I. Sampling of Analog Signals

I.1 Signals

Unidimensional and multidimensional signals

An **unidimensional** (1D) signal is a function of a single variable. Example: a voltage signal $v(t)$ only varies in time.

A **multidimensional** (2D, 3D $\ldots$ M-D) signal is a function of a multiple variables. Example: intensity of a grayscale image $I(x, y)$ across the surface of a photo.

In these lectures we consider only 1D signals. However, the techniques which you will learn for 1D signals can also be used for multidimensional signals (usually 2D signals, images).

Analog and discrete signals

Analog signals are functions of continuous variables, and there exists a signal value for any value of the variable within the defined range. **Discrete** signals are functions of discrete variables. These signals have values only at certain discrete values, typically indexed with integer numbers ($x[-1]$, $x[0]$, $x[1]\ldots$)

```
%matplotlib inline
import matplotlib.pyplot as plt, numpy as np, math
tanalog = np.arange(0,100)                    # this means 0,1,2
vanalog = np.sin(2*math.pi*0.03*tanalog)           # sin(2*pi*f
tdiscrete = np.arange(-2, 9)
vdiscrete = np.array([0, 1, 2, 2, 1, 0, -1, -2, -2, -1, 0])
plt.figure(figsize=(10, 3));
plt.subplot(121)
plt.plot(tanalog,vanalog); plt.title('Analog signal $a(t)$')
plt.subplot(122)
plt.stem(tdiscrete,vdiscrete); plt.title ('Discrete signal $b
```

```
<matplotlib.text.Text at 0xa47699ec>
```

Signals with continuous and discrete values

Besides the variable of the signal, the value of the signal can also be continuous or discrete.

A signal with continuous values can have any value in a certain defined range. For example, the voltage signal in one point in a usual electronic circuit can take any value between, for example, 0V and 5V.

A signal with discrete values can only have a value from a discrete set of possible values. For example, the number of bits received in a second over a binary communication channel can only be a natural number.

I.2 Discrete and analog frequency

Pulsation

For harmonic signals, the pulsation is defined as $\omega = 2\pi f$ for both analog and discrete signals.
We use ω when treating harmonic signals like *cos()* or *sin()*, e.g. $cos(\omega t) = cos(2\pi ft)$.

Domain of existence

For analog signals, the period can be as small as desired, $T \to 0$, and therefore the frequency can be very large, $F_{max} = \infty$.

For discrete signals, the smallest period is $N = 2$ (excluding N=1 which designates a constant signal). Therefore, the largest possible frequency is $f_{max} = \frac{1}{2}$.

Due to mathematical reasons (rememeber SCS class from last year), we will consider negative frequencies as well, mirroring the positive frequencies.

Therefore, the domain of existence for frequencies of analog signals ("analog frequencies") is:

$$F \in (-\infty, \infty),$$

whereas for frequencies of discrete signals ("discrete frequencies") it is:

$$f \in [-\frac{1}{2}, \frac{1}{2}].$$

I.3 Sampling of analog signals

Sampling equation

The process of sampling can be described by the following equation:

$$x[n] = x_a(n \cdot Ts).$$

The *n*-th value in the discrete signal x[n] is the value of the analog signal taken after *n* sampling periods, at time $t = nT_s$.

Sampling of harmonic signals

Consider a cosine signal $x_a(t) = cos(2\pi Ft)$, sampled with sampling frequency Fs. What is the resulting discrete signal $x[n]$?
Applying the sampling equation above, we obtain:

$$x[n] = x_a(n \cdot T_s) = x_a(n \cdot \frac{1}{F_s}) = cos(2\pi Fn\frac{1}{Fs})$$
$$= cos(2\pi\frac{F}{Fs}n) = cos(2\pi fn),$$

where

$$f = \frac{F}{F_s}.$$

Sampling an analog cosine/sine produces a discrete cosine/sine of similar form, but with discrete frequency $f = \frac{F}{F_s}$.

Example

Sampling an 100Hz analog cosine signal $x_a(t) = cos(2\pi 100t)$ with a sampling frequency $F_s = 300Hz$ produces a discrete signal $x[n] = cos(2\pi\frac{1}{3}t)$. The discrete frequency is $f = \frac{1}{3}$
Sampling the same signal with higher sampling frequency $F_s = 500Hz$

Sampling theorem

The Nyquist-Shannon sampling theorem:
If a signal that has maximum frequency F_{max} is sampled with a a sampling frequency $F_s \geq 2F_{max}$, then it can be perfectly reconstructed from its samples using the formula:

$$x_a(t) = \sum_{n=-\infty}^{+\infty} x[n] \cdot \frac{sin(\pi(F_s t - n))}{\pi(F_s t - n)}.$$

Comments * All the information in the original signal is contained in the samples, provided that they are sampled correctly * We can process discrete samples instead of the original analog signals * Sampling with $F_s \geq 2F_{max}$ ensures that the resulting discrete frequencies are smaller than $f = \frac{F}{F_s} \leq \frac{F_{max}}{F_s} \leq \frac{1}{2}$

