6.3000: Signal Processing

Sampling and Aliasing

September 19, 2023

Importance of Discrete Representations

Our goal is to develop **signal processing** tools to model interesting aspects of the world, to analyze the model, and to interpret the results.

$$\begin{tikzcd} \text{Model} \arrow[r, "\text{analyze}", "\text{(math, \textbf{computation})}"'] & \text{Result} \arrow[d, "\text{interpret results}"] \\ \text{World} \arrow[u, "\text{make model}"] & \text{New Understanding} \end{tikzcd}$$

The **increasing power** and **decreasing cost** of computation makes the use of computation increasingly attractive.

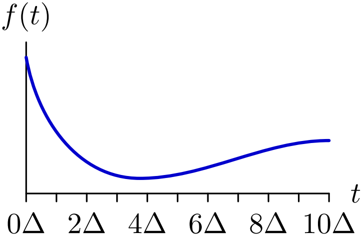
However, many important signals are naturally described with continuous functions, that must be **sampled** in order to be analyzed computationally.

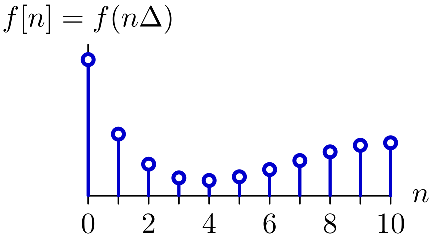
Today: understand relations between **continuous** and **sampled** signals.

Sampling

Sampling refers to the process by which a continuous-time signal $f(t)$ is converted to a discrete-time signal $f[n]$.

We use parentheses to denote functions of continuous domain (e.g., $f(t)$) and square brackets to denote functions of discrete domain (e.g., $f[n]$).





$\Delta =$ sampling interval

$f_s = \frac{1}{\Delta} =$ sampling frequency

How does sampling affect the information contained in a signal?

Effects of Sampling are Easily Heard

Sampling Music

$$f_s = \frac{1}{\Delta}$$

- $f_s = 44.1$ kHz
- $f_s = 22$ kHz
- $f_s = 11$ kHz
- $f_s = 5.5$ kHz
- $f_s = 2.8$ kHz

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Effects of Sampling are Easily Seen

