Introduction to Sampling

What is SAMPLING?

- Process by which an analog signal is measured, often millions of times per second for video, in order to convert the analog signal to digital.

Digital Data Acquisition

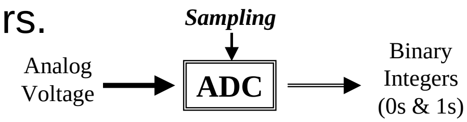
- Data Representation - *Digital vs. Analog*
- Analog-to-Digital Conversion
- Number Systems
  - Binary Numbers
  - Binary Arithmetic
- Sampling & Aliasing

Data Representation - *Digital vs. Analog*

- <u>Digital</u>:
  1s and 0s, $(1001\ 1011)_2$
  - *Advantages*
  - *Disadvantages*
- <u>Analog</u>:
  3.141592687..., 1/3
  - *Advantages*
  - *Disadvantages*

Analog-to-Digital Conversion

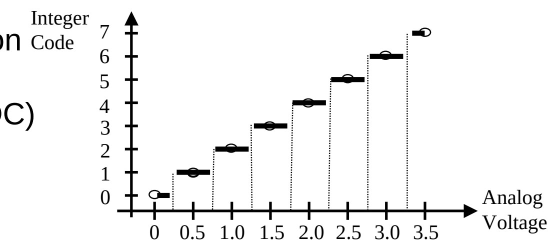
- Converts analog voltages to binary integers.



Analog-to-Digital Conversion

- **ADC calibration**

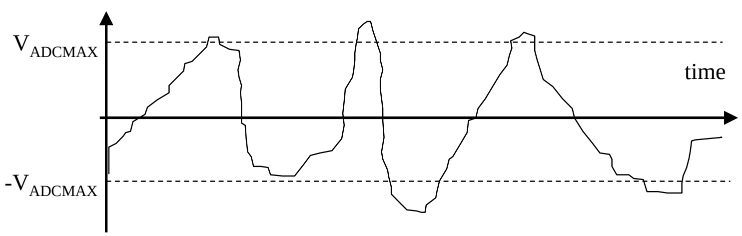
Calibration Curve (3 bit ADC)



Analog-to-Digital Conversion

- Input Range
  - Unipolar: (0, V_{ADCMAX})
  - Bipolar: ($-V_{ADCMAX}$, $+V_{ADCMAX}$) (Nominal Range)
  - **Clipping**:

If $|V_{IN}| > |V_{ADCMAX}|$, then $|V_{OUT}| = |V_{ADCMAX}|$

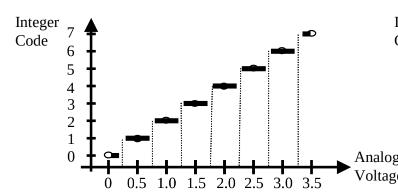


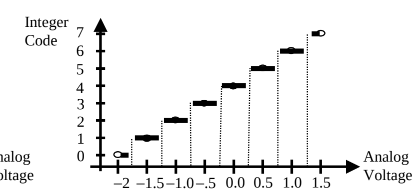
Analog-to-Digital Conversion

- Quantization Interval (Q)
  - n bit ADC, the input range is divided into 2^n-1 intervals.

$$Q = \frac{V_{ADC\mathrm{MAX}} - V_{ADC\,\min}}{2^n - 1}$$

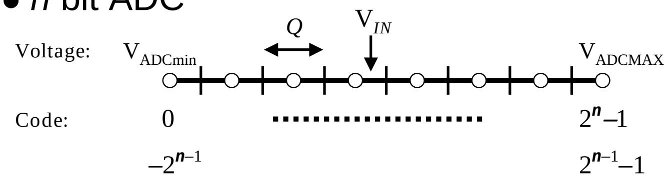
  - 3 bit ADC:





