	0.364	0.1592	0.586	0.2072	1.448
	stage 3	0.0834	1.023	0.1592	1.483	0.1574	1.846
8	stage 1	0.0894	0.024	0.1592	0.038	0.5359	0.522
	stage 2	0.0867	0.213	0.1592	0.337	0.2657	1.379
	stage 3	0.0814	0.593	0.1592	0.889	0.1848	1.711
	stage 4	0.0726	1.184	0.1592	1.610	0.1582	1.913

Analog Filters for Data Conversion

- 4, 6, 8 pole filters
  - Formed by cascading 2, 3, 4 of building blocks, respectively
- A 6 pole Bessel filter

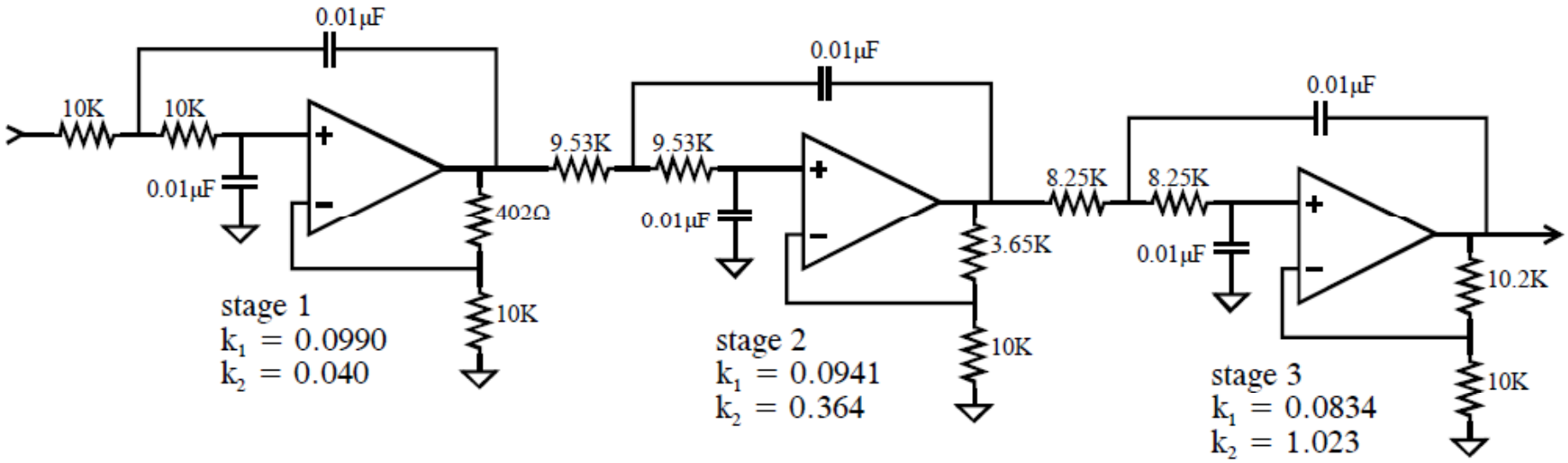


FIGURE 3-9
A six pole Bessel filter formed by cascading three Sallen-Key circuits. This is a low-pass filter with a cutoff frequency of 1 kHz.

Analog Filters for Data Conversion

- Making filter as an IC
  - Problem: difficult to make resistors directly in silicon
  - Solution: switched capacitor filter

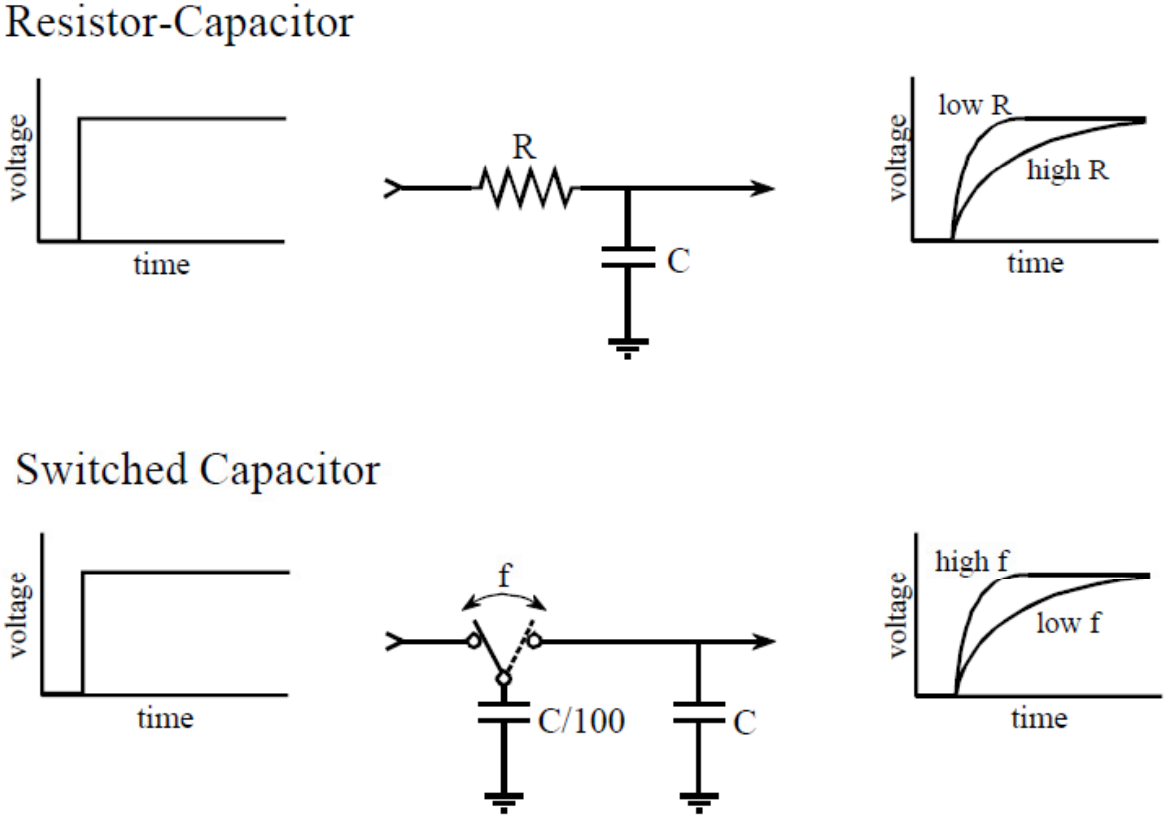


FIGURE 3-10
Switched capacitor filter operation. Switched capacitor filters use switches and capacitors to mimic resistors. As shown by the equivalent step responses, two capacitors and one switch can perform the same function as a resistor-capacitor network.