Sampling Images

original: 2048×1536

Effects of Sampling are Easily Seen

Sampling Images

downsampled: 1024×768

Effects of Sampling are Easily Seen

Sampling Images

downsampled: 512×384

Effects of Sampling are Easily Seen

Sampling Images

downsampled: 256×192

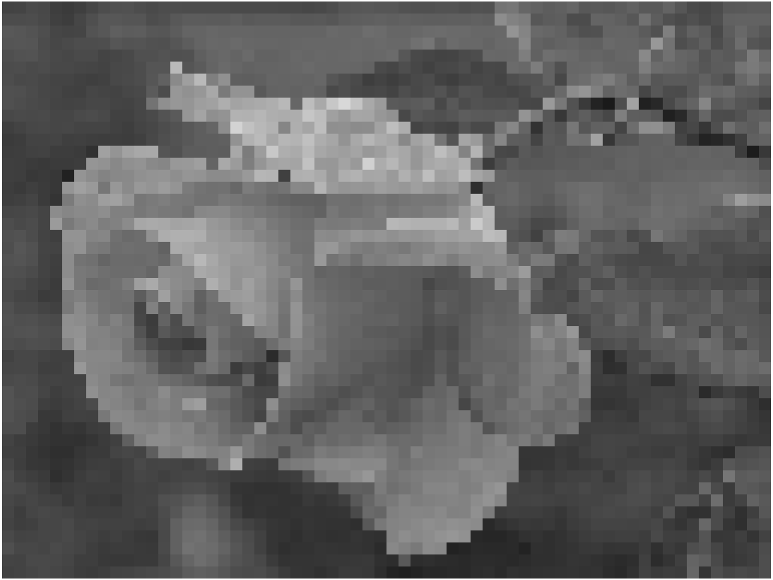
Effects of Sampling are Easily Seen

Sampling Images

downsampled: 128×96

Effects of Sampling are Easily Seen

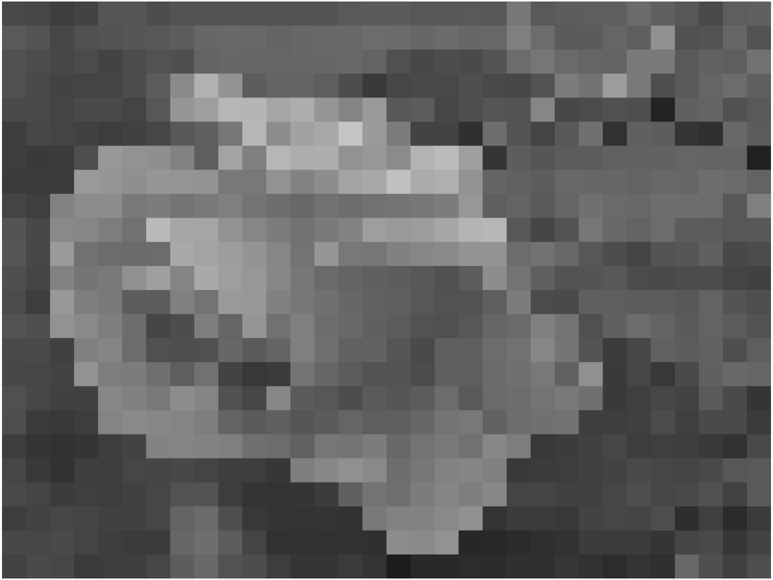
Sampling Images



downsampled: 64×48

Effects of Sampling are Easily Seen

Sampling Images



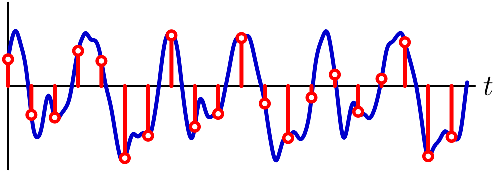
downsampled: 32×24

Characterizing Sampling

We would like to sample in a way that preserves **information**.

However, information is often **lost** in the sampling process.

Example: samples (red) provide no information about intervening values.

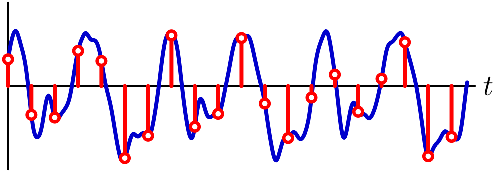


Characterizing Sampling

We would like to sample in a way that preserves **information**.

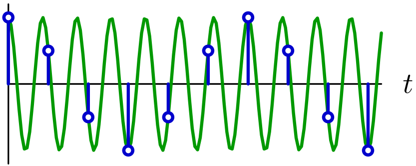
However, information is generally **lost** in the sampling process.

Example: samples (red) provide no information about intervening values.



Furthermore, information that is retained by sampling can be misleading.

Example: samples can suggest patterns not contained in the original.



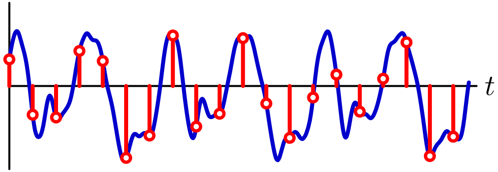
Samples (blue) of the original high-frequency signal (green)

Characterizing Sampling

We would like to sample in a way that preserves **information**.

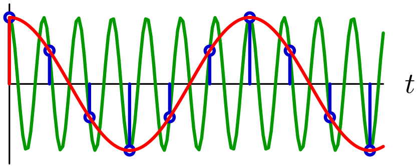
However, information is generally **lost** in the sampling process.

Example: samples (red) provide no information about intervening values.



Furthermore, information that is retained by sampling can be misleading.

Example: samples can suggest patterns not contained in the original.



Samples (blue) suggest an input that is much lower in frequency (red) than the original signal (green).

Characterizing Sampling

Our goal is to understand sampling so that we can mitigate its effects on the information contained in the signals we process.

Characterizing Sampling

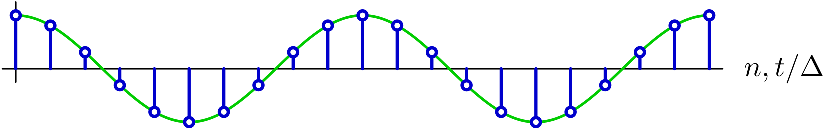
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.1$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.1\, n)$



Characterizing Sampling

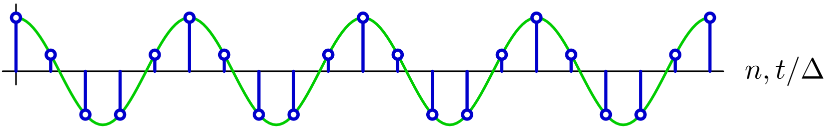
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.2$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.2\, n)$



Characterizing Sampling

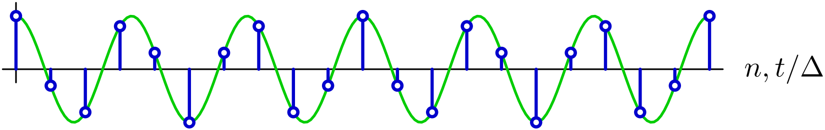
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.3$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.3\, n)$



Characterizing Sampling

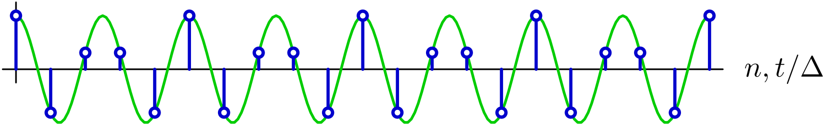
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.4$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.4\, n)$



Characterizing Sampling

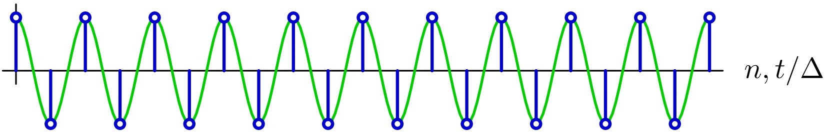
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.5$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.5\, n)$



Characterizing Sampling

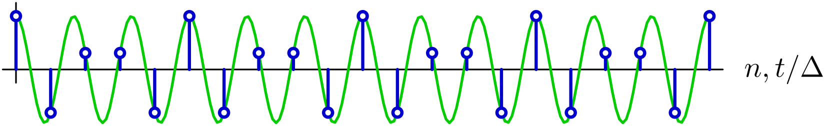
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.6$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.6\, n)$



