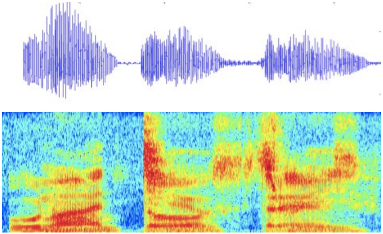
Estimation theory & parametric identification

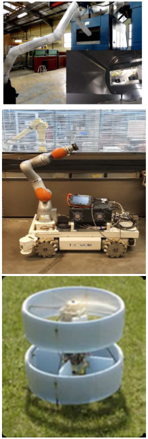


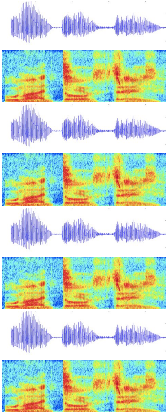
Time, frequency, and time-frequency representations of digital signals

N. Mechbal, **M. Rébillat**, M. Guskov *[PIMM, ENSAM]*
marc.rebillat@ensam.eu

Model estimation of robotic systems [1, 2, 3]

Robots ⇒ **Digital signals** ⇒ **Numerical models**





Transfer function

$$H(p) = \frac{Y(p)}{U(p)}$$

State-space model

$$\dot{X} = f(X, U)$$

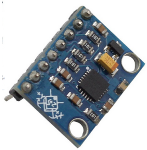
$$Y = g(X, U)$$

Objective: Estimate a reliable model and its parameters for a robotic system from measurements.

Model estimation of robotic systems [1, 2, 3]

Proprioceptive sensors present on robotic systems

- Accelerometers (m/s^2), angular displacements sensors (rad), gyroscopes (rad/s)
- Torque (Nm) and force sensors (N)
- Voltage (V) and current sensors (A)



Inertial Motion Unit (IMU): 3 accelerometers and 3 gyroscopes

Torque sensor

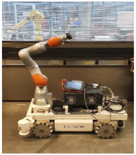
Current clamp sensor

A wide variety of signals coming from various sensors can be exploited on robotic systems.

Model estimation of robotic systems [1, 2, 3]

Noises present on robotic systems

- *Internal mechanical noises:* friction, self induced noise, ...
- *Internal electromagnetic (EM) noises:* diaphony, EM coupling, ...
- *External mechanical noises:* outdoor or industrial environments, ...
- *External EM noises:* electrical network, neighboring installations, ...



Terrestrial drone

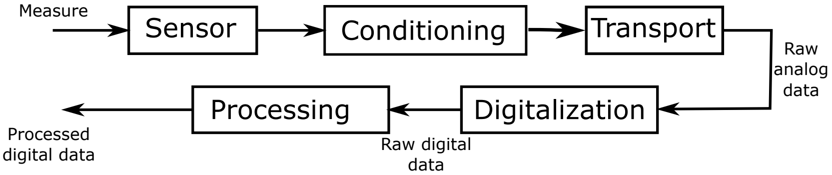
Aerial drone

Industrial robots performing arc soldering

Sensors placed on robotic systems are exposed to a wide variety of noises with various properties (large band or harmonic noises, noises correlated with actuation signals, ...).

Model estimation of robotic systems [1, 2, 3]

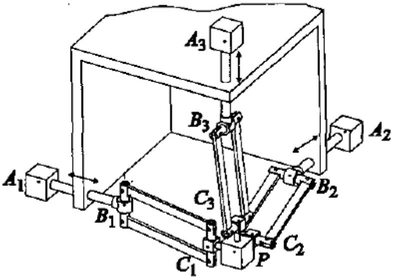
Data acquisition and processing chain



- How to correctly digitize the measurement signal?
- How to efficiently manage digital signals?
- How to process data in the digital domain?

Handling the data acquisition and processing chain is not straightforward.

Example #1: Identification of the dynamic parameters of the orthoglide [4]

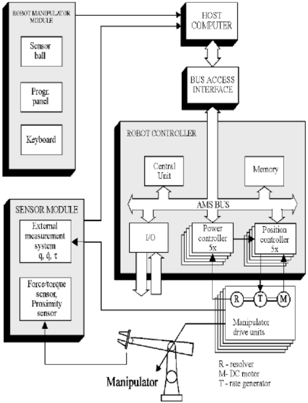


The orthoglide: a 3-DOF parallel robot

*"The **sampling period** is 2.5 ms. The joint positions are measured thanks to **digital encoders**. The joint positions have been filtered with a **4th-order low-pass Butterworth filter in both the forward and reverse directions** to avoid phase distortion. The corresponding **cut-off frequency is** 100 **Hz**."*

Some signal processing steps are indeed necessary for the identification procedure.

Example #2: The IRp-6 robot [3]

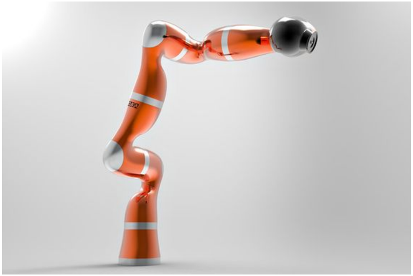


General architecture of the IRp-6 robot

"The strain gauge measurements are analog voltage signals which are very weak and have to be amplified by the amplifier and next converted by the ***analog-to-digital 12 bit converter****.* ***Filtering the measurement data via different filters chosen by the engineer is also possible****. The measurement system monitors* ***axis position, axis velocity, and torque of the driving motor for all degrees of freedom****. The joint torques are not measured directly; instead, the rotor currents of the DC motors are read instead."*

User knowledge regarding signal processing is necessary.

Example #3: FloBaRoID software package for the identification of robot dynamics parameters [5]



Kuka LWR 4+ arm

"The data comprises joint positions and torques as well as contact forces, if available. Numerical differentiation yields velocity and acceleration signals. ***Zero phase forward-backward digital filtering using a Butterworth filter*** *reduces signal and differentiation noise.* ***The filter cut-off frequencies may need to be tuned*** *to reduce noise further, as the optimal values depend on the individual robot mass, operating velocities and sensor properties."*

Signal processing parameters tuning needed for each case under study.

Course objectives

- **With M. Guskov**: Building models of flexible manipulator.
- **With N. Mechbal**: Model parameters estimation.

Within this part of the course

Understanding the signal processing steps of measurements coming from robotic sensors that are necessary to perform model parameters estimation:

1. How do I represent usefully a digital signal?
2. How am I sure to control the data acquisition chain?
3. How do I process reliably acquired signals?
4. Illustration in robotics applications.

Useful references *are made available on the last slide for more details [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18].*

Part #1: t, f and t/f representations of signals

How do I represent usefully a digital signal and how do I control the acquisition chain?

Three examples of signals:

Sinus

White noise

Voice

- Which **representations** are useful for these signals?
- What are **sampling** and **quantification** consequences?
- What are the **relevant** information contained in a signal?

Robotic context

Discrete time representation

Sampling & aliasing

Discrete frequency representation

Time/frequency representations

Quantification

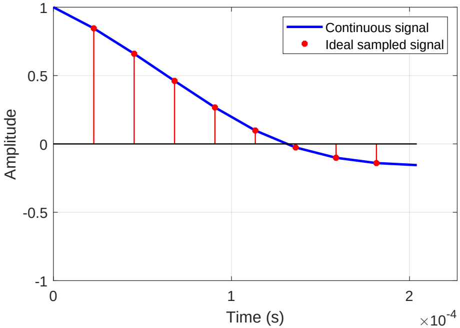
What is a signal in robotics? Tentative "definitions"....

Dynamical signals: We will consider here only **mono-dimensional** (1D) and **time-dependent** signals as they are representative of robotic applications.

Continuous time signal: The **analogic** description of a 1D-physical quantity evolving with time, denoted $e(t)$.

Sampled (or discrete-time) signal: The **digital** representation of a 1D-physical quantity evolving with time sampled uniformly over time. $e[n]$ denotes the n^{th} sample and samples can eventually be grouped in a vector.

