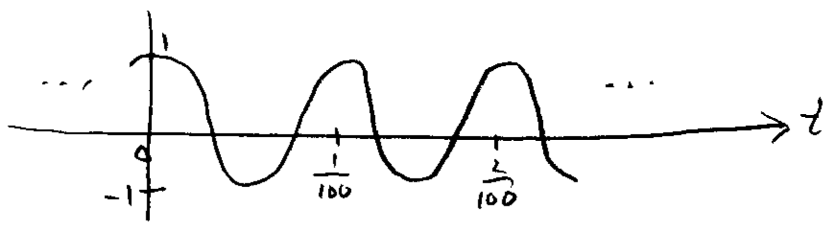
Getting Started

Sampled (Discrete-Time) Signals

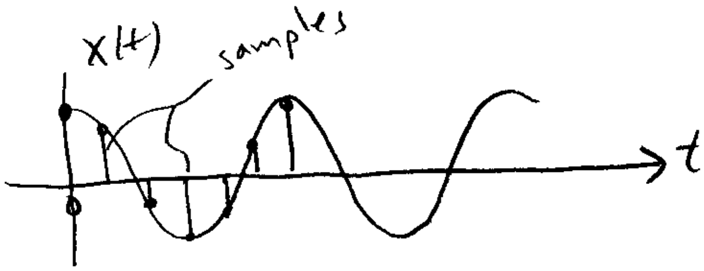
You are used to seeing and working with continuous-time (CT) signals. For example,

$$x(t) = \cos(2\pi(100)t)$$



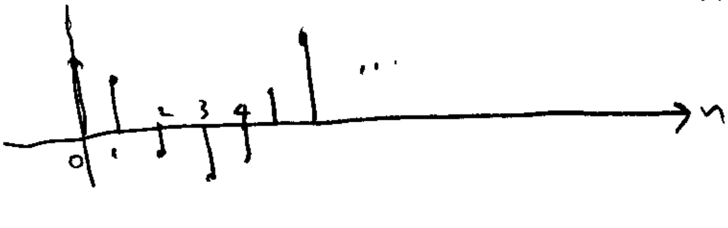
In Digital Signal Processing we focus on discrete-time (DT) signals.

$$x(t) = \cos(2\pi(100)t)$$



$x[n]$ $\quad$ n = sample index, acts like time

n = integer

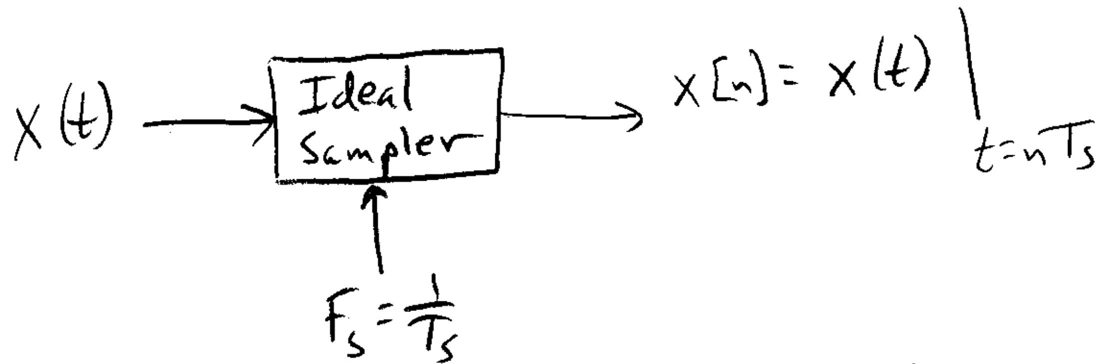


Q: Why do we focus on DT signals?

A: We can't use computers to process C.T. signals, but we can use them for DT signals.

Q: How do we get a D.T. signal from a C.T. signal?

A: Sampling



Ideal means the samples are infinite precision. That is to say, the number of bits/sample, B, is infinity

Notice that in sampling, we lose information. We don't keep what happens between the samples.

What are some consequences of this loss of information?

Consider an example.

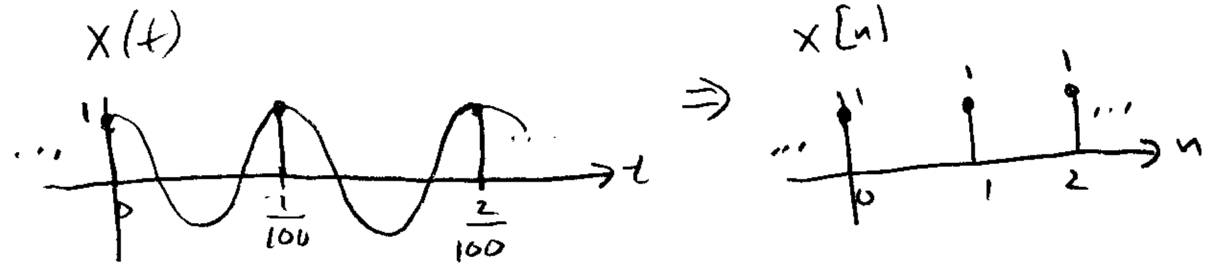
$$x(t) = \cos\left(2\pi(100)t\right) \quad , \text{ period} = T_s = \frac{1}{100} \text{ sec.}$$