Analog-to-Digital Conversion

- Voltage to Integer Code
  - *n* bit ADC



Positive Coding:

$$\text{Code} = \text{Round}\left[\frac{V_{IN} - V_{ADC\,min}}{Q}\right]$$

Positive and Negative Coding:

$$\text{Code} = \text{Round}\left[\frac{V_{IN}}{Q}\right]$$

Analog-to-Digital Conversion

- Voltage to Integer Code **(cont.)**

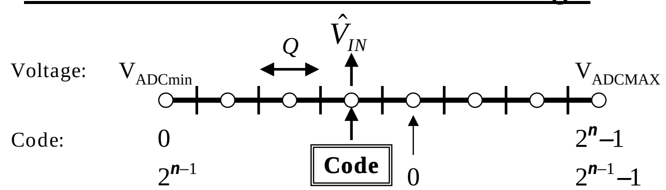
Ex: 3 bit ADC, V_{ADCmin} = –2 V and V_{ADCMAX} = 1.5 V

V_{IN} = 1.25 V

Q: Find the integer code for V_{IN} using only positive integers and using both positive and negative integers.

Analog-to-Digital Conversion

- Convert Code to Estimated Voltage



$$\Rightarrow \quad \hat{V}_{IN} = \text{Code} \times Q + V_{offset}$$

V_{offset}: voltage corresponding to code 0

Analog-to-Digital Conversion

- Convert Code to Estimated Voltage **(cont.)**

Ex: 3 bit ADC, V_{ADCmin} = 0 V and V_{ADCMAX} = 3.5 V

Code = 2

Q: What is the estimated input voltage V_{IN}?

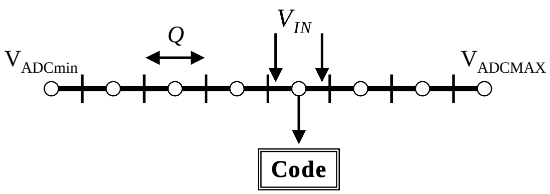
Ex: 3 bit ADC, V_{ADCmin} = –2 V and V_{ADCMAX} = 1.5 V

Code = 2 (Bipolar Coding)

Q: What is the estimated input voltage V_{IN}?

Analog-to-Digital Conversion

Maximum Quantization Error



$$\text{Any } V_{IN} \in \left[\hat{V}_{IN} - \frac{Q}{2},\ \hat{V}_{IN} + \frac{Q}{2}\right] \text{ will be coded to } \hat{V}_{IN}$$

- **Maximum Quantization Error = *Q*/2**

Binary Numbers

Binary Representation of Integers

$$(\overbrace{\underset{\text{MSB}}{d_{(n-1)}}\ d_{(n-2)}\ d_{(n-3)} \cdots d_2\ d_1\ \underbrace{d_0}_{\text{LSB}}}^{n \text{ bits}})_2$$

$$= d_{(n-1)} \times 2^{(n-1)} + d_{(n-2)} \times 2^{(n-2)} + d_{(n-3)} \times 2^{(n-3)} + \cdots + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times \underbrace{2^0}_{1}$$

where d_i's = 0 or 1.

- *n* bits binary: represents 2^n integers.
  Ex: 4 bit Binary: 2^4 = 16 integers, (0, 1, 2, …, 15).
- MSB: Most Significant Bit
- LSB: Least Significant Bit

Binary Numbers - Coding

Positive Integers

Straight Binary

Decimal $(10^1 10^0)$	Binary $(2^3\ 2^2\ 2^1\ 2^0)_2$
0 0	**0 0 0 0**
0 1	**0 0 0 1**
0 2	**0 0 1 0**
0 3	**0 0 1 1**
0 4	**0 1 0 0**
0 5	**0 1 0 1**
0 6	**0 1 1 0**
0 7	**0 1 1 1**

Binary Numbers - Coding

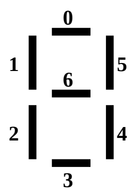
Straight Binary **(cont.)**

Decimal	Binary
$(132)_{10}$	**$(1011\ \ 0100)_2$**

Q: What are the decimal integers represented by an n bit straight binary number?