Characterizing Sampling

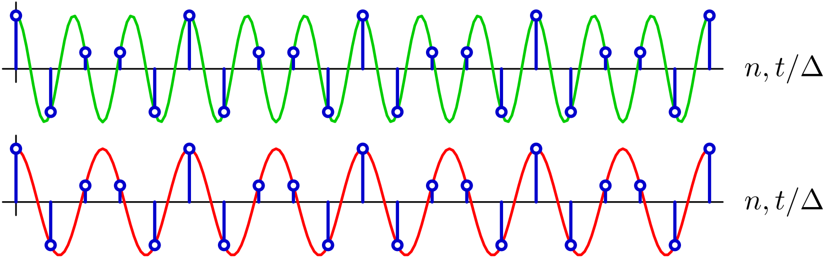
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.6$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.6\, n) = \cos(2\pi\, 0.4\, n)$



$$\begin{aligned}\cos(2\pi\, 0.6\, n) &= \cos(-2\pi\, 0.6\, n) = \cos(-2\pi\, 0.6\, n + 2\pi n) \\ &= \cos(2\pi(1-0.6)n) = \cos(2\pi\, 0.4\, n)\end{aligned}$$

Characterizing Sampling

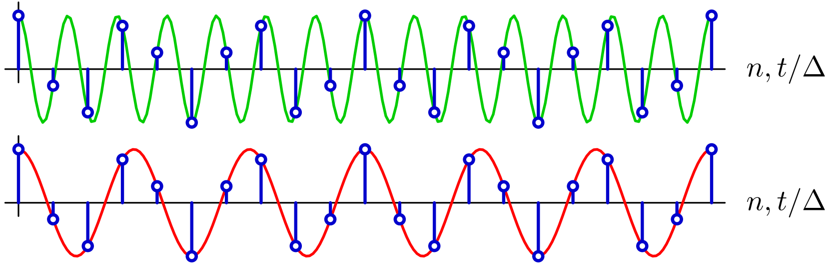
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.7$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.7\, n) = \cos(2\pi\, 0.3\, n)$



$$\begin{aligned}\cos(2\pi\, 0.7\, n) &= \cos(-2\pi\, 0.7\, n) = \cos(-2\pi\, 0.7\, n + 2\pi n)\\ &= \cos(2\pi(1-0.7)n) = \cos(2\pi\, 0.3\, n)\end{aligned}$$

Characterizing Sampling

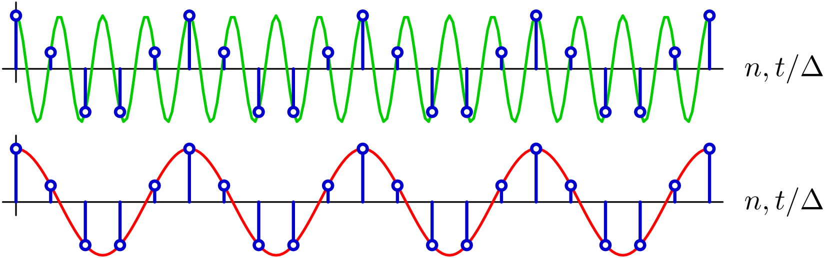
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.8$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 0.8\, n) = \cos(2\pi\, 0.2\, n)$



$$\begin{aligned} \cos(2\pi\, 0.8\, n) &= \cos(-2\pi\, 0.8\, n) = \cos(-2\pi\, 0.8\, n + 2\pi n) \\ &= \cos(2\pi(1-0.8)n) = \cos(2\pi\, 0.2\, n) \end{aligned}$$

Characterizing Sampling

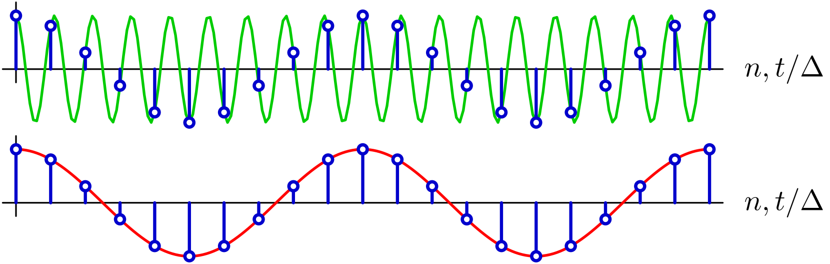
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

$\Delta = 0.9$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\,0.9\,n) = \cos(2\pi\,0.1\,n)$



$$\begin{aligned}\cos(2\pi\,0.9\,n) &= \cos(-2\pi\,0.9\,n) = \cos(-2\pi\,0.9\,n + 2\pi n)\\ &= \cos(2\pi(1-0.9)n) = \cos(2\pi\,0.1\,n)\end{aligned}$$

Characterizing Sampling

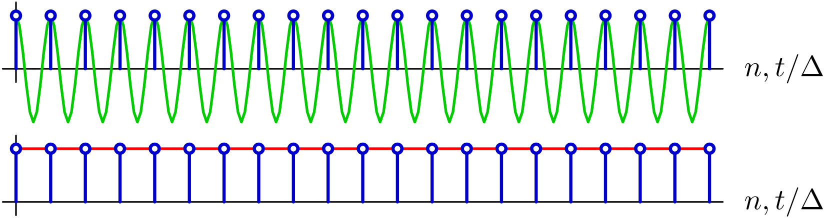
First, consider sampling a cosine function with fixed frequency $\omega = 2\pi$.