Now sample at exactly this period.

$$x[n] = x(t)\Big|_{t = nT_s = \frac{n}{100}}$$

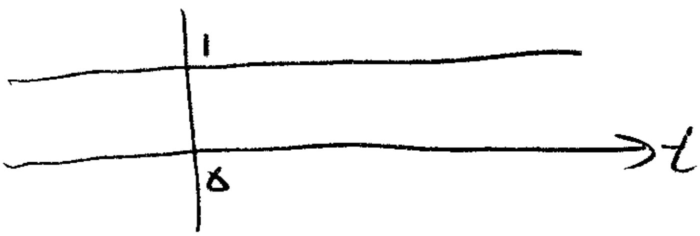
$$= \cos\left(2\pi(100)\frac{n}{100}\right)$$

$$= \cos\left(2\pi n\right) = 1 \text{ for all } n.$$



Notice $x[n]$ is exactly the same as if the orignal $x(t)$ had been the constant $\underline{1}$ for all time.

$x(t) = 1$



Thus, we can't tell $\cos(2\pi(100)t)$ from 1 if $T_s = \frac{1}{100}$.
(aliasing)

It turns out there is a limit on the range of frequencies that we can distinguish given $F_s = \frac{1}{T_s}$.

Shannon's Sample Theorem (Claude Shannon)

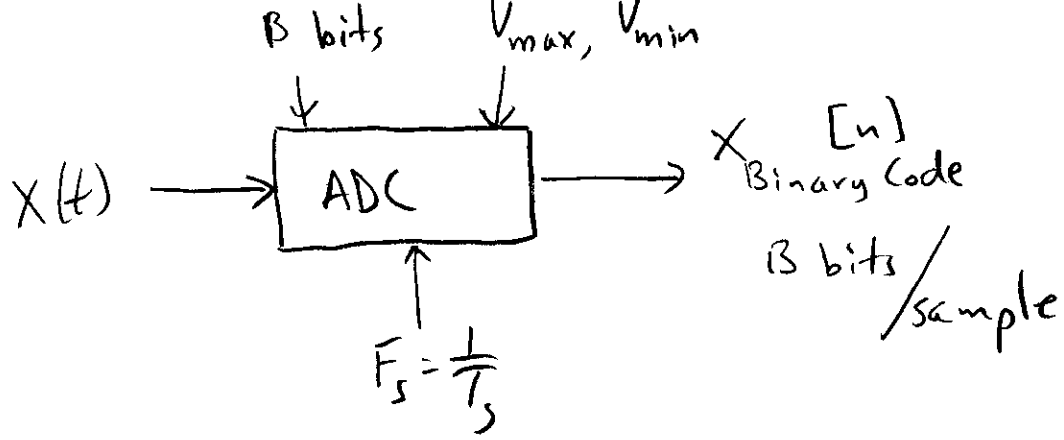
If $x(t)$ is bandlimited, such that the highest frequency in $x(t)$ is F_{max}, then $x(t)$ can be exactly reconstructed from its samples if

$$F_s = \frac{1}{T_s} > 2F_{max}$$

$2F_{max}$ = Nyquist Frequency

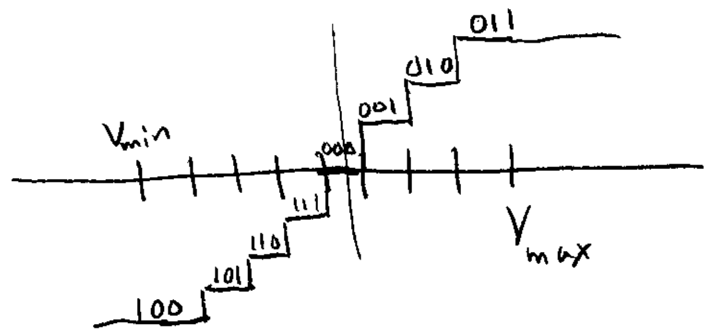
Ideal samplers (i.e., infinite precision samples) do not exist in real life.

However Analog to Digital Converters (ADCs) do exist. For the the number of bits/sample is finite.



$B=3$

$2^B = 8$ levels



In this case the output is a 2's complement signed binary integer.

This is not necessarily the same number as the value of $x(t)$, but a re-mapping onto integers.

ADC examples you may have seen.

PC sound chips (audio codec = coder/decoder)

2 channels → steve

F_s typically up to 44100 Hz, 48kHz, 96 kHz

B = 16, 20, 24 or more

Digital Audio Standard for CDs
16 bits, F_s = 44100 Hz

HDCD
24 bits, F_s = 44100 Hz

Some DVD audio (5.1 surround sound)
24 bits F_s = 48kHz or 96 kHz

24 → audiophile quality

Raspberry Pi - audiophile DAC Hats

iqaudio → Geekworm kit 24 bits } Texas Instruments DAC

inno-make → 32 bits

Hifiberry

DAC+ ADC pro
DAC2 pro
DAC2 HD
} 24 bits Burr Brown DAC

The DAC2 Hats can also use a DSP chip add-on board for real-time filtering