Binary Number - Coding

- <u>Binary Coded Decimal</u> (BCD)
  - Used a lot in LED digital display.
  - Each digit of the decimal number is separately coded into binary number.



0 to 9, need 4 bits.

	0	1	2	3	4	5	6
0000	*	*	*	*	*	*	
0001					*	*	
0010	*		*	*		*	*
0011	*			*	*	*	*

Ex: $(3\ 6\ 4)_{10} = (0011\ \ 0110\ \ 0100)_{BCD}$

Binary Numbers - Coding

Negative Integers

- <u>Sign Bit Convention</u>
  - Uses the MSB as the *sign bit:*

    MSB = 0 Positive
    MSB = 1 Negative
  - Have two zeros.

Ex: $(\underbrace{0}\underbrace{1\,0\,1})_2 =$

$(\underbrace{1}\underbrace{1\,1\,0})_2 =$

MSB (***Sign Bit***)

Binary $(2^2 2^1 2^0)_2$	**Decimal**
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	-0
1 0 1	-1
1 1 0	-2
1 1 1	-3

Binary Numbers - Coding

- <u>2's Complement</u>
  - Positive Numbers:

    0 … $(2^{(n-1)}-1)$
  - Negative Numbers are represented by their 2's complements:

$$6\ \{\text{4 bit straight binary}\} \Rightarrow (0110)_2$$
$$\{\text{Reverse 0s and 1s}\} \mapsto (1001)_2$$
$$\{\text{Add 1}\} \mapsto \underbrace{(1010)_2}_{\text{2's complement}}$$
$$\mapsto -6$$

Binary $(2^2 2^1 2^0)_2$	**Decimal**
0 1 1	3
0 1 0	2
0 0 1	1
0 0 0 ⟷	0
1 1 1	-1
1 1 0	-2
1 0 1	-3
1 0 0	-4

MSB (***Sign Bit***)

Binary Numbers - Coding

- <u>2's Complement</u> **(cont.)**
  - Number of bits is important.
  - *n* bit 2's complement numbers can represent integers from:

$$-\left(2^{(n-1)}\right) \leftarrow 0 \rightarrow \left(2^{(n-1)}-1\right)$$

  - Quick way of getting 2's complement numbers:

$$\text{8 bit numbers:}\ -84 \mapsto 84 = (0101\ \ 0100)_2$$

Q: What is the largest positive integer and the smallest negative integer that a 5 bit 2's complement number can represent?

Q: What are the 8 bit 2's complement representations of 81 and –102?

Binary Arithmetic

- Addition
  - 0 + 0 = 0
  - 0 + 1 = 1, 1 + 0 = 1
  - 1 + 1 = 0, carry 1

$$\begin{array}{r} 4 \\ +6 \\ \hline 10 \end{array} \qquad \begin{array}{r} (0100)_2 \\ +(0110)_2 \\ \hline (1010)_2 \end{array}$$

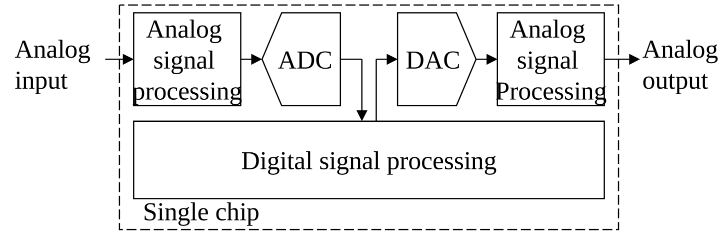
Ex:

(8 bit 2's complement)