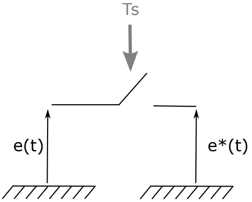
Sampled signals versus continuous signals



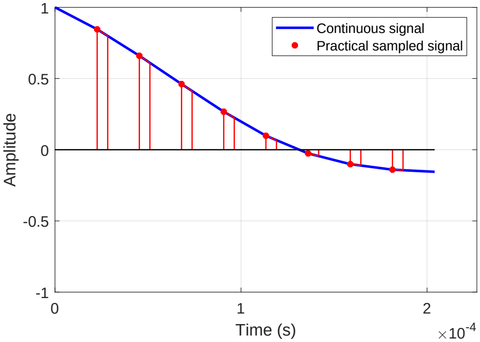
Analog signals are **continuous**, but digital signals are discrete-time and thus **sampled**:

- Digital signals are stored in the form of samples
- $F_s = 1/T_s$: Sampling frequency (ex: 44.1 kHz for audio)
- T_s: Sampling period

Sampling in practice



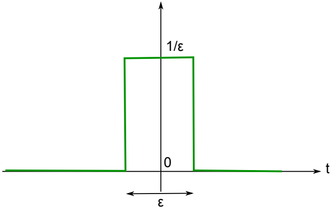
e(t): Continuous signal
e*(t): Sampled signal

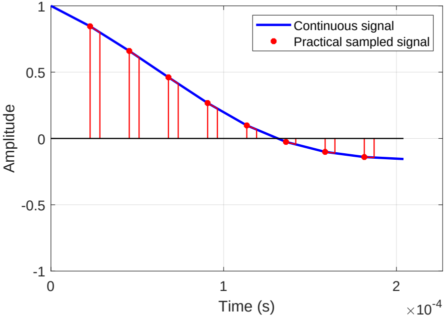


Electronic devices (***"analog to digital"*** **[ADC] converters**) convert the continuous signal $e(t)$ to a sampled signal $e^*(t)$:

- The continuous signal $e(t)$ is measured every T_s during ε.
- The resulting sampled signal is denoted $e^*(t)$.

Sampling in practice

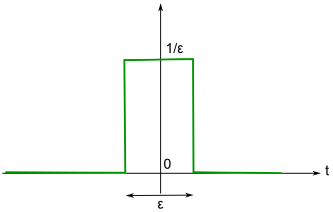


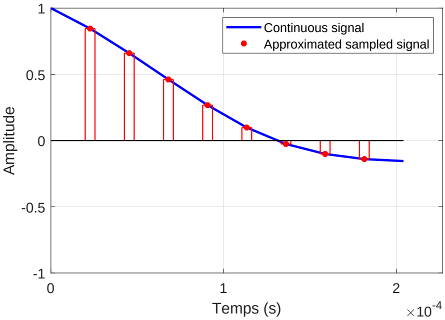


Mathematically, sampling is equivalent to a **multiplication by a *"door"* function** of unitary energy:

- **"Door" function**: $\pi_\varepsilon(t)$ of energy: $\displaystyle\int_{-\infty}^{+\infty} \pi_\varepsilon(t)dt = 1$
- **Sampled signal**: $\displaystyle e^*(t) = e(t) \sum_{n=-\infty}^{+\infty} \pi_\varepsilon(t - nT_e)$

Measurement value on a door



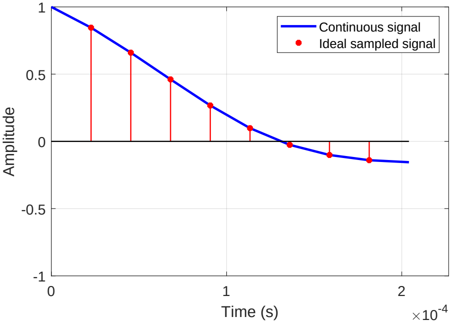


The **closing time ε of the door is extremely small**:

- **Assumption:** a constant value $e(nT_s)$ is maintained during acquisition (mean value over ε).
- **Sample value:**

$$e[n] = e^*(nT_s) = \int_{nT_s-\varepsilon/2}^{nT_s+\varepsilon/2} e(t)\frac{1}{\varepsilon}dt = e(nT_s)$$

Ideal sampling



- **The acquisition time ε tends toward 0**, while keeping constant energy (*i.e.* the same mean value, $\forall \varepsilon$).
- **Definition**: Dirac pulse
  - $\delta(t) = \lim_{\varepsilon \to 0} \pi_\varepsilon(t)$
  - $\int_{-\infty}^{+\infty} \delta(t) e(t) dt = e(0)$
  - $\int_{-\infty}^{+\infty} \exp(-j2\pi f t) df = \delta(t)$

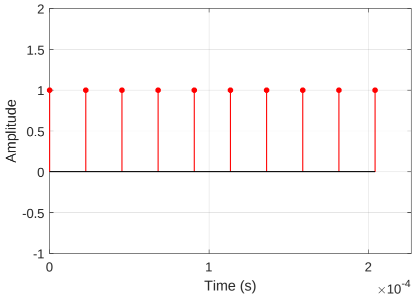
Sampled signal can be rewritten as:

$$e^*(t) = e(t) \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)$$

Dirac Comb

Definition:

$$\Pi(t) = \sum_{n\in\mathbb{N}} \delta(t - nT_s)$$



The Dirac Comb allows to directly link the continuous time signal $e(t)$ with the sampled signal $e^*(t)$:

$$\boxed{e^*(t) = e(t)\Pi(t)}$$

However, this unfortunately does not provide any insight with respect to the consequences of the sampling process...

Sampled signals within MATLAB/OCTAVE

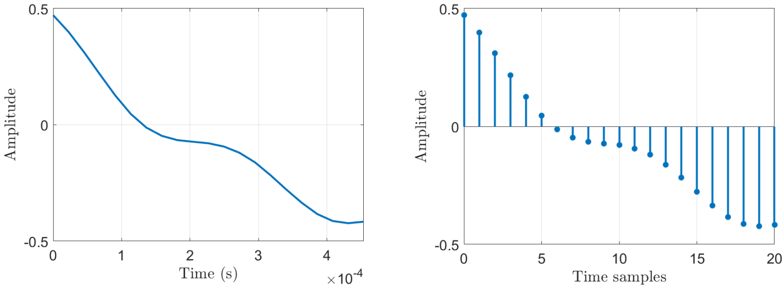
A sampled digital signal is in practice built up with:

- **Samples** contained within a real vector.
- **The sampling frequency** associated with samples.

To load and play an *audio* sampled signal within MATLAB/OCTAVE:

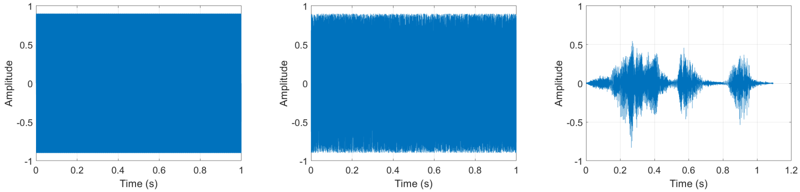
- **[Y,FS]=audioread(FILE)** returns the sample rate (FS) in Hertz and the number of bits per sample (NBITS) used to encode the data in the file.
- **soundsc(Y,FS)** sends the signal in vector Y (with sample frequency FS) out to the speaker on platforms that support sound.

Sampled signal within MATLAB/OCTAVE (example)



- The **time vector** associated with the sampled signal needs to be built.
- A time $T_s = 1/F_s$ elapses between two samples
- The time vector is given as **t = [0:length(x)-1]/Fs**

Conclusions regarding the temporal approach



(a) Sinus (b) Bruit blanc (c) Voix

Advantages and drawbacks of the temporal approach:

- **Intuitive** representation of the signals.
- **Difficult interpretation** of signals content.
- **Mathematical consequences** of sampling do not clearly appear.

[Homeworks #1] Practicing the temporal approach

1. Write a function that generates:
   1.1 A **sine wave** of duration T, amplitude A, and of frequency f_0 as well as the associated time vector.
   1.2 A **Gaussian white noise** of duration T with mean μ and standard deviation σ as well as the associated time vector.
   1.3 A **linear sweep** (*i.e.* a sine with a frequency evolving linearly with time) of duration T with amplitude A, start frequency f_0 and stop frequency f_1 as well as the associated time vector.
2. **Plot those signals** with respect to time and check for the sine wave that the frequency f_0 can be recovered graphically by measuring the period on the resulting figure.
3. **Generate** those signals with a sampling frequency $f_s = 44.1$ kHz and **play** them using the following sampling frequencies: $f_s = 44.1, 22.05, 11.025$ kHz. Conclude.

Robotic context

Discrete time representation

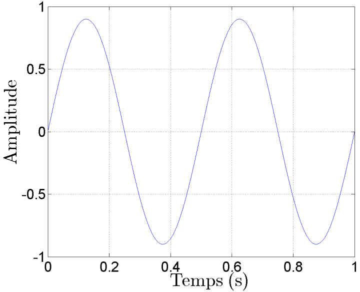
Sampling & aliasing

Discrete frequency representation

Time/frequency representations

Quantification

Toward frequency analysis...



$$x(t) = X_0 \sin(2\pi f_0 t)$$

Joseph FOURIER [1768-1830]

One of the most important questions raised and solved by the French mathematician Joseph FOURIER:

- What are the **sinusoids** contained **within a given signal**?
- What are their **amplitudes and frequencies**?