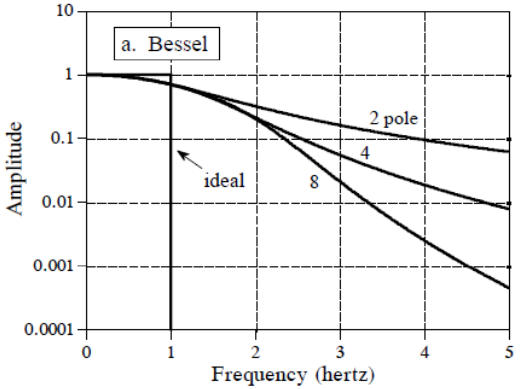
Analog Filters for Data Conversion

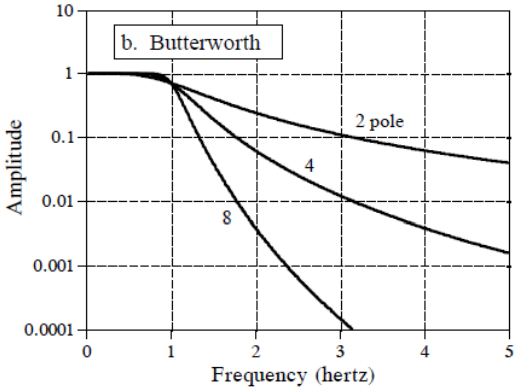
- In a resistor
  - Rate of charge transfer determined by its resistance
- In a switched capacitor
  - Rate of charge transfer determined by
    - Value of small capacitor
    - Switching frequency
  - Cut-off frequency of filter is directly proportional to clock frequency used to drive switches
    - Makes switched capacitor filter ideal for data acquisition systems that operate with more than one sampling rate

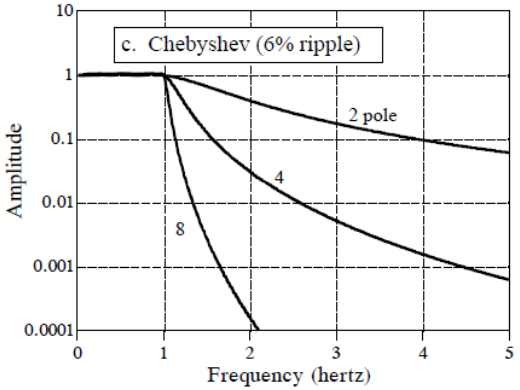
Analog Filters for Data Conversion

- Stopband
  - All frequencies above cutoff frequency
  - Low pass filter designed to block it
- Passband
  - All frequencies below cutoff frequency
  - Low pass filter designed to pass it
- Cutoff frequency sharpness
  - Chebyshev is best
    - Roll-off (drop in amplitude) as rapidly as possible
  - Next figure shows frequency response of different filters on a logarithmic scale
    - with one hertz cutoff frequency
    - Can be directly scaled to any cutoff frequency

Analog Filters for Data Conversion

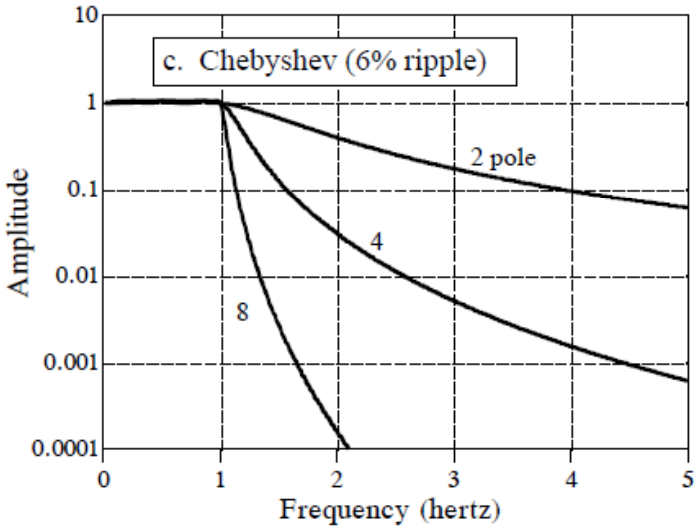






Analog Filters for Data Conversion

- Example (8 pole Chebyshev filter)
  - A 12 bit system – 10,000 samples/s – Frequencies above 5 kHz to be reduced in amplitude by a factor of 100
  - f_c = 1 Hz → attenuation of 100 at 1.35 Hz
  - Scaling to example → f_c must be set to 3.7 kHz
  - Results in band between 3.7 and 5 kHz being wasted
  - A frequency to alias into passband
    - Must be > 6300 Hz (or 1.7 * 3700)
    - Attenuation at 1.7 f_c = 1300
    - Much more adequate than 100



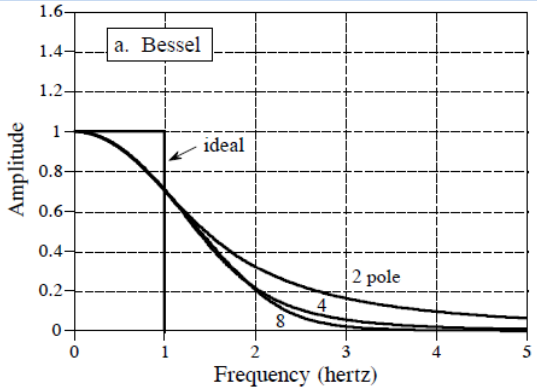
Analog Filters for Data Conversion

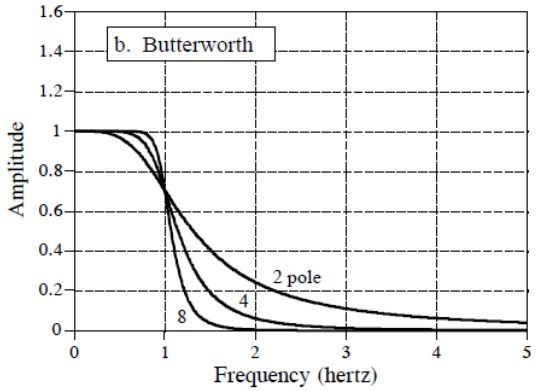
- Result of inadequate roll-off of analog filters
  - In most systems, frequency band between about 0.4 and 0.5 of sampling frequency is an unusable wasteland of filter roll-off and aliased signals

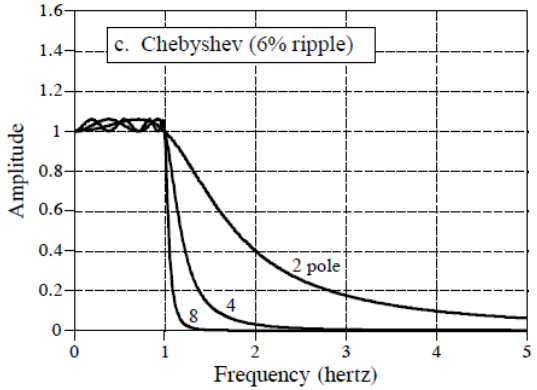
Analog Filters for Data Conversion

- Passband ripple
  - Wavy variations in amplitude of passed frequencies
  - Is seen in Chebyshev filter
    - Obtains its excellent roll-off by allowing this ripple
  - More passband ripple allowed in a filter → faster roll-off
  - Butterworth filter provides flattest passband
- Elliptic filter
  - Allows ripple in both passband and stopband