32 + 75 =

24 - 32 =

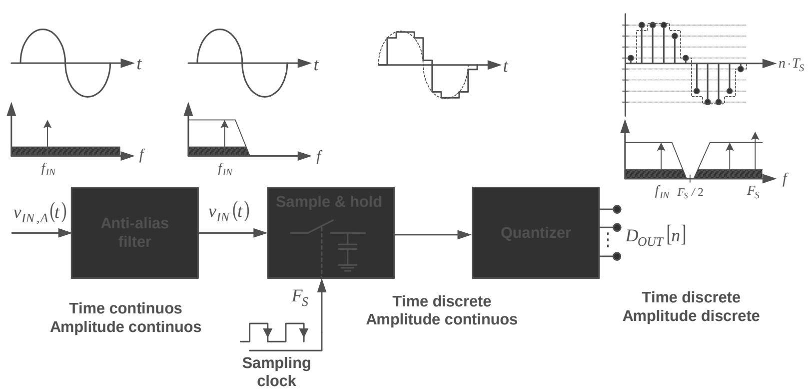
Why A/D-conversion?



- Signals are analog by nature
- ADC necessary for DSP
- Digital signal processing provides:
  - Close to infinite SNR
  - Low system cost
  - Repetitive system

- ADC bottle necks:
  - Dynamic range
  - Conversion speed
  - Power consumption

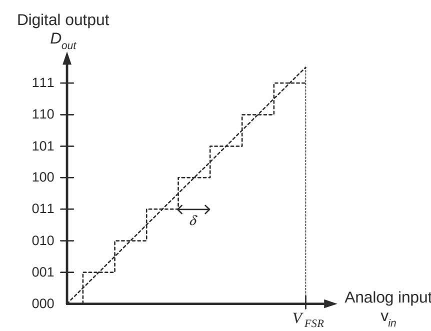
A/D-converter basics

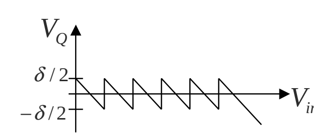


$$D_{OUT}^{ideal}[n] = G_{ideal} \cdot v_{IN}(n \cdot Ts) + q(n)$$

$$D_{OUT}^{real}[n] = G_{ideal} \cdot (1+\varepsilon) \cdot v_{IN}(n \cdot Ts) + q(n) + e_{offset}(n) + e_{noise}(n) + e_{jitter}(n) + e_{distortion}(n)$$

Quantization noise





•N-bit converter: $\delta = \dfrac{V_{FSR}}{2^N}$

Quantization noise (2)

- Noise energy:

$$V_{Q(RMS)} = \sqrt{\frac{1}{\delta}\int_{-\delta/2}^{\delta/2} V_Q^2 \, dV_Q} = \sqrt{\frac{\delta^2}{12}}$$

- Signal energy:

$$V_{in(RMS)} = \frac{\delta \cdot 2^N}{2\sqrt{2}}$$

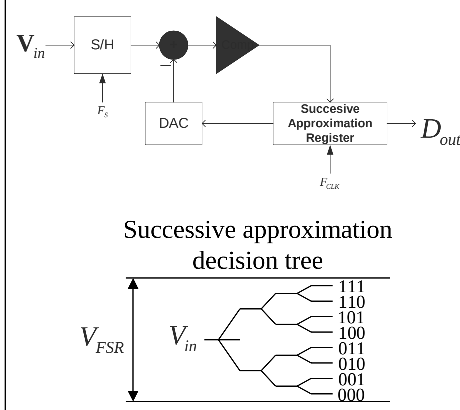
- SNR for ideal ADC:

$$SNR = 20\log\left(\frac{V_{in(RMS)}}{V_{Q(RMS)}}\right)$$

$$SNR = 20\log\left(2^N \cdot \sqrt{\frac{3}{2}}\right)$$

$$SNR = 6.02 \times N + 1.76[dB]$$

Successive approximation ADC



- N clock cycles conversion

  $F_{CLK} = F_S \cdot N$

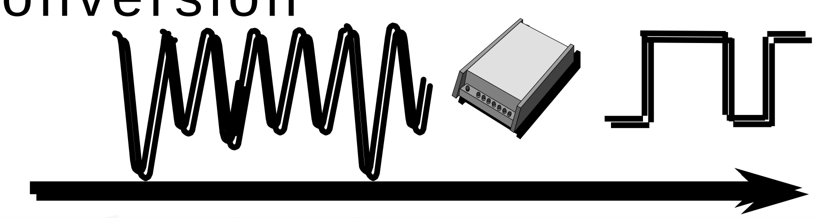
+ High resolution (selfcalibration)
+ Easy implementation