Sample $x(t) = \cos(2\pi t)$ every Δ seconds to obtain $x[n] = \cos(2\pi\Delta n)$.

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

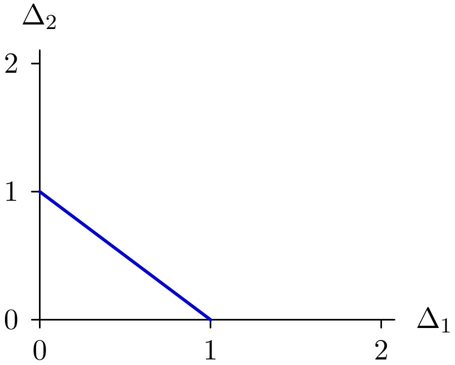
$\Delta = 1.0$

$x[n] = \cos(2\pi\Delta n) = \cos(2\pi\, 1.0\, n) = \cos(2\pi\, 0.0\, n)$



Characterizing Sampling

The same sequence of samples results when $x(t)=\cos(2\pi t)$ is sampled at intervals Δ_1 or Δ_2 if $\Delta_2=1-\Delta_1$.

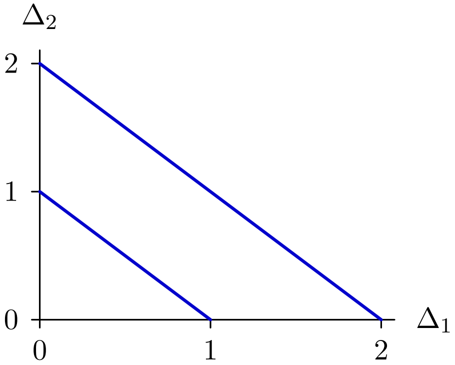


$$x[n] = \cos(2\pi\Delta_2 n) = \cos\Big(2\pi(1-\Delta_1)n\Big) = \cos(2\pi n - 2\pi\Delta_1 n) = \cos(2\pi\Delta_1 n)$$

Points on this line represent pairs of sampling intervals (Δ_1 and Δ_2) that generate the same sequence of samples.

Characterizing Sampling

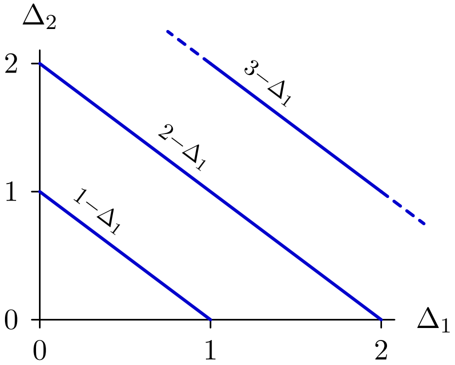
Similarly, the same sequence of samples results when $x(t)=\cos(2\pi t)$ is sampled at intervals Δ_1 or Δ_2 if $\Delta_2=2-\Delta_1$.



$$x[n] = \cos(2\pi\Delta_2 n) = \cos\Big(2\pi(2-\Delta_1)n\Big) = \cos(4\pi n - 2\pi\Delta_1 n) = \cos(2\pi\Delta_1 n)$$

Characterizing Sampling

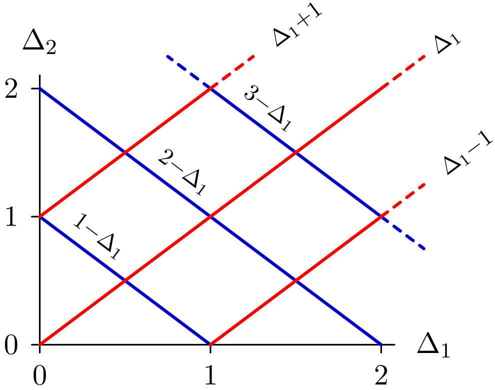
Any integer shift also works.



$$x[n] = \cos(2\pi\Delta_2 n) = \cos\Big(2\pi(N-\Delta_1)n\Big) = \cos(2N\pi n - 2\pi\Delta_1 n) = \cos(2\pi\Delta_1 n)$$

Characterizing Sampling

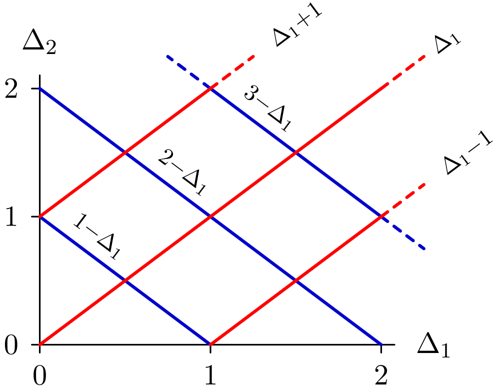
Sampling $x(t)=\cos(2\pi t)$ at $t=\Delta_1 n$ or $t = \Delta_2 n$ also generates the same sequence of samples when $\Delta_2 = N+\Delta_1$.



$$x[n]=\cos(2\pi\Delta_2 n)=\cos\Big(2\pi(N+\Delta_1)n\Big)=\cos(2N\pi n+2\pi\Delta_1 n)=\cos(2\pi\Delta_1 n)$$

Characterizing Sampling

Sampling $x(t)=\cos(2\pi t)$ at $t=\Delta_1 n$ or $t = \Delta_2 n$ also generates the same sequence of samples when $\Delta_2 = N+\Delta_1$.