Fourier serie decomposition of a continuous periodic signal

A first tool to look at the frequency content of periodic continuous signals:

- For T-periodic signals only.
- $x(t) = \sum_{n \in \mathbb{N}} X_n e^{-j2\pi nt/T}$ with

  $X_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi nt/T} dt.$

Interpretation

- A T-periodic signal contains frequencies $\left\{\frac{n}{T}\right\}_{n \in \mathbb{N}}$.
- The amplitude with which the frequency $\frac{n}{T}$ is present is given by X_n.

But what about non-periodic signals?

Training: Fourier series of the Dirac comb

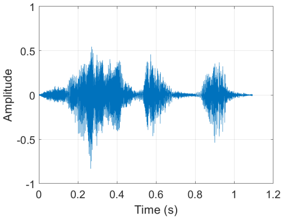
- Is the Dirac comb periodic? What is its period?
- What are the coefficients γ_n of its Fourier series decomposition?

$$\Pi(t) = \sum_{n \in \mathbb{N}} \delta(t - nT_s) = \sum_{n=-\infty}^{n=+\infty} \gamma_n exp(-j2\pi nt/T_s)$$

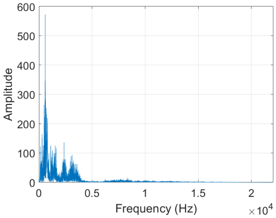
Continuous Fourier transform

A second tool to look at the frequency content of arbitrary continuous signals:

- Moving from the **temporal** domain to the **frequency** domain for arbitrary signals:
$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi f t}dt$$
- $x(t)$ and $X(f)$ are **equivalent** (i.e. same information inside).
- $X(f)$ is a **complex valued function** in general.



(a) $x(t)$



(b) $|X(f)|$

Training: Fourier transform of the Dirac comb

What is the Fourier transform of the Dirac Comb?

$$\epsilon(f) = \int_{-\infty}^{+\infty} \Pi(t) \exp(-j2\pi f t) dt$$

Training: Fourier transform of the sampled signal

Reminder: Convolution in the frequency domain

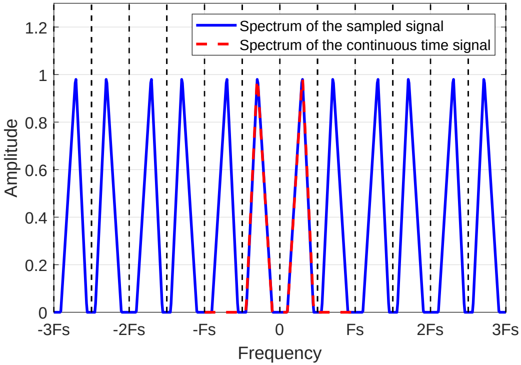
$$E^*(f) = (E * \epsilon)(f) = \int_{-\infty}^{+\infty} E(\nu)\epsilon(f-\nu)d\nu$$

What is the Fourier transform of the sampled signal?

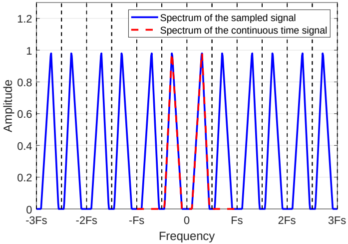
Fourier Transform of a sampled signal

$$\boxed{E^*(f) = \sum_{n=-\infty}^{n=+\infty} E(f - nF_s)}$$

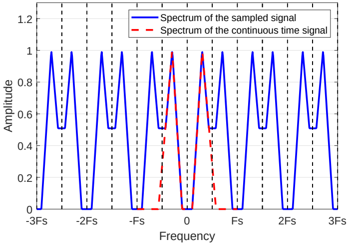
The spectrum of the sampled signal is a F_s**-periodic** repetition of the sprectrum of the continuous signal.



Spectral aliasing (a.k.a. Shannon's theorem)



The signal to be sampled does not contain energy above $F_s/2$.

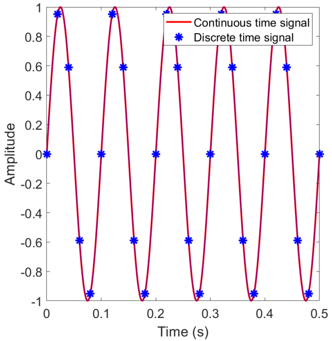


The signal to be sampled contains energy above $F_s/2$.

Spectral aliasing: If the signal to be sampled contains energy above $F_s/2$, then the spectra of the sampled and continuous time signals will be different in the range $[-F_s/2, F_s/2]$.

Spectral aliasing examples on sinusoids: $f_0 < F_s/2$

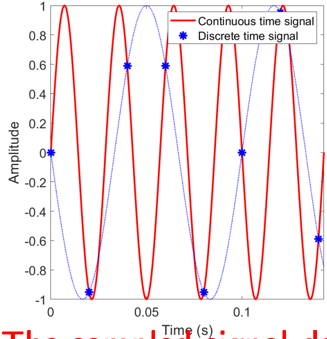
Sampling of a sinusoid with $F_s = 50$ Hz and $f_0 = 10$ Hz.

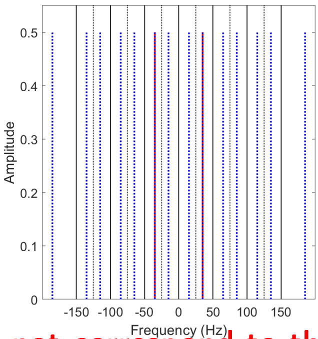


The sampled signal corresponds to the continuous time signal!

Spectral aliasing examples on sinusoids: $f_0 > F_s/2$

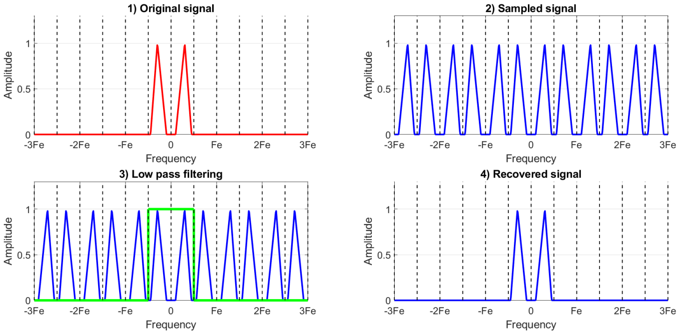
Sampling of a sinusoid with $F_s = 50$ Hz and $f_0 = 35$ Hz.





The sampled signal does not correspond to the continuous time signal but to a sinusoid at $\tilde{f}_0 = F_s - f_0 = 15$ Hz.

Reconstruction of a band-limited continuous time signal from the sampled one [frequency approach]



- If $f_{max} < F_s/2$, knowledge of the samples is enough to recover the whole continuous time signal.
- **Reconstruction is equivalent to band pass filtering.**

Training 1: Impulse response of the reconstruction filter

Reminder: Inverse Fourier transform

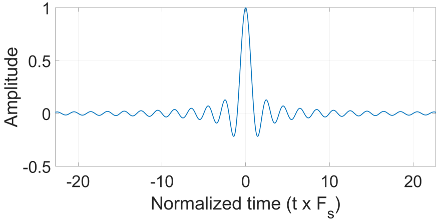
$$h(t) = \int_{-\infty}^{+\infty} H(f) e^{j2\pi f t} df$$

What is the impulse response $h(t)$ of the reconstruction filter $H(f)$ (i.e. its inverse Fourier transform)?

Training 2: What is the response of a filter to a Dirac?

Let's compute the impulse response of the filter $h(t)$ to one of the Diracs contained in the sampled signal.

Ideal reconstruction filter impulse response



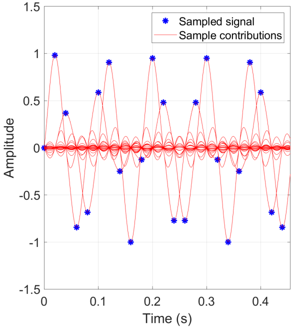
The ideal reconstruction filter is given by:

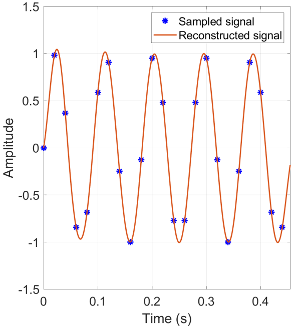
$$\boxed{h(t) = \frac{sin(\pi t F_s)}{\pi t}}$$

Usefull properties:

- $h(0) = 1$: It takes the sample value at the corresponding sample time.
- $\forall n \in \mathbb{N}^* \quad h(n) = 0$: It does not disturb at other sample times.
- It exactly interpolates between samples.