– Distortion limited by component matching
– Need high frequency clock

  $T_{Conversion} = N \cdot T_{DAC\_settling}$

– Low speed

4 Steps of Analog / Digital Conversion



Antialiasing (filters)

Sampling (MHz)

Quantizing (bits)

Encoding

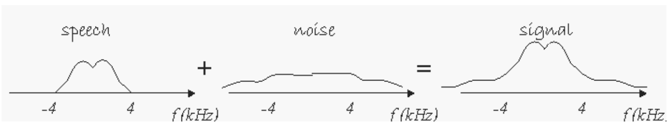
Step 1: Antialiasing

- **Sometimes an electronic signal will contain a range of frequencies that is greater than the range of frequencies contained in the information-bearing signal.**

For Example:

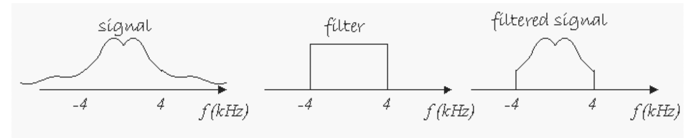
Step 1: Antialiasing

- Most information in a speech signal is contained in frequencies below 4 kHz, but noise and other factors may introduce frequency components greater than 4 kHz into an electronic speech signal.

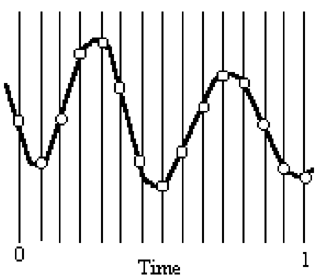


Step 1: Antialiasing

- **A pre-filter is used to remove the unwanted part of the signal.**

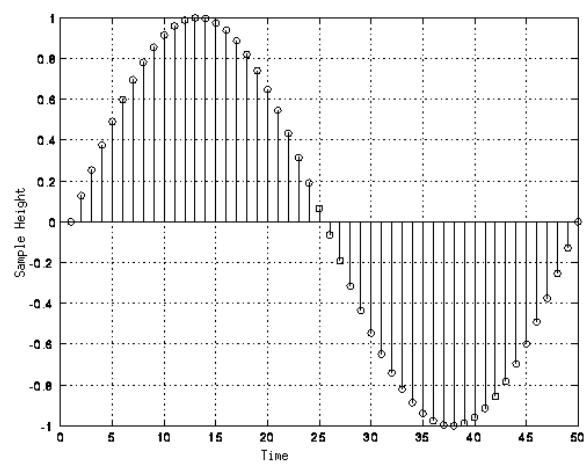


Step 2: Sampling



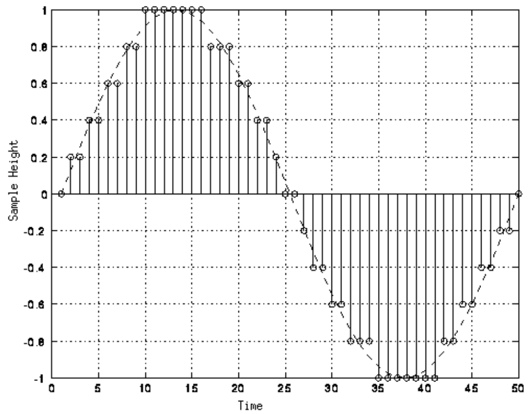
- Value of the analog signal is read at evenly spaced time intervals.
- Sample rate (frequency) is measured in megahertz.
- 1 mHz=1,000,000 cps.
- (Cycles per second).