Analog Filters for Data Conversion



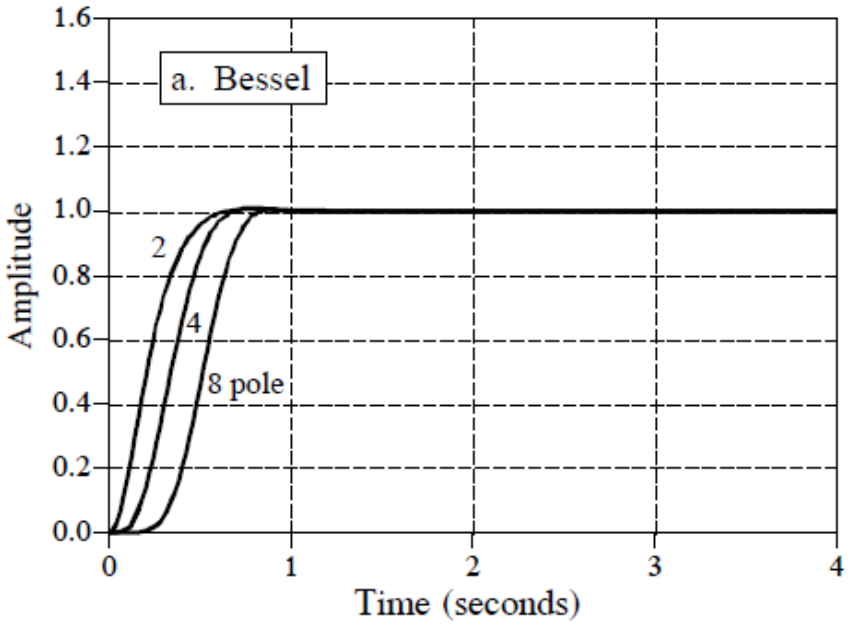


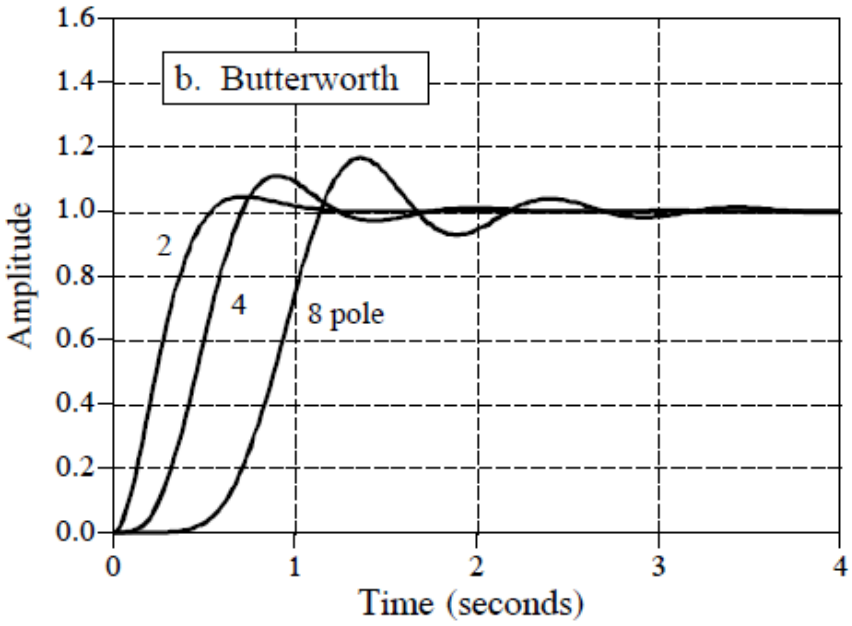


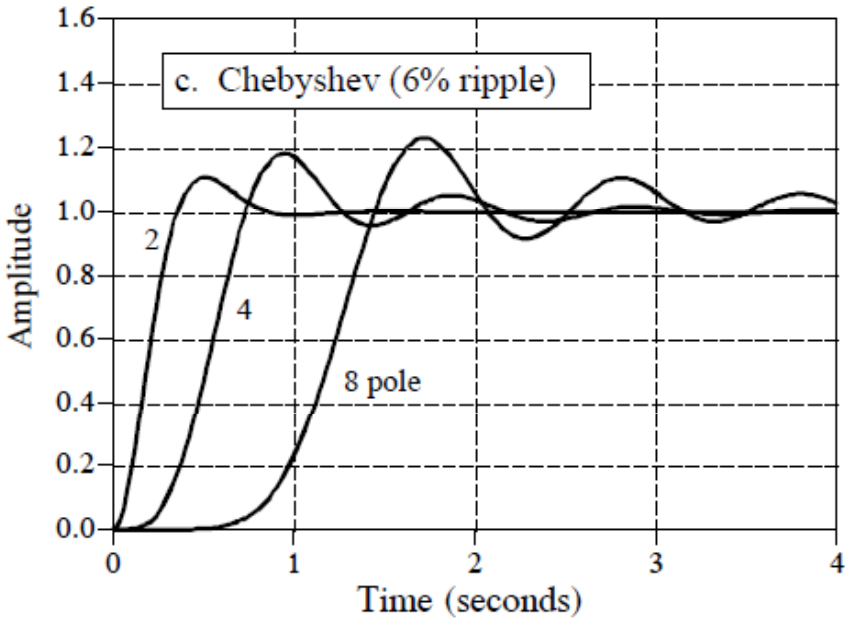
Analog Filters for Data Conversion

- Step response
  - How filter responds when input rapidly changes from one value to another
  - Butterworth and Chebyshev
    - Overshoot and show ringing
  - Bessel filter is optimal
  - Next figure shows step response
    - For filters with 1 Hz f_c
    - Can be scaled inversely for higher f_cs – e.g., a 1000 Hz f_c shows a step response in milliseconds