Reconstruction of a band-limited continuous time signal from the sampled one [temporal approach]





- **Sampled signal:**

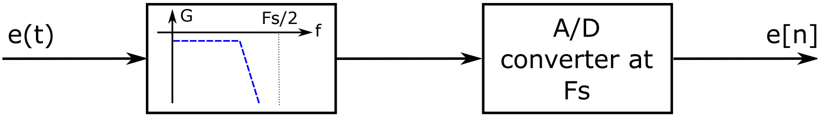
$$e^*(t) = e(t) \sum_{n=-\infty}^{+\infty} \delta(t - nT_e) = \sum_{n=-\infty}^{+\infty} e[n]\delta(t - nT_e)$$

- **Reconstructed signal:**

$$e^r(t) = (h * e^*)(t) = \sum_{n=-\infty}^{+\infty} e[n]h(t - nT_e)$$

Aliasing in practice: anti-aliasing filters

In order to avoid in practice spectral aliasing, it is mandatory to filter high frequencies (higher than $F_s/2$) of signals to be sampled. This is the role of the **anti aliasing filter**.

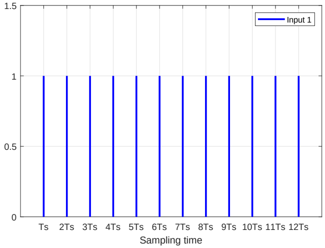


Note: All the digital to analog converters are not necessarily equipped with anti-aliasing filters. You have in practice to check this point yourself.

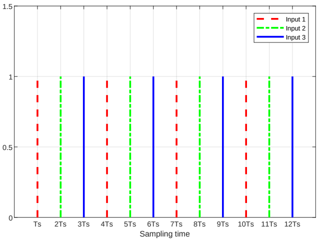
Temporal multiplexing due to shared A/D converters

Some acquisition boards possess several analog inputs with only one digital to analog converter working at the sampling frequency F_s.

- They provide the sampling frequency in **samples by second** (S/s).
- The more analog inputs are used, the lower the sampling frequency per input is.



Case with one input: The sampling frequency is F_s.



Case with three input: The sampling frequency is $F_s/3$.

Practical case study: Acquisition board NI USB 621x

- What is the sampling frequency of this converter?
- Does it possess anti aliasing filters?
- Are the various analog inputs temporally multiplexed?
- What is the sampling frequency when all input channels are used?

NI USB-621*x* Specifications

Analog Input

Number of channels
- USB-6210/6211/6212/ 6215/6216 .... 8 differential or 16 single ended
- USB-6218 .... 16 differential or 32 single ended

ADC resolution .... 16 bits

DNL .... No missing codes guaranteed

INL .... Refer to the *AI Absolute Accuracy Tables*

Sampling rate
- Maximum
  - USB-6210/6211/6215/6218 .... 250 kS/s single channel, 250 kS/s multichannel (aggregate)
  - USB-6212/6216 .... 400 kS/s single channel, 400 kS/s multichannel (aggregate)
- Minimum .... 0 S/s
- Timing accuracy .... 50 ppm of sample rate
- Timing resolution .... 50 ns

Input coupling .... DC

Input range .... ±10 V, ±5 V, ±1 V, ±0.2 V

Maximum working voltage for analog inputs (signal + common mode) .... ±10.4 V of AI GND

CMRR (DC to 60 Hz) .... 100 dB

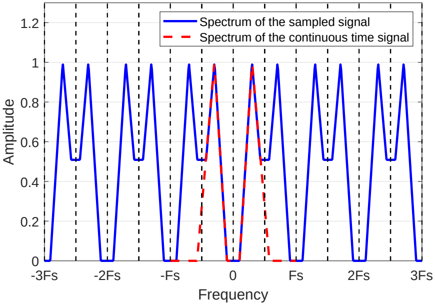
Input impedance
- Device on
  - AI+ to AI GND .... >10 GΩ in parallel with 100 pF
  - AI– to AI GND .... >10 GΩ in parallel with 100 pF
- Device off
  - AI+ to AI GND .... 1200 Ω
  - AI– to AI GND .... 1200 Ω

Input bias current .... ±100 pA

Crosstalk (at 100 kHz)
- Adjacent channels .... –75 dB
- Non-adjacent channels .... –90 dB

Small signal bandwidth (–3 dB)
- USB-6210/6211/6215/6218 .... 450 kHz
- USB-6212/6216 .... 1.5 MHz

Conclusions regarding sampling & aliasing



In order to correctly achieve sampling:

- **Ensure that $f_{max} < F_s/2$** (eventually using anti-aliasing filters).
- Keep in mind that in this case the **original signal can be exactly recovered**.
- Carefully **read the hardware documentation** to avoid troubles.

[Homeworks #2] Practicing Shannon (from [10])

1. Let us consider the continuous-time signal $x(t) = 2te^{-t}$, $t \in [0, 10]$. Show that sampling this signal results in spectrum aliasing if the sampling period is $T_s = 0.005$ s.
2. The aim of this exercise is to show that if the sampling frequency is not properly chosen, serious interpretation errors may occur while simulating or generating synthetic signals.
   - 2.1 Generate during 0.5 s a signal obtained by the sum of two sinusoids having the same amplitude (1 V), sampled at 256 Hz and whose frequencies are 100 Hz and 156 Hz respectively. Plot this signal and conclude about its shape.
   - 2.2 Generate and plot a sinusoid of 356 Hz sampled at 256 Hz. Compare this signal to another sinusoid of 100 Hz and sampled at the same frequency.
   - 2.3 Generate during 0.5 s a signal obtained by the sum of two sinusoids, etc...

Robotic context

Discrete time representation

Sampling & aliasing

Discrete frequency representation

Time/frequency representations

Quantification

Practical limits of the continuous Fourier transform

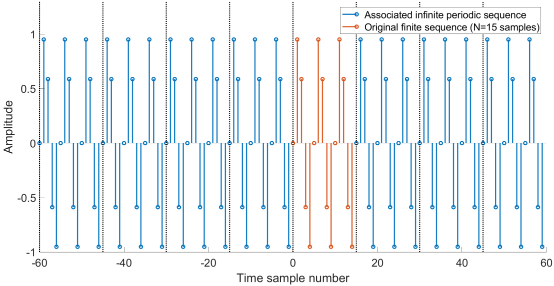
$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt$$

The continuous time Fourier transform cannot be applied in practice as:

- It requires a full knowledge of $x(t)$ for $\in [-\infty, +\infty]$.
- It is not based on a discrete-time signal.
- It is not itself discrete, and thus tractable for computers.

In practice, what we need to analyze in the frequency domain is a finite number of samples $\{e[n]\}_{n\in[1,N]}$.

Infinite periodic interpretation of a finite discrete sequence



Consider a periodic discrete sequence build up with a finite number of samples $\{e[n]\}_{n\in[1,N]}$ as $e_p[n] = \sum_{r\in\mathbb{N}} e[n-rN]$.

- Mathematically, it can be considered as a N-periodic infinite sequence $\{e_p[n]\}_{n\in\mathbb{N}}$.
- The following periodicity condition holds: $\forall r \in \mathbb{N} \quad e_p[n+rN] = e[n]$
- Then, **analysis tools for infinite periodic signals** can be used.

Fourier serie decomposition of a discrete signal

A useful tool to look at the frequency content of N-periodic discrete signals:

$$x[n] = \frac{1}{N} \sum_{k \in \mathbb{N}} X[k] e^{-j2\pi kn/N} \quad \text{with} \quad X[k] = \sum_{n=0}^{N-1} x[n] e^{j2\pi kn/N}$$

Difference with the continuous case:

$$\boxed{\forall r \in \mathbb{N} \quad e^{j2\pi(k+rN)n/N} = e^{j2\pi kn/N}}$$

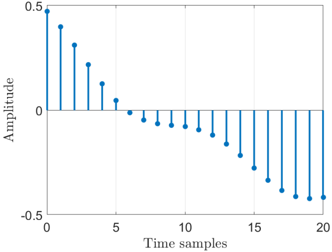
- A N-periodic signal contains **only N *"frequencies"*** $\left\{ \frac{n}{N} \right\}_{n \in [1,N]}$.
- **The sum is finite**: $x[n] = \sum_{k=0}^{N-1} X[k] e^{-j2\pi kn/N}$
- The **Fourier serie coefficients are N-periodic**: $X[k+N] = X[k]$.

Discrete Fourier Transform (DFT) over N points

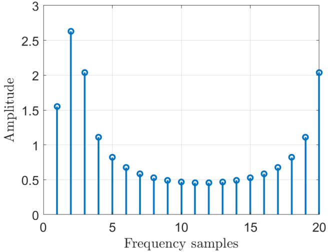
- From the **temporal** to the **frequency** domain for sampled signal:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/N}$$

- $x[n]$ and $X[k]$ represent the same sampled signal.
- $X[k]$ is complex in general.