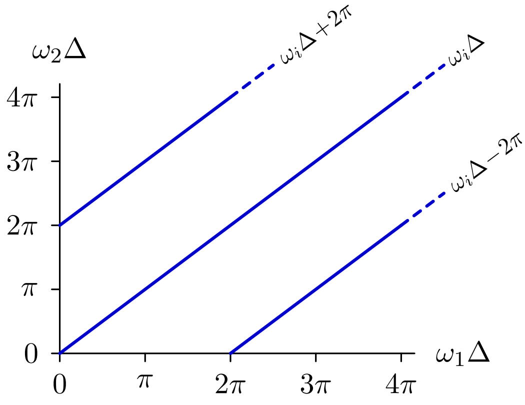
$$x[n]=\cos(2\pi\Delta_2 n)=\cos\Big(2\pi(N+\Delta_1)n\Big)=\cos(2N\pi n + 2\pi\Delta_1 n)=\cos(2\pi\Delta_1 n)$$

Many different sampling intervals result in the same sequence of samples.
$\rightarrow$ another special property of sinusoids

Characterizing Sampling

Sampling $\cos(\omega_1 t)$ and $\cos(\omega_2 t)$ with the same sampling interval Δ can also generate the same sequence of samples. For example, the same sequence of samples results if $\omega_2\Delta = \omega_1\Delta \pm 2\pi k$ for any integer value of k.

$$x[n] = \cos(\omega_2\Delta n) = \cos((\omega_1\Delta \pm 2\pi k)n) = \cos((\omega_1\Delta)n)$$

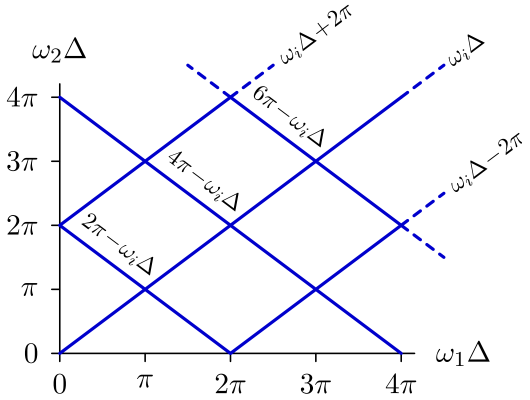


Each point on the lines above show a pair of frequencies (ω_1 and ω_2) that generate the same sequence of samples: $x[n] = \cos(\omega_1\Delta n) = \cos(\omega_2\Delta n)$.

Characterizing Sampling

Sampling $\cos(\omega_1 t)$ and $\cos(\omega_2 t)$ with the same sampling interval Δ can also generate the same sequence of samples. As a second example, the same sequence of samples results if $\omega_2\Delta = 2\pi k - \omega_1\Delta$ for any integer value of k.

$$x[n] = \cos(\omega_2\Delta n) = \cos((2\pi k - \omega_1\Delta)n) = \cos((-\omega_1\Delta)n) = \cos(\omega_1\Delta n)$$

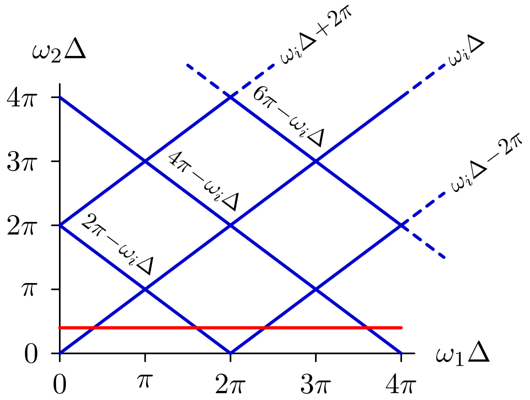


Each point on the lines above show a pair of frequencies (ω_1 and ω_2) that generate the same sequence of samples: $x[n] = \cos(\omega_1\Delta n) = \cos(\omega_2\Delta n)$.

Aliasing

Many input frequencies ω_1 generate the same output sequence of samples.

For example, the same samples would result if the input frequency ω_1 times Δ were 0.4π or 1.6π or 2.4π or ... Therefore, it's impossible to determine what frequency produced an output at frequency 0.4π.

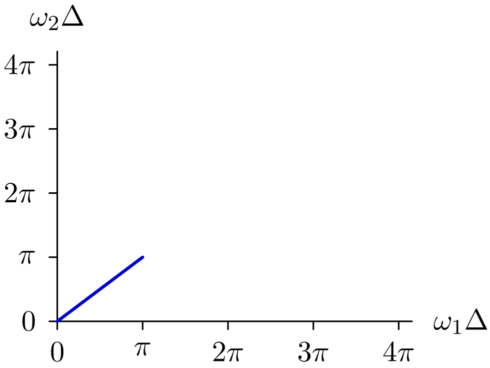


Since multiple frequencies ω_1 generate the same discrete samples, we say that these frequencies are **aliases** of each other.