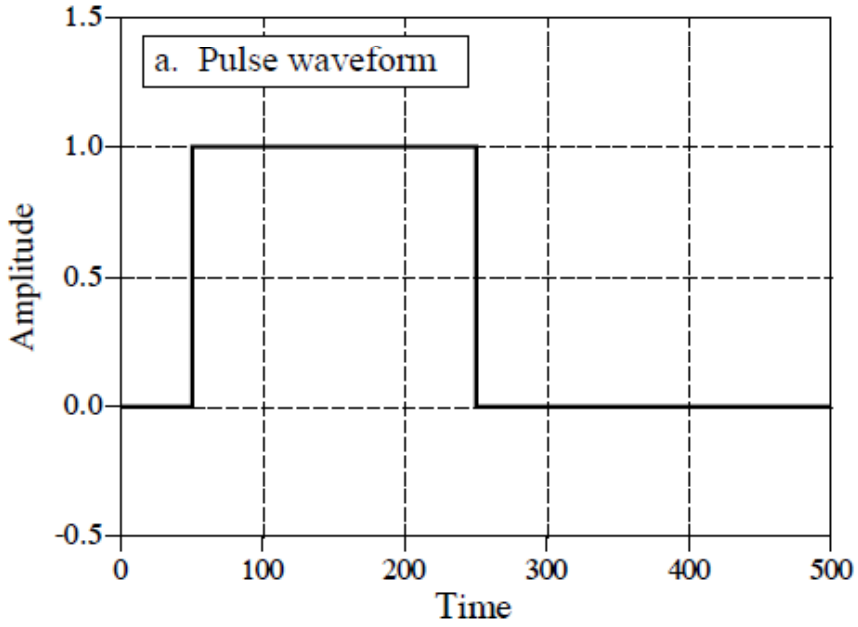
Analog Filters for Data Conversion

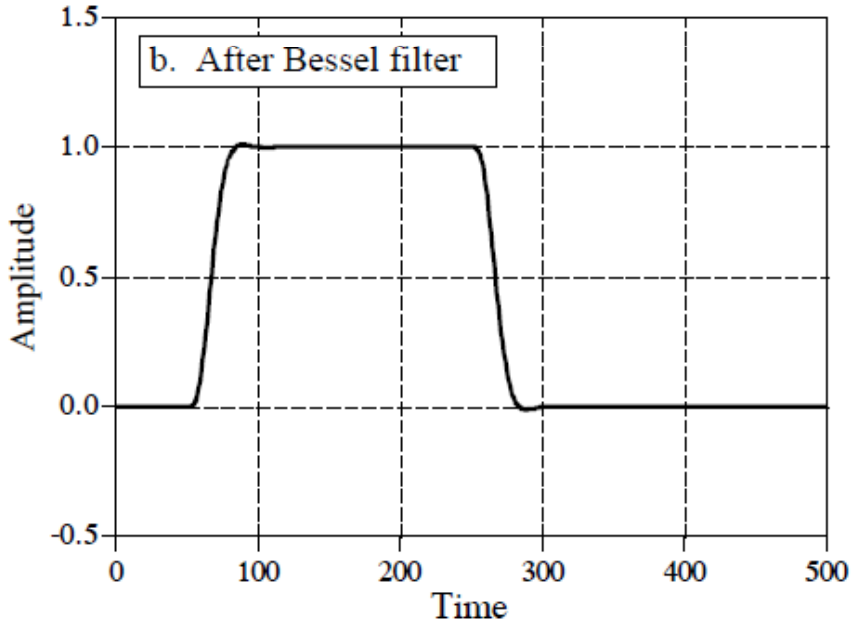


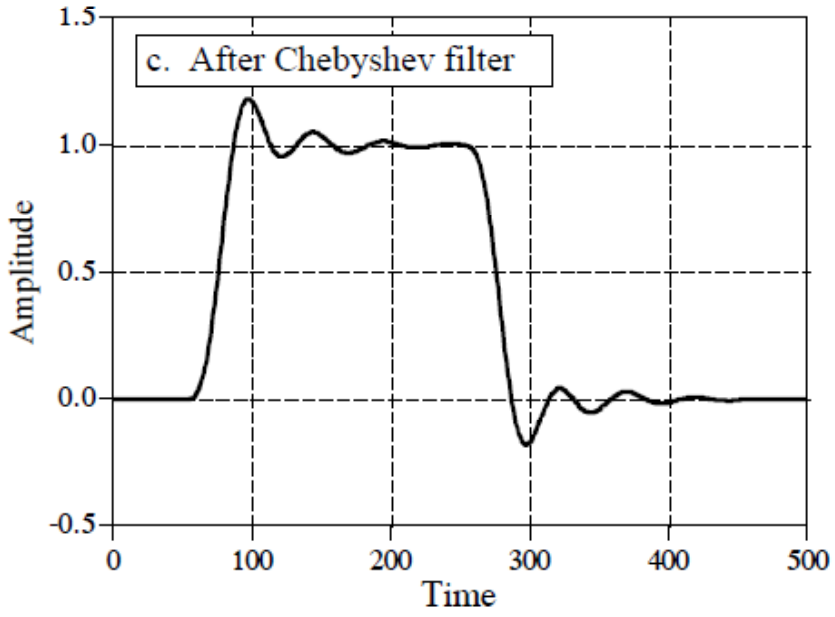




Analog Filters for Data Conversion







Selecting the Antialias Filter

- Characteristics of three classic filters

	Voltage gain at DC	Step Response			Frequency Response		
		Overshoot	Time to settle to 1%	Time to settle to 0.1%	Ripple in passband	Frequency for x100 attenuation	Frequency for x1000 attenuation
Bessel							
2 pole	1.27	0.4%	0.60	1.12	0%	12.74	40.4
4 pole	1.91	0.9%	0.66	1.20	0%	4.74	8.45
6 pole	2.87	0.7%	0.74	1.18	0%	3.65	5.43
8 pole	4.32	0.4%	0.80	1.16	0%	3.35	4.53
Butterworth							
2 pole	1.59	4.3%	1.06	1.66	0%	10.0	31.6
4 pole	2.58	10.9%	1.68	2.74	0%	3.17	5.62
6 pole	4.21	14.3%	2.74	3.92	0%	2.16	3.17
8 pole	6.84	16.4%	3.50	5.12	0%	1.78	2.38
Chebyshev							
2 pole	1.84	10.8%	1.10	1.62	6%	12.33	38.9
4 pole	4.21	18.2%	3.04	5.42	6%	2.59	4.47
6 pole	10.71	21.3%	5.86	10.4	6%	1.63	2.26
8 pole	28.58	23.0%	8.34	16.4	6%	1.34	1.66

Selecting the Antialias Filter

- Characteristics of filters
  - Each optimizes a particular parameter at expense of everything else
  - Chebyshev optimizes roll-off
  - Butterworth optimizes passband flatness
  - Bessel optimizes step response
- Selection of antialias filter
  - Depends on how information is represented in signals for processing
    - Time domain encoding
    - Frequency domain encoding