(a) $x[n]$



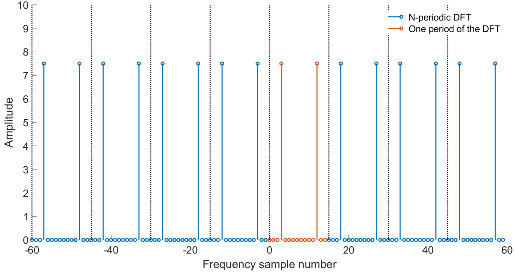
(b) $|X[k]|$

Periodicity of the DFT

The Discrete Fourier Transform is **N-periodic:**

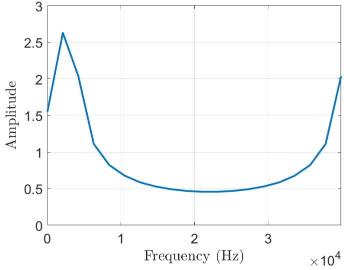
$$\forall r \in \mathbb{N}: \quad X[k+rN] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi n(k+rN)/N} = X[k]$$

As a consequence, **it is only necessary to represent the DFT over one period**.

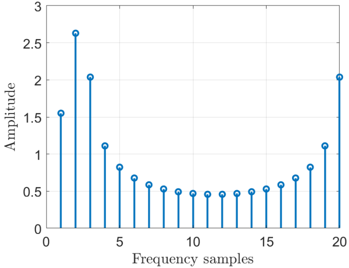


Note: Keep in mind that the original sequence is also N-periodic.

From DFT to Continuous Fourier Transform



(a) Continuous Fourier Transform



(b) Discrete Fourier Transform

The link between both representations is:

$$\frac{k}{N} = \frac{k}{NT_s} T_s = \frac{f_k}{F_s} \quad \text{with} \quad f_k = \frac{k}{NT_s}$$

As a consequence, **the k^{th} frequency sample is associated with the physical frequency $f_k = \frac{k}{N} \times F_s$.**

Increasing the frequency resolution [measurements]

The frequency resolution Δf, i.e. the physical frequency between two successive frequency samples is given by:

$$\Delta f = f_{k+1} - f_k = \frac{k+1}{NT_s} - \frac{k}{NT_s} = \frac{1}{NT_s}$$

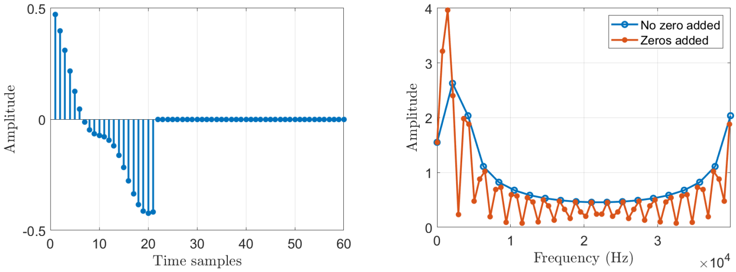
Trough measurements, it can be increased by:

- **Increasing the measurement time** NT_s for a given sampling frequency T_s. This requires to perform additional and longer measurements.
- **Increasing the sampling time** T_s for a given number of sampling points. This requires to perform additional and longer measurements. As T_s become larger F_s will become smaller and **spectral aliasing can occur**.

Increasing the frequency resolution [zero padding]

It can be assumed that the sequence for which the DFT must be processed is null after the measurements, leading artificially to more points in the sequence:

$$\tilde{x}[n] = \begin{cases} x[n], & \text{if } n \in [0, N-1] \\ 0, & \text{otherwise} \end{cases}$$



(a) Zero padded signal (b) Effect of zero padding

The appearance in the frequency domain resembles the true spectrum. However, **it does not increase the "true" frequency resolution.**

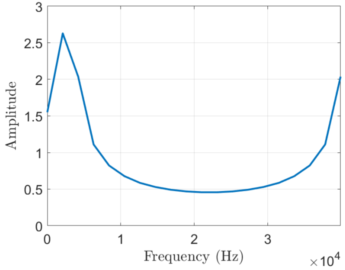
Efficiently computing the DFT: the FFT

DFT is the basis of many signal processing routines. How to compute it efficiently?

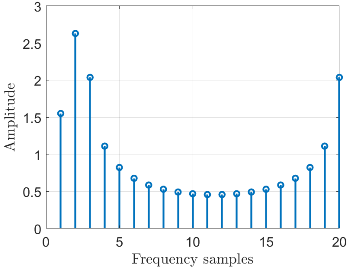
- For a discrete signal of length N, the computational cost of the the direct implementation of the DFT is in N^2.
- However the symmetry and periodicity properties of the exponential functions can be used to greatly reduce computational costs.
- This leads to the class of *Fast Fourier Transform* (FFT) algorithms.
- For FFT algorithms, if $N = 2^\alpha$, the associated computational cost is in $Nlog(N)$

Do not forget to select N as a power of 2 to speed up computations!

The DFT in practice with MATLAB/OCTAVE



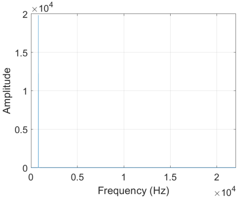
(a) Continuous Fourier Transform



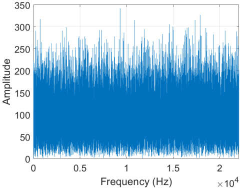
(b) Discrete Fourier Transform

- The ***fft* function** can be used to compute efficiently DFT.
- Zero padding can be achieved by using the second argument.
- For a DFT X, **the frequency vector can be built as [0:length(X)-1]/length(X) $\times F_s$**.
- The *fft function* returns DFT coefficients in the order $[-N/2, N/2]$ instead of $[0, N]$. Use the function *fftshift* if needed.

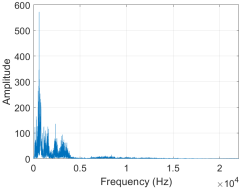
Conclusions regarding the frequency approach



(a) Sinus



(b) Bruit blanc



(c) Voix

Advantages and drawbacks:

- The DFT amplitude provides a **good overview of the spectral content** of the signal.
- **No temporal information** is available.
- Do not forget to **select N as a power of 2** to speed up computations if needed.

[Homeworks #3] Practicing the DFT (from [10])

1. A chirp pulse of width T can be expressed as: $x(t) = A\cos(\phi(t))$, where the instantaneous phase is given by: $\phi(t) = \Omega_0 t + \beta t^2$. A linear variation of the instantaneous frequency $f(t)$ during the time support T is then obtained according to: $\Omega(t) = 2\pi f(t) = d\phi(t)/dt$. Generate a chirp signal, whose instantaneous frequency varies between 2 and 5 kHz, during its time support $T = 10$ s. The sampling frequency is considered 20 kHz. Compute the DFT of this signal. Check that the start and stop frequencies corresponds to what has been chosen. Discuss the shape of its spectrum.
2. The goal of this exercise is to compare the spectral representation of a discrete-time signal to that obtained when it is periodized. Fisrt, generate a signal of duration $T = 10$ s, sampled at 10 kHz, who is equal to 1 during the first 2 seconds and equals to zero everywhere else. Compute and plot its DFT. Then, periodize this signal by concatenating 10 repetitions of it. Compute and plot the DFT of the periodized signal. Finally, explain the difference between both DFT.
3. Generate and plot the following signals: $x_1[n] = \begin{cases} 4, & \text{if } n \in [-3,3] \\ 0, & \text{otherwise} \end{cases}$ and $x_2[n] = \begin{cases} n, & \text{if } n \in [-3,3] \\ 0, & \text{otherwise} \end{cases}$. Check that the imaginary part of the DFT of signal $x_1[n]$ is zero and do the same for the real part of the DFT of signal $x_2[n]$. Explain why.

Why? Because of non-stationarity

Three example of signals:

Sinus [stationary]

White noise [stationary]

Voice [non-stationary]

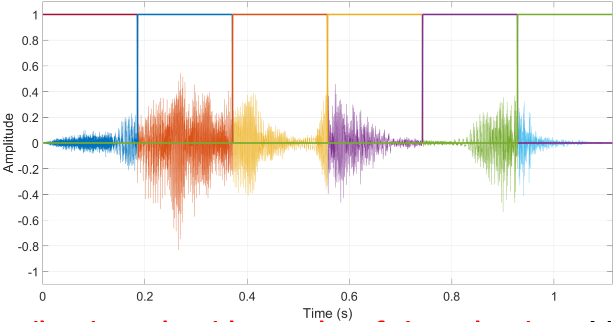
Signals of interest are not necessarily stationary:

- They can possess transient (starting of a tool, switch from a mode to another one, ...).
- Important information often appears through a simultaneous analysis of time and frequency properties of signals.
- Concentrating on transients is probably a strategy for selecting important information from recorded data (damage monitoring, impact detection, ...).