Anti-Aliasing

We can prevent aliasing by removing **input** frequencies $\omega_1\Delta > \pi$ and disregarding **output** frequencies $\omega_2\Delta > \pi$.

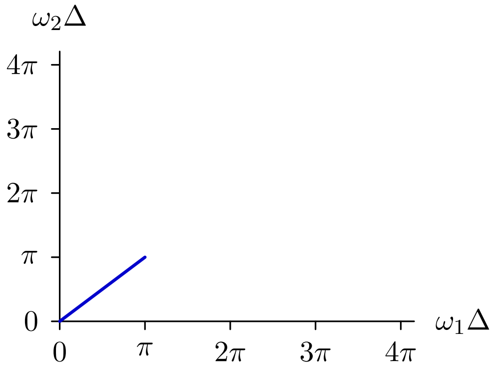
We call this low-frequency range of frequencies the **baseband**.



Anti-Aliasing

The maximum frequency that can be represented using this scheme is called the **Nyquist** frequency: $\omega_m = \pi/\Delta$, which equals half the sampling rate f_s.

$$f_m = \frac{\omega_m}{2\pi} = \frac{\pi/\Delta}{2\pi} = \frac{1}{2\Delta} = \frac{f_s}{2}$$



Check Yourself

Consider 3 CT signals:

$$f_1(t) = \cos(4000t) \quad ; \quad f_2(t) = \cos(5000t) \quad ; \quad f_3(t) = \cos(6000t)$$

Each of these is sampled so that

$$f_1[n] = f_1(n\Delta) \quad ; \quad f_2[n] = f_2(n\Delta) \quad ; \quad f_3[n] = f_3(n\Delta)$$

where $\Delta = 0.001$.

Which list goes from lowest to highest (baseband) frequency?

0. $f_1[n] \quad f_2[n] \quad f_3[n]$
1. $f_1[n] \quad f_3[n] \quad f_2[n]$
2. $f_2[n] \quad f_1[n] \quad f_3[n]$
3. $f_2[n] \quad f_3[n] \quad f_1[n]$
4. $f_3[n] \quad f_1[n] \quad f_2[n]$
5. $f_3[n] \quad f_2[n] \quad f_1[n]$

Check Yourself

The CT signals are simple sinusoids:

$$f_1(t) = \cos(4000t) \quad ; \quad f_2(t) = \cos(5000t) \quad ; \quad f_3(t) = \cos(6000t)$$

The DT signals are sampled versions ($\Delta = 0.001$):

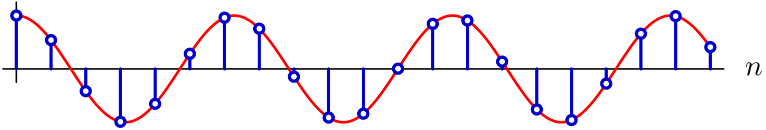
$$f_1[n] = \cos(4n) \quad ; \quad f_2[n] = \cos(5n) \quad ; \quad f_3[n] = \cos(6n)$$

How do these discrete-time functions differ?

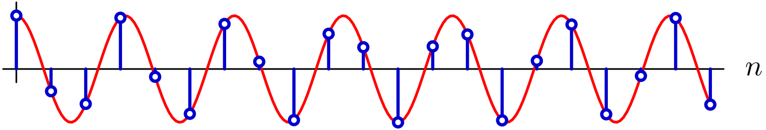
Check Yourself

As frequency increases, the shapes of the sampled signals deviate from those of the underlying CT signals.

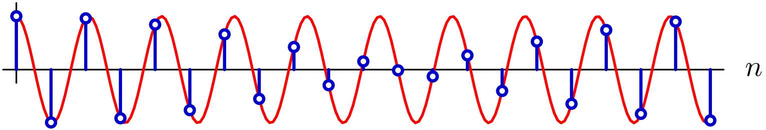
$\Omega = 1 : x[n] = \cos(n)$



$\Omega = 2 : x[n] = \cos(2n)$



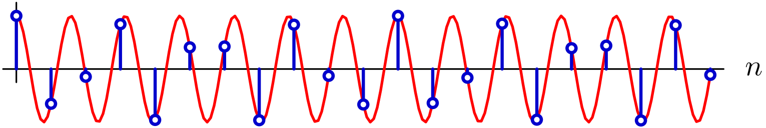
$\Omega = 3 : x[n] = \cos(3n)$



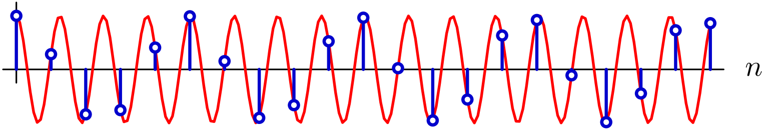
Check Yourself

Worse and worse representation.