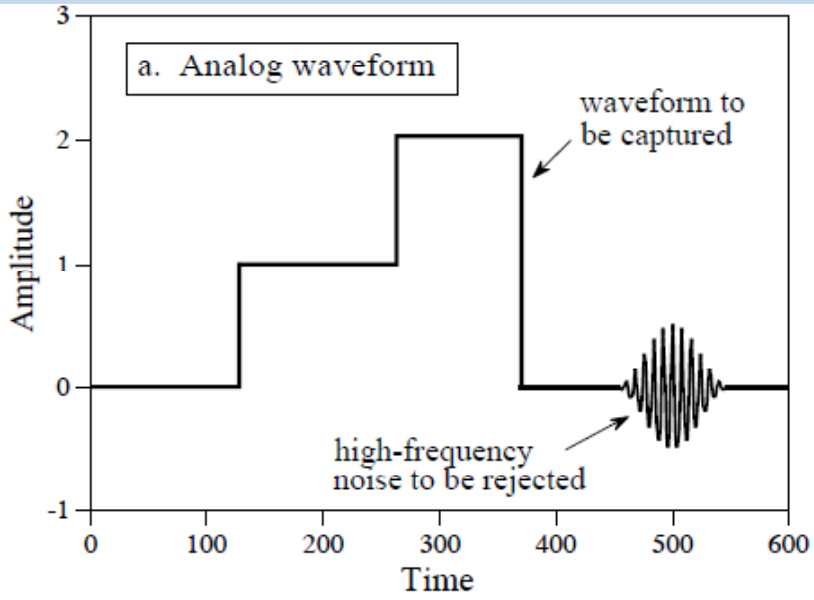
Selecting the Antialias Filter

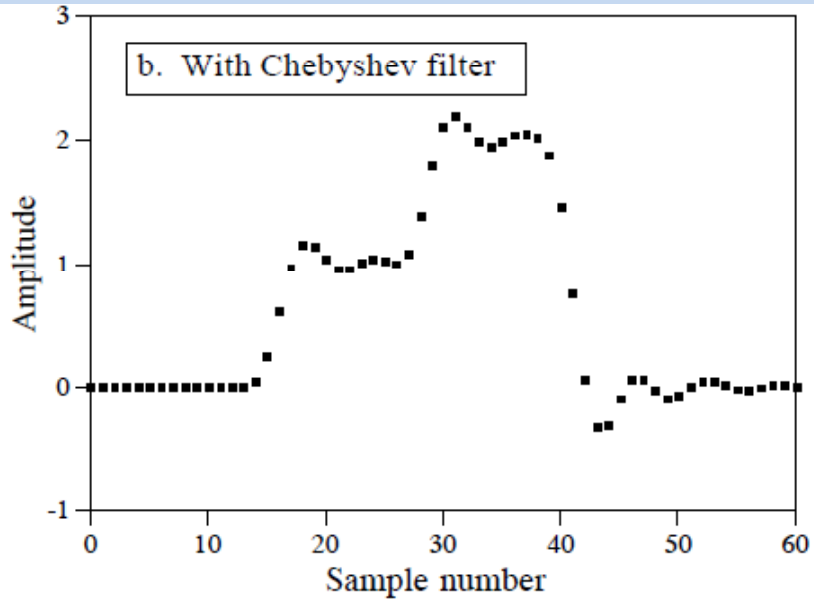
- Frequency domain encoding
  - Information contained in sinusoidal waves that combine to form signal
  - E.g., audio signals
    - Perceived sound depends on frequencies present, not on particular shape of waveform
    - Changing phase of sinusoids, while retaining frequency and amplitude → different waveform shape but sounds identical
  - Aliasing destroys info encoded in frequency domain
    - Use antialias filter with a sharp cutoff ,e.g., Chebyshev, Elliptic, or Butterworth
    - Encoded info not affected by nasty step response of these filters

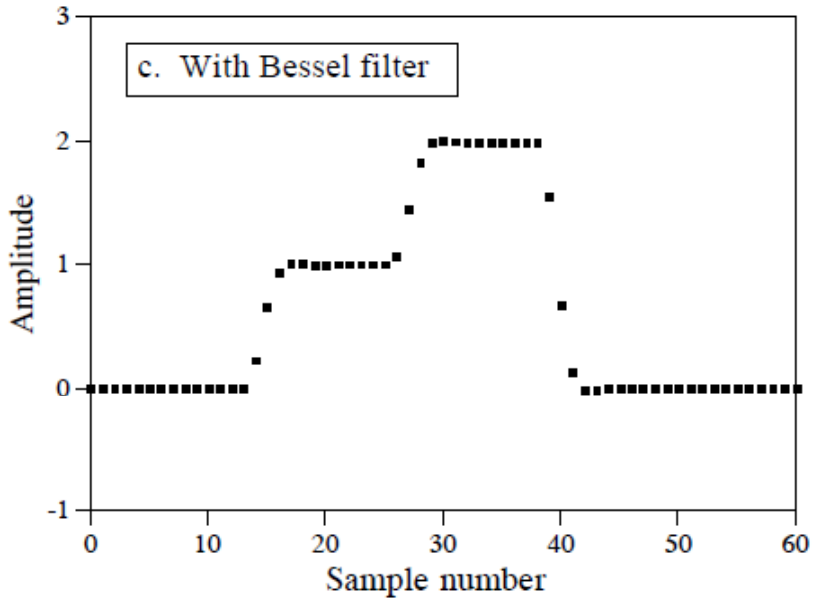
Selecting the Antialias Filter

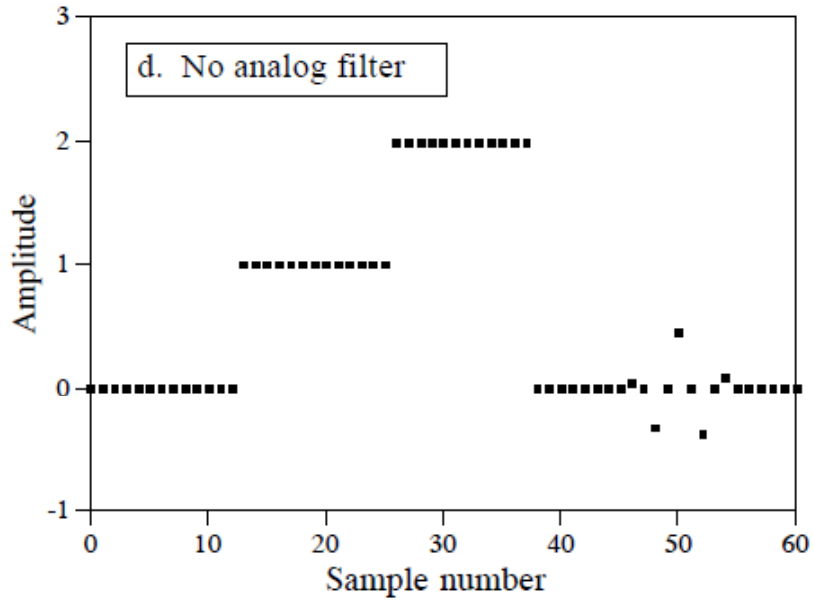
- Time domain encoding
  - Uses shape of waveform to store information
  - E.g., electrocardiogram, or images
  - Chebyshev filter severely distorts waveform
    - Should not be used
  - Bessel filter is best

Selecting the Antialias Filter









Multirate Data Conversion

- Trend in electronics
  - Replace analog circuitry with digital algorithms
- Example: design of a digital voice recorder
  - Speech between 100 and 3000 Hz
  - Pass analog signal through 8 pole Chebyshev at 3 kHz
  - Sample at 8 kHz
  - DAC reconstructs analog signal at 8 kHz
    - Zeroth order hold
  - Another Chebyshev filter at 3 kHz to produce final signal