How do we proceed?

Main idea

- Temporal approaches are efficient to see **transients and non stationarities** in signals.
- Frequency approaches are efficient to look at the **frequency content** of signals.



Can we split a long signal in a series of short signals and look at their frequency contents evolution with time?

Short-Time Continuous Fourier Transform (STCFT)

Definition:

$$\tilde{X}(t,f) = \int_{-\infty}^{+\infty} x(\tau)w(\tau - t)e^{-j2\pi f\tau}d\tau$$

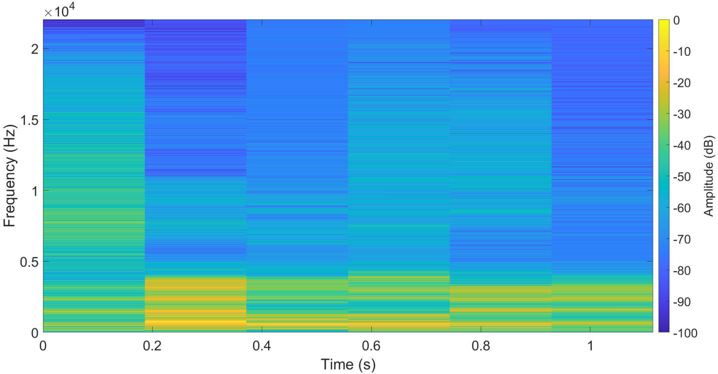
with:

- $w(\tau - t)$: the observation window centered on time t.
- $x(\tau)$: the continuous signal to analyze.

Comments:

- Also refered to as *"spectrogram"*.
- Many window type can be used (Rectangular, Hanning, etc...)
- Windows can overlap in the time domain in order to increase the temporal resolution.
- Like CFT, it cannot be directly used in practice ...

Understanding a spectrogram



- The x-axis represents the time (in s).
- The y-axis represents the frequency (in Hz).
- Colors are coding the DFT amplitude (in dB).

Can we be as precise as we want in both time and frequency? [1/2]

- **Round #1: The Dirac in time domain $\delta(t)$:**
  - Extremely well localized in the time domain (cannot be better)
  - In the frequency domain, not localized at all as:

$$\int_{-\infty}^{+\infty} \delta(t) e^{j2\pi f t} dt = 1$$

- **Round #2: The Dirac in frequency domain $\delta(f)$**
  - Extremely well localized in the frequency domain (cannot be better)
  - In the time domain, not localized at all as:

$$\int_{-\infty}^{+\infty} \delta(f) e^{-j2\pi f t} df = 1$$

Can we be as precise as we want in both time and frequency? [2/2]

- **Round #3: Scaling a temporal window from $g_1(t) = w(t)$ to $g_2(t) = \sqrt{s}w(st)$ with $s > 1$**
  - As $s > 1$, the energy will be better localized in time with $g_2(t)$ than with $g_1(t)$.
  - In the frequency domain:

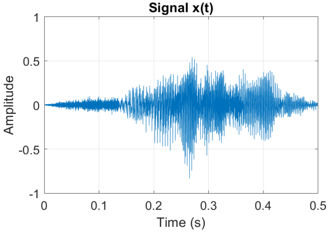
$$G_2(f) = \int_{-\infty}^{+\infty} \sqrt{s}w(st)e^{j2\pi ft}dt = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} w(u)e^{j2\pi \frac{f}{s}t}du$$

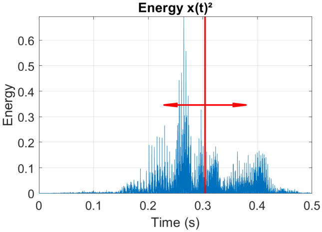
$$G_2(f) = \frac{1}{\sqrt{s}} G_1(\frac{f}{s})$$

the energy will be less localized in frequency with $g_2(t)$ than with $g_1(t)$

We cannot be as precise as we want in both domains. There is a limit.

Time uncertainty



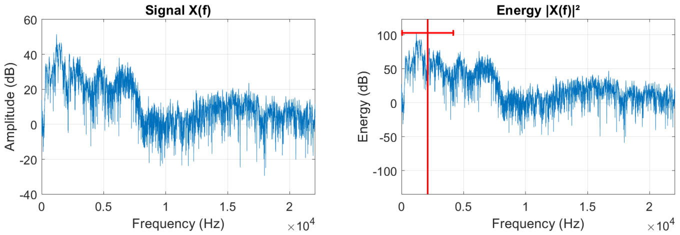


If we consider the signal $x(t)$ with energy $|x(t)|^2$, we can define:

- The mean time m_t value along which the energy of $x(t)$ is concentrated : $m_t = \int_{-\infty}^{+\infty} t|x(t)|^2 dt$
- The standard deviation σ_t in time along which the energy is spread: $\sigma_t = \int_{-\infty}^{+\infty} (t - m_t)|x(t)|^2 dt$

σ_t represents the **uncertainty with which the energy of $x(t)$ is localized in time**.

Frequency uncertainty



If we consider $X(f)$ the CFT of the signal $x(t)$, with energy $|X(f)|^2$, we can define:

- The mean frequency m_f value along which the energy of $X(f)$ is concentrated : $m_f = \int_{-\infty}^{+\infty} f|X(f)|^2 df$
- The standard deviation σ_f in frequency along which the energy is spread: $\sigma_f = \int_{-\infty}^{+\infty} (f - m_f)|X(f)|^2 df$

σ_f represents the **uncertainty with which the energy of $X(f)$ (or equivalently $x(t)$) is localized in frequency**.

Heisenberg uncertainty principle

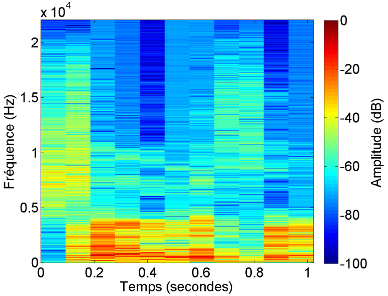
We cannot be as precise as we want in both the frequency and temporal domains: $\boxed{\sigma_t \times \sigma_f \geq \frac{1}{4}}$

- If a high time precision is required, large uncertainties in the frequency domain are expected.
- If a high frequency precision is required, large uncertainties in the time domain are expected.
- If a Gaussian window is chosen, then: $\sigma_t \times \sigma_f = \frac{1}{4}$.

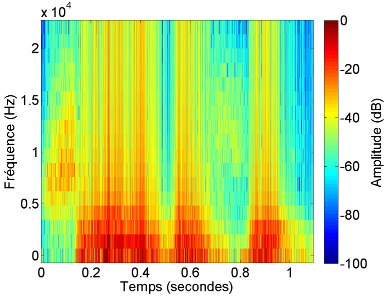
Illustration of Heisenberg principle in signal processing

Voice [non-stationary]

One signal, two windows of different widths:



(a) $T = 93$ ms



(b) $T = 0.7$ ms

- **Short** windows: large **temporal** precision.
- **Long** windows: large **frequency** precision.

Short-Time Discrete Fourier Transform (STDFT)

Definition:

$$\tilde{X}[m,k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]w[n-m]e^{-j2\pi nk/N}$$

with:

- $w[n-m]$: the observation window centered on sample m.
- $x[n]$: the discrete signal to analyze.

Comments:

- Many window type can be used (Rectangular, Hanning, etc...)
- Windows can overlap in the time domain in order to increase the temporal resolution.
- Like DFT, it can be directly used in practice!

Spectrogram in practice in MATLAB/OCTAVE

[S,F,T] = spectrogram($x[n]$,$w[n]$,$N_{overlap}$,N_{FFT},F_s)

With inputs:

- The signal to analyze $x[n]$ using the observation window $w[n]$, both sampled at F_s.
- $N_{overlap}$ which specifies the overlap (in samples) between two observation windows.
- N_{FFT} which specifies the number of points to compute the FFT.

And outputs:

- F: Vector of frequencies
- T: Vector of times
- S: Spectrogram matrix with rows corresponding to frequencies and columns to times.

To plot the spectrogram, use the function *"imagesc"* to plot the image and then the option *"axis('xy')"* to put the axis in the correct order.

Toward Wavelets...