Step 2: Sampling



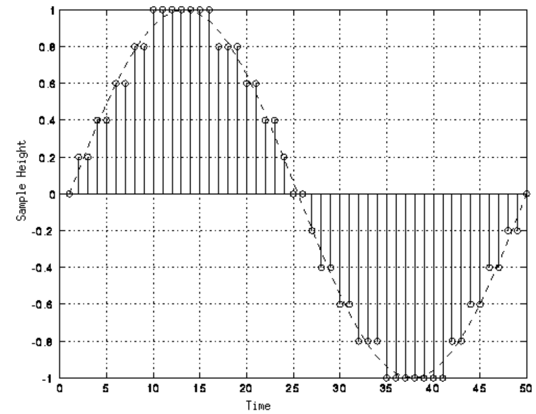
Step 3: Quantization

- The digital signal is defined only at the points at which it is sampled.



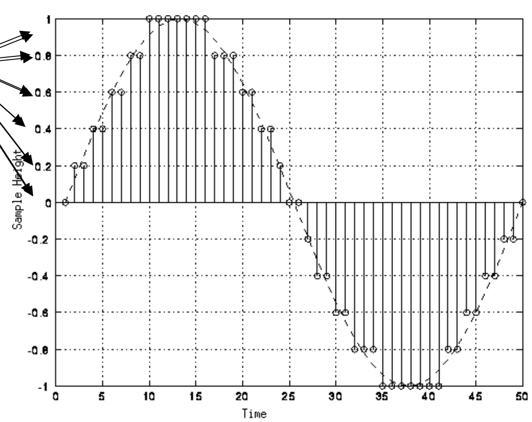
Step 3: Quantization

- The height of each vertical bar can take on only certain values, shown by horizontal dashed lines, which are sometimes higher and sometimes lower than the original signal, indicated by the dashed curve.



Step 3: Quantization

- If the graphic has 11 quantization levels, how many bits are needed to encode each sample?



Step 3: Quantization

- 4 bits… why?
- 1 bit would allow up to 2 levels
- 2 bits would allow up to 4 levels
- 3 bits would allow up to 8 levels
- 4 bits would allow up to 16 levels

Step 3: Quantization

- The difference between a quantized representation and an original analog signal is called the *quantization noise*.
- The more bits for quantization of a signal, the more closely the original signal is reproduced.

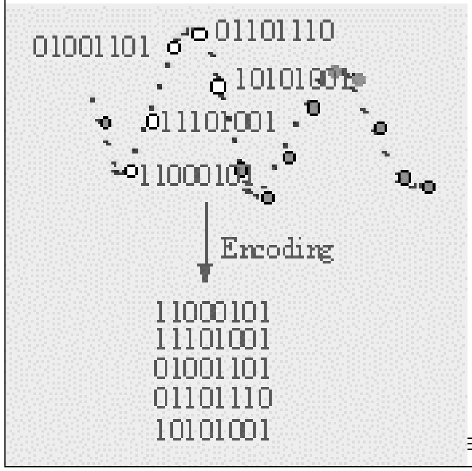
Step 3: Quantization

- Using higher sampling frequency and more bits for quantization will produce better quality digital video and audio.
- But for the same length of video and audio, the file size will be much larger than the low quality signal.

Step 3: Quantization

- The number of bits available to describe sampling values determines the resolution or accuracy of quantization.
- For example, if you have 8-bit analog to digital converters, the varying analog voltage must be quantized to 1 of 256 discrete values;
- a 16-bit converter has 65,536 values.

Step 4: Encoding



- Conversion of data into machine readable format.

Nyquist Theorem

Nyquist Theorem

- A theorem, developed by Harry Nyquist, which states that an analog signal waveform may be uniquely reconstructed, without error, from samples taken at equal time intervals.

Nyquist Theorem

- The sampling rate must be equal to, or greater than, twice the highest frequency component in the analog signal.

Nyquist Theorem

- Stated differently:
- The highest frequency which can be accurately represented is one-half of the sampling rate.