Multirate Data Conversion

- Faster sampling is useful
  - Redesigning digital voice recorder example
    - 64 kHz sampling rate
    - Antialias filter needs to pass frequencies below 3 kHz while rejecting frequencies above 32 kHz
    - Similar simplification for reconstruction filter
  - Higher sampling rate allows 8 pole filters to be replaced with simple RC networks
  - Problem
    - Digital system swamped with data from higher sampling rate

Multirate Data Conversion

- Multirate techniques
  - Use more that one sampling rate in same system
  - For digital voice recorder example
    - Pass voice signal through a simple RC filter – sample at 64 kHz
    - 100 < desired band < 3000 Hz, 3 < unusable band < 32 kHz
    - Remove unusable band by digital low-pass filter at 3 kHz
    - Decimation: resample digital signal from 64 to 8 kHz by discarding every seven out of eight samples
    - Resulting digital data = data produced by aggressive analog filtering and direct 8 kHz sampling

Multirate Data Conversion

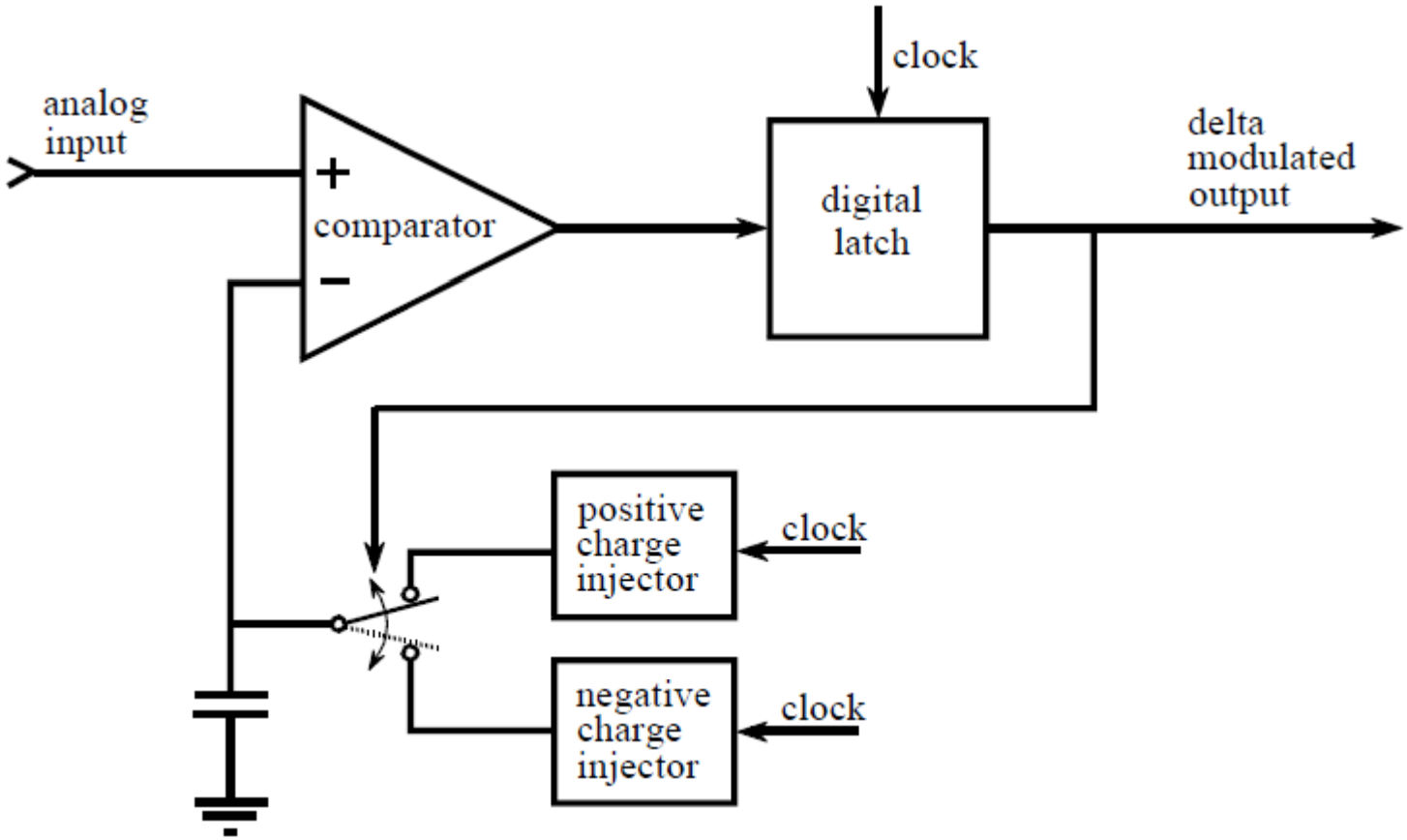
- Multirate techniques
  - Can be used in output portion of example system
    - Interpolation: 8 kHz data pulled from memory and converted to a 64 kHz sampling rate
    - Place seven samples with a value of zero between each of samples obtained from memory
    - Resulting signal is a digital impulse train
    - 100 < desired band < 3000 Hz, 3 < spectral duplications < 32 kHz
    - Everything above 3 kHz removed with digital low-pass filter
    - After conversion to an analog signal, a simple RC network used

Single Bit Data Conversion

- Single bit ADC and DAC
  - Used in telecommunications and music reproduction
  - Are multirate techniques
    - Higher sampling rate traded for a lower number of bits
  - Are mostly based on use of delta modulation circuit