In STFT, the observation window $g_t(\tau)$ selects a time range around t of the signal.

$$\tilde{X}(t,f) = \int_{-\infty}^{+\infty} x(\tau) g_t(\tau) e^{-j2\pi f\tau} d\tau \quad \text{with} \quad g_t(\tau) = w(t-\tau)$$

This can be rewritten as:

$$\tilde{X}(t,f) = \int_{-\infty}^{+\infty} x(\tau) g_{t,f}(\tau) d\tau \quad \text{with} \quad g_{t,f}(\tau) = w(t-\tau) e^{j2\pi f\tau}$$

Can this *"observation window"* $g_{t,f}(\tau)$ also selects a frequency range of the signal?

Continuous wavelets

A continuous wavelet ψ is defined as:

- A function of zero average $\int_{-\infty}^{+\infty} \psi(t)dt = 0$
- Which can be dilated (in frequency domain) with a scale parameter s and translated (in the temporal domain) by u as follows:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)$$

The corresponding wavelet transform is then:

$$\tilde{X}(u,s) = \int_{-\infty}^{+\infty} x(\tau)\frac{1}{\sqrt{s}}\psi\left(\frac{\tau-u}{s}\right)d\tau$$

Continuous wavelets are a "generalization" of the STCFT with dedicated "observation windows".

Discrete wavelets

A discrete wavelet ψ is defined as:

- A function of zero average $\int_{-\infty}^{+\infty} \psi(t)dt = 0$
- Which can be dilated (in discrete frequency domain) with a scale parameter $s = a^j$ and translated (in the sampled temporal domain) by m as follows:

$$\psi_{j,m}[n] = \frac{1}{\sqrt{a^j}} \psi \left(\frac{n-m}{a^j} \right)$$

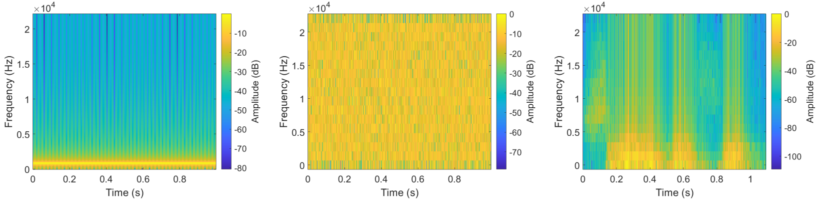
- a is defined as $a = 2^{1/\nu}$ in order to provide ν intermediate scales in each octave $[2^j, 2^{j+1}]$.

The corresponding discrete wavelet transform is then:

$$\tilde{X}[n,j] = \sum_{n=0}^{N-1} x[n] \psi_{j,m}[n]$$

Discrete wavelets can be seen as a "generalization" of the STDFT.

Conclusions regarding the time/frequency approach



(a) Sinus ($T = 92$ ms) (b) Noise ($T = 23$ ms) (c) Voice ($T = 1.5$ ms)

- **Good idea regarding** spectral content of the signal.
- **Temporal information** also available.
- But **several parameters to choose**: length T and shape of the observation windows, wavelet family, ...
- One **fundamental limitation**: the Heisenberg principle.

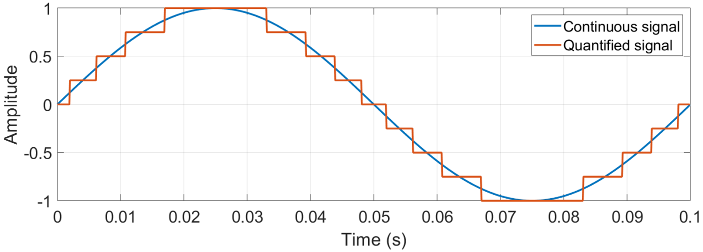
[Homework #4] Practicing Time/Frequency approaches (taken from [10]).

- A chirp signal, whose instantaneous frequency linearly sweeps the band between f_1 and f_2, is expressed as $s(t) = sin[(2\pi f_1 + 2\pi\beta t)t]$ with $\beta = (f_2 - f_1)/(2T)$, T being the signal duration. Generate this signal using a MATLAB code. Plot the generated signal and its CFT and comment. Illustrate the time frequency evolution of the chirp using the spectrogram and conclude.
- The spectrogram is the most traditional time-frequency analysis tool. However, it requires a trade-off between the time and frequency resolutions. Illustrate this spectrogram drawback as follows: generate a mixture of a Dirac pulse, which occurs at 26 ms, and a sinusoid with the frequency of 1 kHz, which occurs between 5 and 16 ms. Plot the spectrogram of this signal for different lengths of the analysis window. Comment on the results obtained.
- The scalogram is obtained by taking the square of the wavelet transform. Similarly to the spectrogram, it is also submitted to the Heisenberg-Gabor inequality involving the time and frequency resolutions. Write a MATLAB function to generate a Morlet wavelet, having as input parameters its length, the central frequency and the sampling frequency. The Morlet wavelet is defined by the following modulated Gaussian function:

$$h(t) = \left(\pi t_0^2\right)^{-1/4} \exp\left[-\frac{1}{2}\left(\frac{t}{t_0}\right)^2\right] \exp(-j2\pi\nu_0 t)$$

  Although this wavelet does not meet the zero-mean admissibility constraint, a good approximation is however obtained for $2\pi\nu_0 t_0 = 5.4285$. Plot the Morlet wavelet for different parameters to check that assumption. Write a MATLAB function to calculate the wavelet transform, having the following input parameters: sampling frequency, number of calculated points, maximum frequency, number of octaves and number of voices per octave. Generate the sum of a Dirac pulse and two truncated sinusoids and plot its scalogram. Discuss the variation of the time and frequency resolutions over the time-frequency plane.

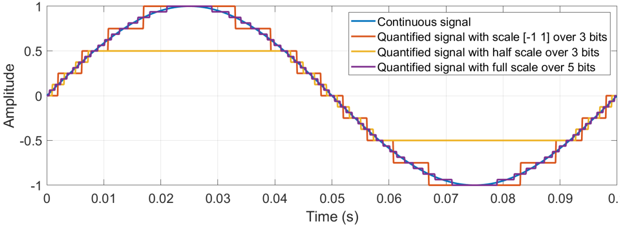
Analogic versus digital signals



Analogic signals: they vary continuously and this variation can be as small as desired.

Digital signals: the variation corresponds to a step with respect to the scale with which the signal is defined.

What is the smallest quantifiable value?



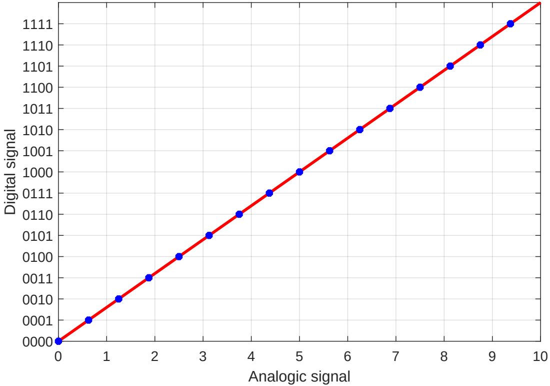
Scale : Values between which quantification is achieved. Defined as V_p.

- **Positive scale**: scale with only positive values and defined as $[0, V_p]$.
- **Symetric scale**: scale with both positive and negative values and defined as $[-V_p/2, V_p/2]$.

Number of bit: Number of binary coded values used to digitally store the quantified values. With n bits, 2^n binary coded values (numbered from 0 to $2^n - 1$) can be defined over a scale V_p.

Quantum: Smallest variation Q quantifiable by a given analog/digital converter and given by $Q = \dfrac{V_p}{2^n}$.

Discretizing the tension scale



Quantification: Discretisation using a digital number N of the tension v within the scale V_p.

$$\frac{v}{Q} = N + R \quad \text{with} \quad 0 \leq R < 1$$

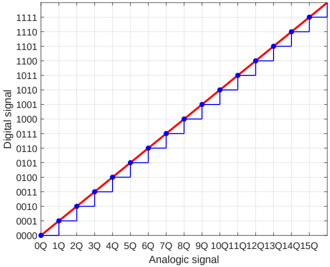
Transition threshold [1/2]

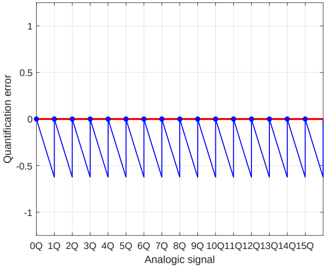
How do we switch from one digital value to another?

Solution #1: Switching from $N-1$ to N if $\dfrac{v}{Q} = N$.