Aliasing

What happens when the sampling frequency is not high enough? The resulting discrete frequency is higher than $\frac{1}{2}$ and becomes **identical** to a frequency smaller than $\frac{1}{2}$.
This phenomenon is known as **aliasing**.
Example. Signal $x_a(t) = cos(2\pi 10t)$ is sampled with $F_s = 15Hz$.
According to the sampling equation, the result is:

$$x[n] = cos(2\pi \frac{10}{15} n) = cos(2\pi \frac{2}{3} n).$$

But, the frequency $\frac{2}{3}$ is actually identical to $\frac{1}{3}$:

$$cos(2\pi \frac{2}{3} n) = cos(2\pi \frac{1}{3} n).$$

Proof: since cos() is periodical, we can subtract a multiple 2π:

$$\begin{aligned} cos(2\pi \frac{2}{3} n) &= cos(2\pi \frac{2}{3} n - 2\pi n) = cos(2\pi (\frac{2}{3} - 1) n) \\ &= cos(-2\pi \frac{1}{3} n) = cos(2\pi \frac{1}{3} n) \end{aligned}$$

Aliasing

http://www.dictionary.com/browse/alias:
"alias": *a false name used to conceal one's identity; an assumed name*
Rule: every discrete frequency that exceeds $f_{max} = \frac{1}{2}$ is equivalent (an **alias**) to a frequency that is lower than $f_{max} = \frac{1}{2}$:

$$cos(2\pi(\frac{1}{2} + \epsilon)n) = cos(2\pi(\frac{1}{2} - \epsilon)n)$$

$$sin(2\pi(\frac{1}{2} + \epsilon)n) = -sin(2\pi(\frac{1}{2} - \epsilon)n)$$

Every frequency $f \in \mathbb{R}$ is actually identical to a frequency in the interval $f \in [-\frac{1}{2}, \frac{1}{2}]$, up to a minus sign (for *sin()*). Therefore it makes no sense to consider frequencies outside of this interval.
Aliasing is only valid for discrete frequencies, not analog!

What's the problem with aliasing?

Problem: sampling different analog signals, with different frequencies, will lead to exactly the same samples. How can be know from what signal did the samples come from? It is impossible.

Note: Always, there is only a single analog frequency $F \in [-\frac{Fs}{2}, \frac{Fs}{2}]$ that corresponds to the samples. For every discrete frequencies $f \in [-\frac{1}{2}, \frac{1}{2}]$ there exists a single correponding analog frequencies $F \in [-\frac{Fs}{2}, \frac{Fs}{2}]$, so for these there is no confusion. Confusion can appear only for analog frequencies higher than $\frac{F_s}{2}$, which are not sampled according to the sampling theorem.

Example If sampling frequency is $F_s = 15Hz$, all the following signals $cos(2\pi 5t)$, $cos(2\pi 10t)$, $cos(2\pi 20t)$, $cos(2\pi 30t)$ will produce the same samples.

Exercise: which are the next signals in the above sequence, which produce the same samples?

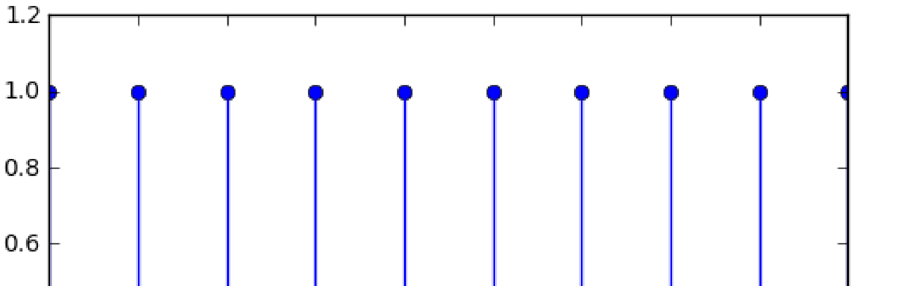
```
%matplotlib inline
import matplotlib.pyplot as plt, numpy as np, math
n = np.arange(0,10)
```

Exercise

What signals can produce the following samples:

```
%matplotlib inline
import matplotlib.pyplot as plt, numpy as np, math
n = np.arange(0,10)
plt.stem(n, np.ones(10));
plt.axis([0,9,-0.2,1.2])
```

```
[0, 9, -0.2, 1.2]
```



Ideal reconstruction of analog signals from samples

A discrete frequency $f \in [-\frac{1}{2}, \frac{1}{2}]$ will be reconstructed with the following equation:

$$x_r(t) = x[\frac{t}{T_s}] = x[t \cdot F_s].$$

For a discrete frequency outside this interval, the reconstruction uses the same equation, but the frequency f will be replaced with the equivalent frequency from the interval $[-\frac{1}{2}, \frac{1}{2}]$.
As a consequence, **reconstruction leads only to analog signals with frequencies** $F \in [-\frac{Fs}{2}, \frac{Fs}{2}]$.
Analog signals which were sampled according to the condition in the sampling theorem ($F_s \geq 2F$, $F \leq \frac{F_s}{2}$) will be reconstructed identically.
Analog signals which were not sampled accordingly will not be reconstructed identically. The reconstructed frequency will be the corresponding frequency from the base interval $[-\frac{Fs}{2}, \frac{Fs}{2}]$

Anti-alias filtering

If a signal has frequencies $F > \frac{F_s}{2}$, it is better to eliminate these frequencies from the signal before sampling. If they are left, the samples will just overlap with the samples of the corresponding frequency from the base interval $[-\frac{Fs}{2}, \frac{Fs}{2}]$ and will create confusion.
An **anti-alias filter** is an analog low-pass filter usually situated before a sampling circuit, designed to reject all frequencies $F > \frac{F_s}{2}$ from the signal before sampling.
They are typically integrated in hardware systems designed for audio processing or similar tasks.

I.4 Signal quantization and coding

I.5 A/D and D/A Conversion