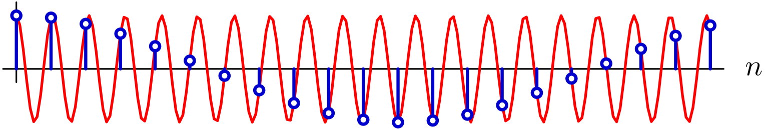
$$\Omega = 4 : x[n] = \cos(4n) = \cos\left((2\pi - 4)n\right) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos\left((2\pi - 5)n\right) \approx \cos(1.283n)$$



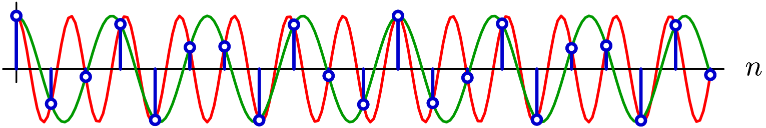
$$\Omega = 6 : x[n] = \cos(6n) = \cos\left((2\pi - 6)n\right) \approx \cos(0.283n)$$



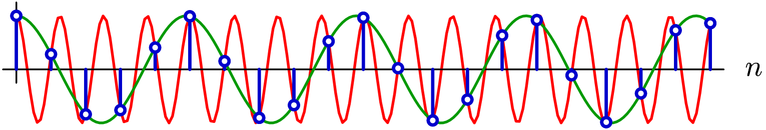
Check Yourself

For $\Omega > \pi$, a lower frequency Ω_L has the same sample values as Ω.

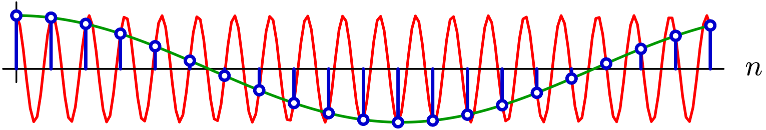
$$\Omega = 4 : x[n] = \cos(4n) = \cos\Big((2\pi - 4)n\Big) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos\Big((2\pi - 5)n\Big) \approx \cos(1.283n)$$



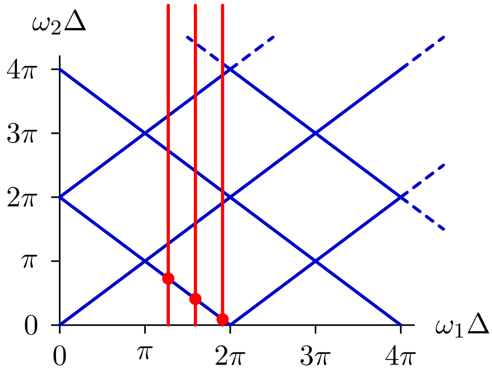
$$\Omega = 6 : x[n] = \cos(6n) = \cos\Big((2\pi - 6)n\Big) \approx \cos(0.283n)$$



The same DT sequence represents multiple different values of Ω.

Check Yourself

Graphically: As the input frequency $\omega_1\Delta$ goes from 4 to 5 to 6, the output baseband frequency decreases from approximately 2.3 to 1.3 to 0.3.



Check Yourself

Consider 3 CT signals:

$$f_1(t) = \cos(4000t) \quad ; \quad f_2(t) = \cos(5000t) \quad ; \quad f_3(t) = \cos(6000t)$$

Each of these is sampled so that

$$f_1[n] = f_1(n\Delta) \quad ; \quad f_2[n] = f_2(n\Delta) \quad ; \quad f_3[n] = f_3(n\Delta)$$

where $\Delta = 0.001$.

Which list goes from lowest to highest DT frequency? 5

0. $f_1[n] \quad f_2[n] \quad f_3[n]$
1. $f_1[n] \quad f_3[n] \quad f_2[n]$
2. $f_2[n] \quad f_1[n] \quad f_3[n]$
3. $f_2[n] \quad f_3[n] \quad f_1[n]$
4. $f_3[n] \quad f_1[n] \quad f_2[n]$
5. $f_3[n] \quad f_2[n] \quad f_1[n]$

Anti-Aliasing Demonstration

Sampling Music.

- $f_s = 11$ kHz without anti-aliasing
- $f_s = 11$ kHz with anti-aliasing
- $f_s = 5.5$ kHz without anti-aliasing
- $f_s = 5.5$ kHz with anti-aliasing
- $f_s = 2.8$ kHz without anti-aliasing
- $f_s = 2.8$ kHz with anti-aliasing

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto
Nathan Milstein, violin

Why does the aliased version (i.e., without anti-aliasing) sound so bad?
Why is the anti-aliased version so much better?

Importance of Discrete Representations

Our goal is to develop **signal processing** tools to model interesting aspects of the world, to analyze the model, and to interpret the results.

$$
\begin{tikzcd}
\text{Model} \arrow[r, "\text{analyze}", "\text{(math, \textbf{computation})}"'] & \text{Result} \arrow[d, "\text{interpret results}"] \\
\text{World} \arrow[u, "\text{make model}"] & \text{New Understanding}
\end{tikzcd}
$$