All the analog values corresponding to the digital value N are:

$$NQ \leq v < (N+1)Q$$





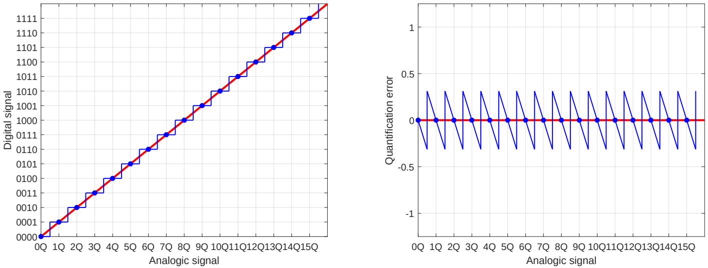
Issue: The mean quantification error is not equal to zero (presence of a bias).

Transition threshold [2/2]

Solution #2: Switching from $N-1$ to N if $\dfrac{v}{Q} = N - \dfrac{1}{2}$.

All the analog values corresponding to the digital value N are:

$$(N - \frac{1}{2})Q \leq v < (N + \frac{1}{2})Q$$



Conclusion: The mean quantification error is equal to zero and the maximum quantification error is $Q/2$.

Quantification

How to take into account quantification effects?

Model: In practice measured signals are numerous and **quantification** can be modeled as an **additive noise**.

$$\boxed{\text{Quantification error} \Leftrightarrow \text{Additive noise } \epsilon.}$$

$$NQ = v + \epsilon \quad \text{with} \quad -Q/2 \leq \epsilon < Q/2$$

Associated probability density function: Quantification noise probability uniformly distributed over $[-Q/2, Q/2]$.

$$\text{Then} \quad \int_{-Q/2}^{Q/2} p(\epsilon)d\epsilon = 1 \quad \text{and} \quad p(\epsilon) = \frac{1}{Q}$$

"Signal to noise" ratio (SNR)

Power: $P = \frac{V^2}{R_{in}}$ (R_{in}: acquisition interface input resistance)

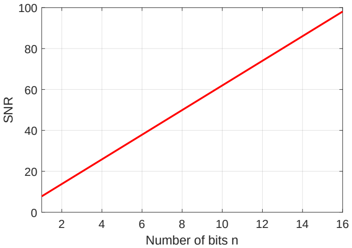
Signal: Sine of amplitude $V_p/2$: $P = \frac{1}{T}\int_0^T \left[\frac{V_p}{2}\sin(2\pi\frac{t}{T})\right]^2$

Noise:

$$V_b^2 = \int_{-Q/2}^{Q/2} \epsilon^2 p(\epsilon) d\epsilon = \frac{1}{Q}\int_{-Q/2}^{Q/2} \epsilon^2 d\epsilon = \frac{Q^2}{12} \quad (Q = \frac{V_p}{2^n})$$

$$\boxed{\text{SNR} = 10\log_{10}\left(\frac{V_p^2}{8V_b^2}\right) = 20n\log_{10}(2) + 10\log_{10}(4/3)}$$

The **larger** the **number of bits**, the **higher** the **signal to noise ratio** (SNR).



Practical case study: Acquisition board NI USB 621x

- What are the available scales?
- On how many bits is achieved quantification?
- What are possible quantum values?
- What is quantification noise SNR?

NI USB-621x Specifications

Analog Input

Number of channels
- USB-6210/6211/6212/6215/6216 .... 8 differential or 16 single ended
- USB-6218 .... 16 differential or 32 single ended

ADC resolution .... 16 bits

DNL .... No missing codes guaranteed

INL .... Refer to the *AI Absolute Accuracy Tables*

Sampling rate
- Maximum
  - USB-6210/6211/6215/6218 ... 250 kS/s single channel, 250 kS/s multichannel (aggregate)
  - USB-6212/6216 .... 400 kS/s single channel, 400 kS/s multichannel (aggregate)
- Minimum .... 0 S/s
- Timing accuracy .... 50 ppm of sample rate
- Timing resolution .... 50 ns

Input coupling .... DC

Input range .... ±10 V, ±5 V, ±1 V, ±0.2 V

Maximum working voltage for analog inputs (signal + common mode) .... ±10.4 V of AI GND

CMRR (DC to 60 Hz) .... 100 dB

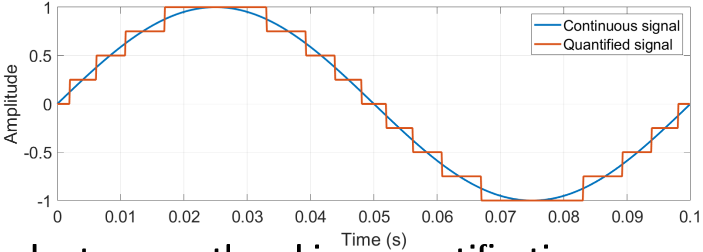
Input impedance
- Device on
  - AI+ to AI GND .... >10 GΩ in parallel with 100 pF
  - AI– to AI GND .... >10 GΩ in parallel with 100 pF
- Device off
  - AI+ to AI GND .... 1200 Ω
  - AI– to AI GND .... 1200 Ω

Input bias current .... ±100 pA

Crosstalk (at 100 kHz)
- Adjacent channels .... –75 dB
- Non-adjacent channels .... –90 dB

Small signal bandwidth (–3 dB)
- USB-6210/6211/6215/6218 .... 450 kHz
- USB-6212/6216 .... 1.5 MHz

Conclusions regarding quantification



In order to correctly achieve quantification:

- **Ensure that the number of bits is sufficient** (associated SNR higher that experimental SNR).
- Select the **adequate scale** according to your signals dynamic in order not to miss interesting information.
- Carefully **read the hardware documentation** to avoid troubles.

[Homeworks #5] Practicing quantification

1. Program a *"quantification"* function that take as input arguments the signal to be quantified, the number of bits, and the scale and that returns the quantified signal. Illustrate quantification and the effect of a badly chosen scale or number of bits on a sinusoidal signal. Load a recorded segment of speech and plot it. Increase the quantification noise by reducing the number of bits and listen to the result. Conclude regarding the effect of the number of bits.
2. What is the default MATLAB/OCTAVE choice for coding floating point numbers? Generate a random noise coded on 32 bits. Simulate the effect of decreasing the number of quantification bits. Plot the quantification error and its associated probability density function. Compute the SNR and demonstrate that it increases with 6 dB for each additional quantification bit.
3. Load an audio signal using the *"audioread"* function and get the associated number of bits using the "audioinfo" command. Determine graphically the quantum value for this audio signal by looking to two consecutive samples. Is this method reliable? Provide a reliable method for quantum and scale values estimations for this signal.

Bibliography

[1] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[2] J. Wu, J.and Wang and Z. You. An overview of dynamic parameter identification of robots. *Robotics and computer-integrated manufacturing*, 26(5):414–419, 2010.

[3] K. R. Kozlowski. *Modelling and identification in robotics*. Springer Science & Business Media, 2012.

[4] S. Guegan, W. Khalil, and P. Lemoine. Identification of the dynamic parameters of the orthoglide. In *2003 IEEE International Conference on Robotics and Automation*, volume 3, pages 3272–3277. IEEE, 2003.

[5] S. Bethge, J. Malzahn, N. Tsagarakis, and D. Caldwell. FloBaRoID: A software package for the identification of robot dynamics parameters. In *International Conference on Robotics in Alpe Adria Danube Region*, pages 156–165. Springer, 2017.

[6] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[7] Y. Meyer and P. Flandrin. Time-frequency/time-scale analysis. *Wavelet Analysis and its applications*, 1999.

[8] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[9] W. M. Hartmann. *Signals, sound, and sensation*. Springer Science & Business Media, 2004.

[10] S. K. Mitra and Y. Kuo. *Digital signal processing: a computer-based approach*, volume 2. McGraw-Hill New York, 2006.

[11] K. Shin and J. Hammond. *Fundamentals of signal processing for sound and vibration engineers*. John Wiley & Sons, 2008.

[12] D. Havelock, S. Kuwano, and M. Vorländer. *Handbook of signal processing in acoustics*. Springer Science & Business Media, 2008.

[13] M. Weeks. *Digital signal processing using MATLAB & wavelets*. Jones & Bartlett Publishers, 2010.

[14] Étienne Tisserand, Jean-François Pautex, and Patrick Schweitzer. *Analyse et traitement des signaux-2e éd.: Méthodes et applications au son et à l'image*. Dunod, 2009.

[15] Francis Cottet. *Traitement des signaux et acquisition de données: cours et exercices corrigés*. Dunod, 2020.

[16] Frédéric De Coulon. *Théorie et traitement des signaux*, volume 6. PPUR Presses polytechniques, 1998.

[17] Maurice Bellanger. *Traitement numérique du signal*. Ed. Techniques Ingénieur, 1981.

[18] Messaoud Benidir. *Théorie et traitement du signal: Cours et exercices corrigés. Méthodes de base pour l'analyse et le traitement du signal*. Dunod, 2004.