Single Bit Data Conversion

- Block diagram of a delta modulation circuit

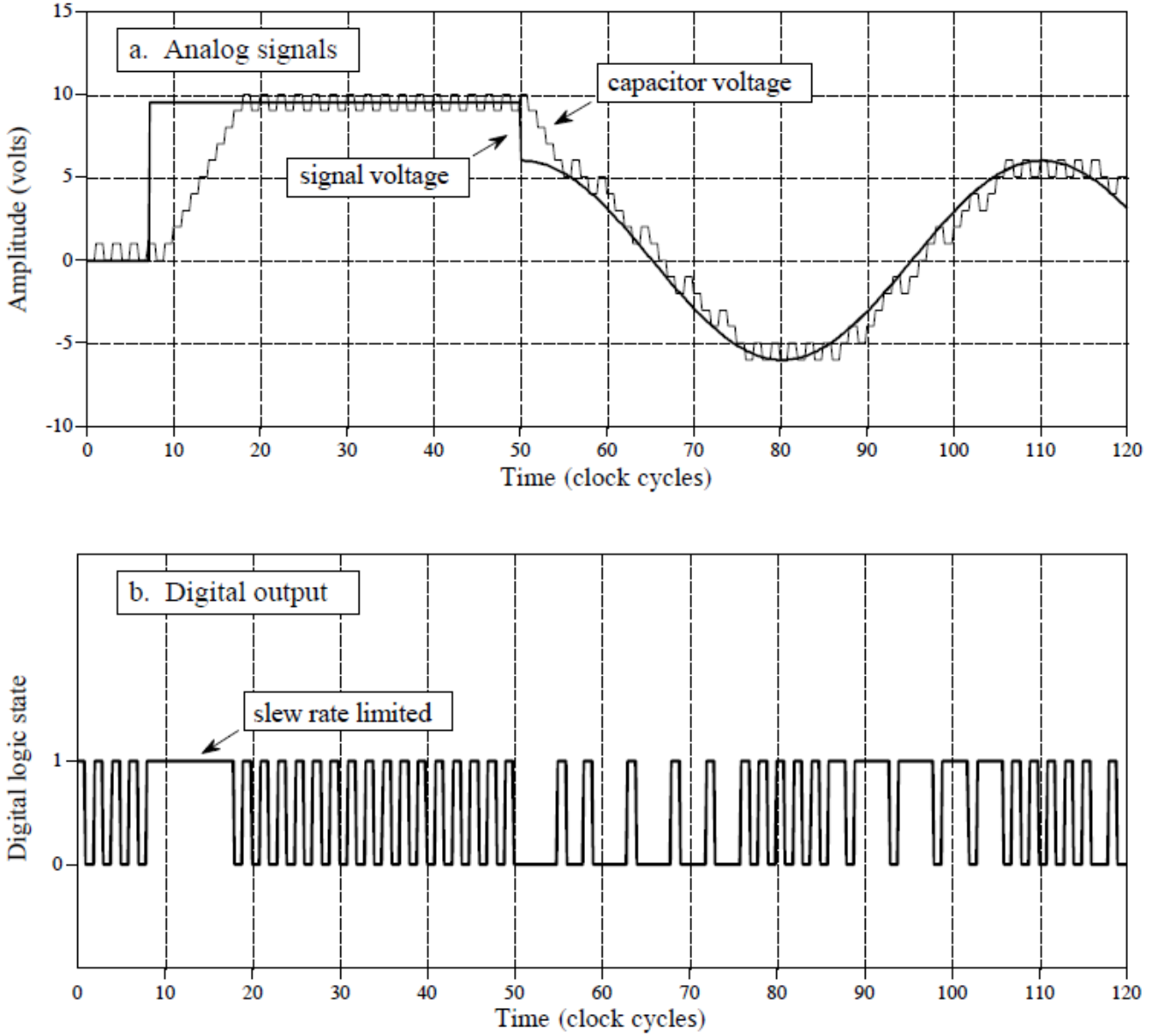


Single Bit Data Conversion

- In previous figure
  - Analog input (voice) – digital output (1s and 0s)
  - Comparator decides which has greater voltage
    - Incoming analog signal or capacitor voltage
    - This decision (1 or 0) applied to input of latch
  - Latch insures output is synchronized with clock
    - Defines sampling rate
  - Feedback loop takes digital output to drive an electronic switch
    - Output = 1 → capacitor connected to positive charge injector
    - Output = 0 → capacitor connected to negative charge injector – decreases voltage on capacitor by same fixed amount

Single Bit Data Conversion

- Example of signals produced by delta modulator



Single Bit Data Conversion

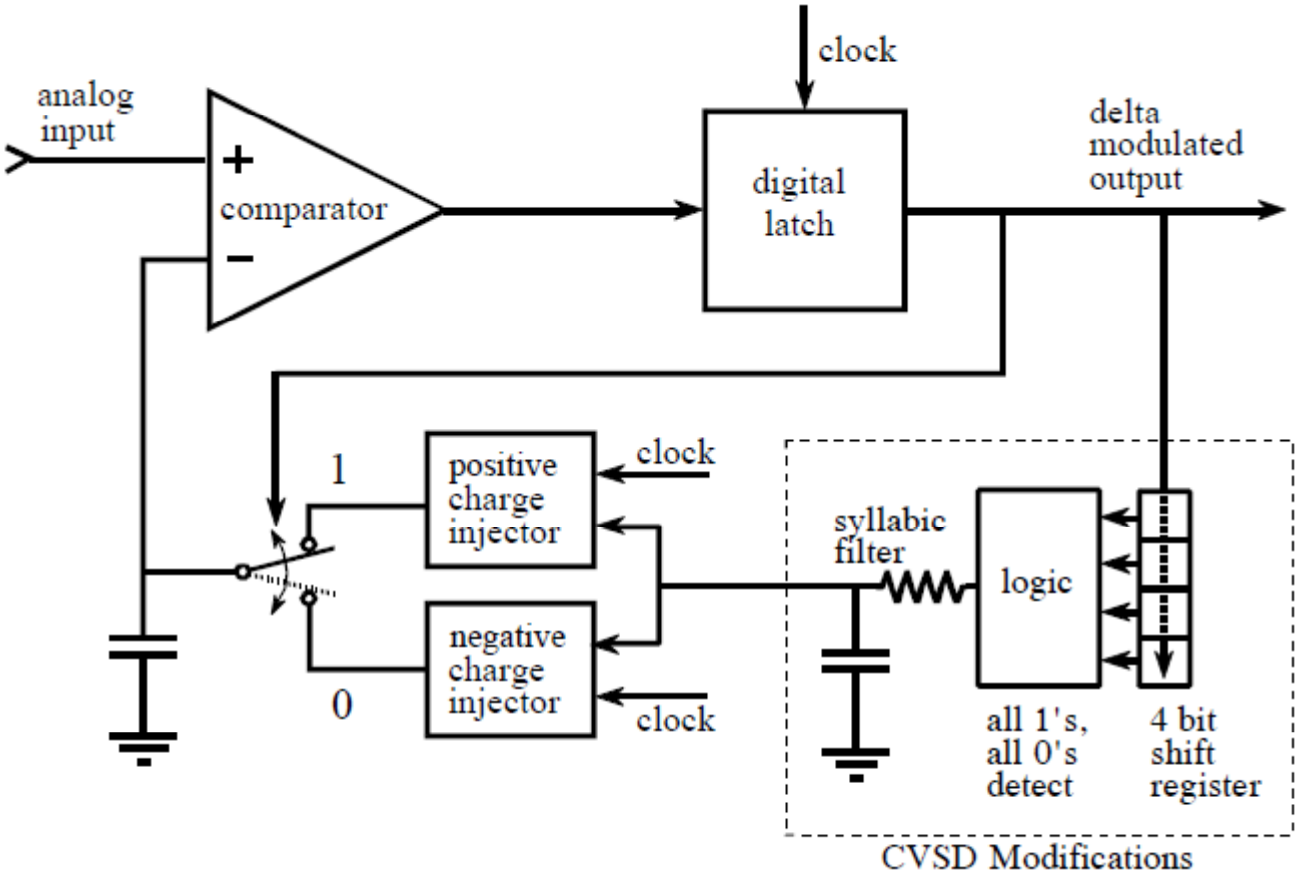
- In previous figure
  - At time = 0, analog input and capacitor voltage = 0
  - Slew rate
    - If input signal changes very rapidly, capacitor voltage changes at a constant rate (slew rate) until a match obtained
  - Analog input increasing
    - Output signal consists of more 1s than 0s
  - Analog input decreasing
    - Output signal consists of more 0s than 1s
  - Analog input constant
    - Digital output alternates equally between 0 and 1
  - Relative number of 1s versus 0s
    - Proportional to slope of analog input