Error

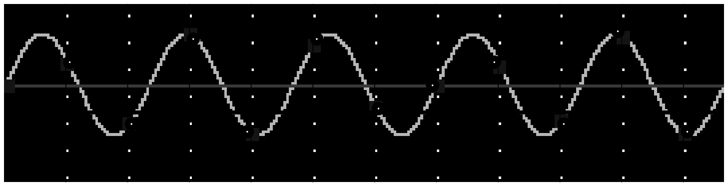
- Sampling an analog signal can introduce ERROR.
- ERROR is the difference between a computed, estimated, or measured value and the true, specified, or theoretically correct value.

Nyquist Theorem

- By sampling at TWICE the highest frequency:
  - One number can describe the positive transition, and…
  - One number can describe the negative transition of a single cycle.

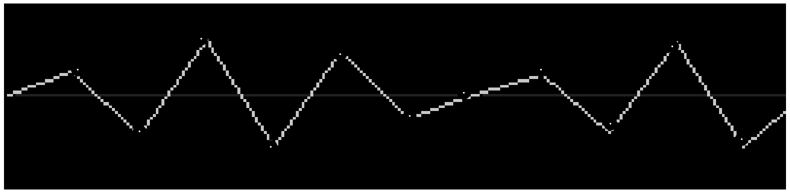
Nyquist Theorem

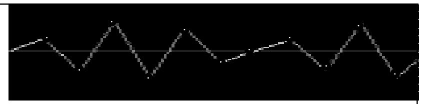
The vertical lines are sample intervals, and the white dots are the crossing points - the actual samples taken by the conversion process.



Nyquist Theorem

The sampling rate was below the Nyquist frequency, so the reconstructed waveform does not accurately reproduce the original:





Nyquist Theorem

This under-sampling results in **aliasing** which shows up as noise in digitized sound.

To correct the aliasing, A/D converters use lowpass filters to remove all signals above the Nyquist frequency.

To eliminate aliasing and to get high-fidelity sound, use a high sample rate.

Frequency aliasing

- When the highest frequency of the signal F_{input} is greater than half the sampling ($F_{sampling}/2$).
- E.g. F_{input} =20KHz, $F_{sampling}$ must be over 40KHz.
- Remedy: Use a low pass filter to cut off the input high frequency content before ADC sampling.

upper => sampling 6 times per cycle(fs=6f);
middle => sampling 3 times per cycle(fs=3f);

lower=> sampling 6 times in 5 cycles, from[1]

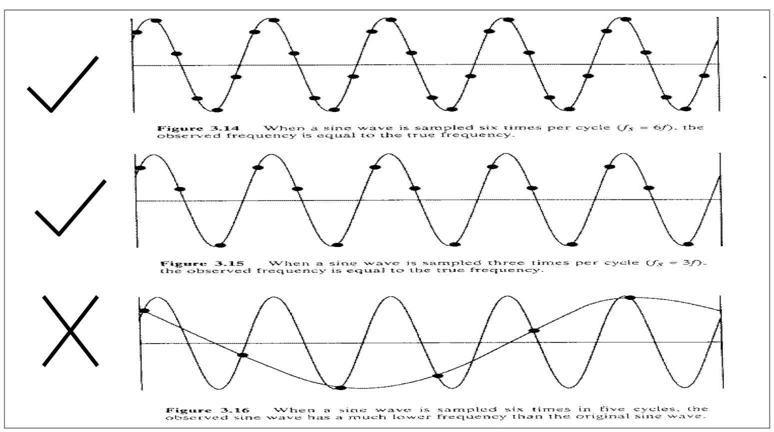


Figure 3.14 When a sine wave is sampled six times per cycle ($f_s = 6f$), the observed frequency is equal to the true frequency.

Figure 3.15 When a sine wave is sampled three times per cycle ($f_s = 3f$), the observed frequency is equal to the true frequency.

Figure 3.16 When a sine wave is sampled six times in five cycles, the observed sine wave has a much lower frequency than the original sine wave.

Method to reduce aliasing noise

Input voltage = V

Use low pass filter to remove high frequency before sampling