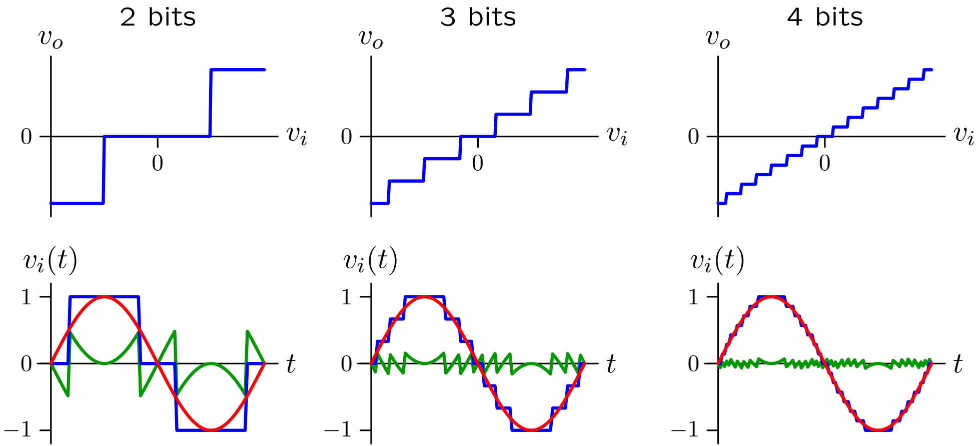
The **increasing power** and **decreasing cost** of computation makes the use of computation increasingly attractive.

However, many important signals are naturally described with continuous functions, that must be **sampled** in order to be analyzed computationally.

Today: understand relations between **continuous** and **sampled** signals.

Quantization

The information content of a signal depends not only with sample rate but also with the number of bits used to represent each sample.



Bit rate = (# bits/sample)×(# samples/sec)

Check Yourself

We hear sounds that range in amplitude from 1,000,000 to 1.

How many bits are needed to represent this range?

1. 5 bits
2. 10 bits
3. 20 bits
4. 30 bits
5. 40 bits

Check Yourself

How many bits are needed to represent 1,000,000:1?

bits	range
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	$1,024$
11	$2,048$
12	$4,096$
13	$8,192$
14	$16,384$
15	$32,768$
16	$65,536$
17	$131,072$
18	$262,144$
19	$524,288$
20	$1,048,576$

Check Yourself

We hear sounds that range in amplitude from 1,000,000 to 1.

How many bits are needed to represent this range? 3

1. 5 bits
2. 10 bits
3. 20 bits
4. 30 bits
5. 40 bits

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music

- 16 bits/sample
- 6 bits/sample
- 5 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantizing Images

Converting an image from a continuous representation to a discrete representation involves the same sort of issues.

This image has 280×280 pixels, with brightness quantized to 8 bits.

Quantizing Images

8 bit image

7 bit image

Quantizing Images

8 bit image

6 bit image

Quantizing Images

8 bit image

5 bit image

Quantizing Images

8 bit image

4 bit image

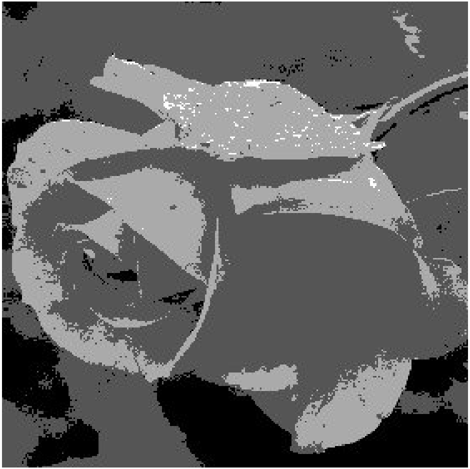
Quantizing Images

8 bit image

3 bit image

Quantizing Images

8 bit image



2 bit image

Quantizing Images

8 bit image

1 bit image

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto

Nathan Milstein, violin

Quantization Demonstration

Quantizing Music With and Without (Robert's) Dither

- 4 bits/sample
- 4 bits/sample with dither
- 3 bits/sample
- 3 bits/sample with dither
- 2 bits/sample
- 2 bit/sample with dither

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto
Nathan Milstein, violin

In what way is the dithered version better?

Summary

We are highly motivated to develop discrete representations of signals – especially when they represent signals that are naturally described with continuous functions.

Information is generally lost in such discretization processes.

Today we discussed two mechanisms that can alter the information contained in a signal: **aliasing** and **quantization**.

Next time, we will develop representations that are specialized for discrete-time signals.

Trig Table

```
sin(a+b) = sin(a) cos(b) + cos(a) sin(b)
sin(a-b) = sin(a) cos(b) - cos(a) sin(b)
cos(a+b) = cos(a) cos(b) - sin(a) sin(b)
cos(a-b) = cos(a) cos(b) + sin(a) sin(b)
tan(a+b) = (tan(a)+tan(b))/(1-tan(a) tan(b))
tan(a-b) = (tan(a)-tan(b))/(1+tan(a) tan(b))

sin(A) + sin(B) =  2 sin((A+B)/2) cos((A-B)/2)
sin(A) - sin(B) =  2 cos((A+B)/2) sin((A-B)/2)
cos(A) + cos(B) =  2 cos((A+B)/2) cos((A-B)/2)
cos(A) - cos(B) = -2 sin((A+B)/2) sin((A-B)/2)

sin(a+b) + sin(a-b) =  2 sin(a) cos(b)
sin(a+b) - sin(a-b) =  2 cos(a) sin(b)
cos(a+b) + cos(a-b) =  2 cos(a) cos(b)
cos(a+b) - cos(a-b) = -2 sin(a) sin(b)

2 cos(A) cos(B) = cos(A-B) + cos(A+B)
2 sin(A) sin(B) = cos(A-B) - cos(A+B)
2 sin(A) cos(B) = sin(A+B) + sin(A-B)
2 cos(A) sin(B) = sin(A+B) - sin(A-B)
```