Single Bit Data Conversion

- Advantage of delta modulator
  - All bits have same meaning in transmission or storage
    - Unlike serial format: start bit, LSB, …, MSB, stop bit
  - Circuit at receiver identical to feedback portion of transmitting circuit
    - Capacitor voltage reconstructs signal
- Limitation of delta modulator
  - Unavoidable tradeoff between
    - Maximum slew rate
    - Quantization size
    - Data rate
  - Max. slew rate and quantization size adjusted for voice
    - Data rate ends up in MHz range → too high

Single Bit Data Conversion

- CVSD modulation
  - Continuously Variable Slope Delta
  - A solution to problem of delta modulation



Single Bit Data Conversion

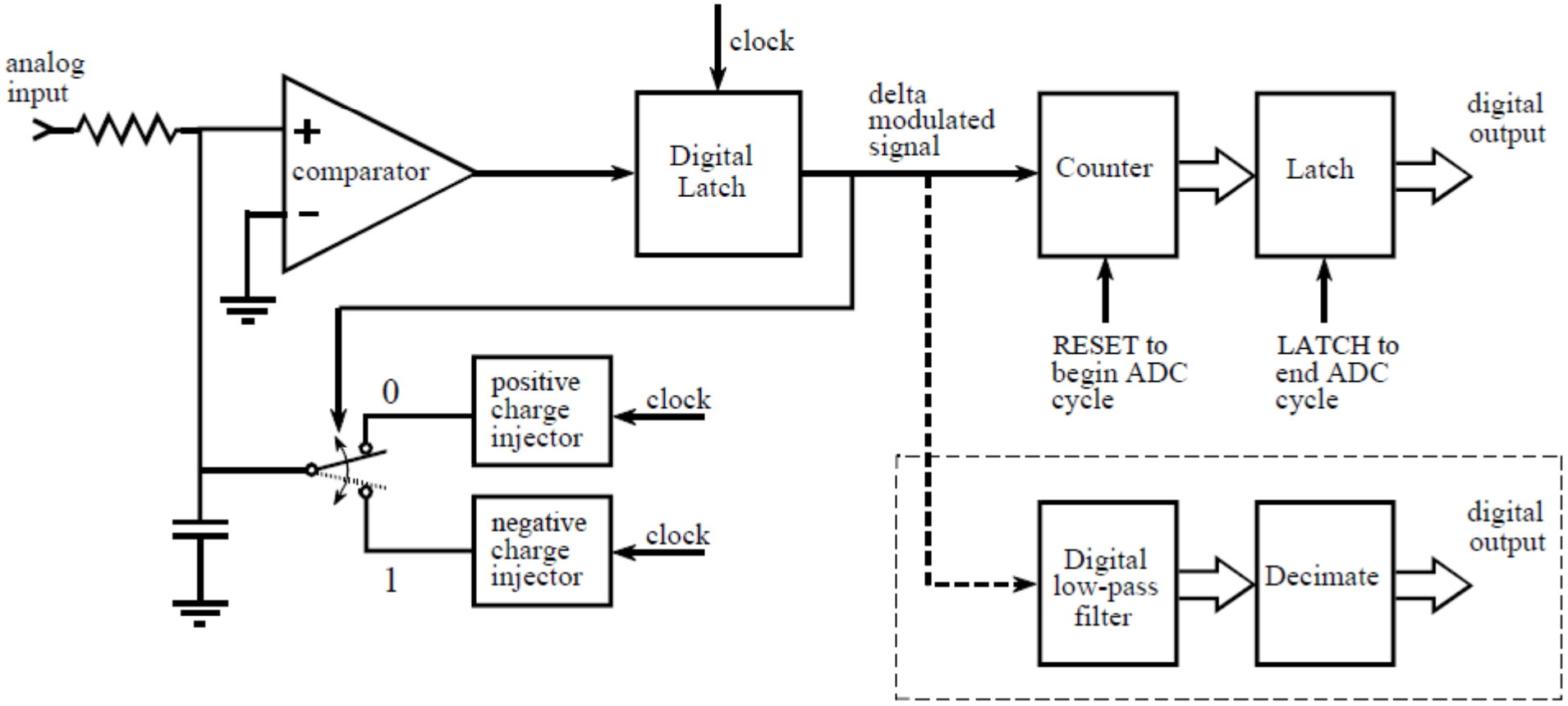
- In CVSD
  - Clock rate and quantization size set to acceptable values
    - E.g., 30 kHz, and 2000 levels
    - Results in terrible slew rate → corrected with additional circuit
  - Circuit is in slew rate limited condition
    - Last 4 bits all 1s or 0s
    - A shift register continually looks at last 4 bits
    - A logic circuit detects → produces an analog signal that increases level of charge produced by charge injectors
    - Syllabic filter allows step size to depend on how long circuit is in slew limited condition → step size gets larger and larger
    - At receiver, analog signal reconstructed by incorporating a syllabic filter identical to one in transmitter

Single Bit Data Conversion

- CVSD modulation
  - Great for encoding voice signals
  - Cannot be used for general purpose ADC
    - Digital data related to derivative of input signal
    - Changing step size is problematic
    - DC level of analog signal usually not captured in digital data

Single Bit Data Conversion

- Delta-sigma converter
  - Eliminates problems of CVSD by combining analog electronics with DSP algorithms



Single Bit Data Conversion

- In delta-sigma converter
  - Voltage on capacitor compared with ground potential
  - Feedback loop modified
    - Voltage on capacitor decreased when output = 1
    - Voltage on capacitor increased when output = 0
  - Input voltage positive
    - Digital output composed of more 1s than 0s
    - Excess number of 1s needed to generate negative charge that cancels with positive input signal
  - Input voltage negative
    - Digital output composed of more 0s than 1s
  - Input signal = 0
    - Equal number of 1s and 0s

Single Bit Data Conversion

- In delta-sigma converter
  - Relative number of 1s and 0s in output
    - Related to level of input voltage, not slope → simpler
  - Example: forming a 12 bit ADC
    - Feed digital output into a counter
    - Count number of 1s over 4096 clock cycles
    - Digital number 4095 corresponds to maximum positive input
    - Digital number 0 corresponds to maximum negative input
    - 2048 corresponds to an input voltage of 0

Single Bit Data Conversion

- In delta-sigma converter
  - To transform 1s and 0s back into analog signal
    - A simple analog low-pass filter required
    - High and low voltages corresponding to 1s and 0s average out to form correct analog voltage
    - E.g., suppose 1s and 0s represented by 5 and 0 volts → if 80% of bits are 1s and 20% 0s, output of low-pass filter = 4 volts
  - A way to replace counter in delta-sigma ADC circuit
    - Binary signal passed through digital low-pass filter and then decimated
    - E.g., change each of 1s and 0s into a 12 bit sample: 1s → 4095, 0s → 0, use a digital low-pass filter and then decimate