# Signal Analysis

Week

II

As we saw last week, sensors are the perception layer of our machines, interacting and actively reacting to the environment. This reaction to a given stimuli $s_i$ translates into an output signal $E_i$ following what we call the sensor transfer function $F(s)$. In particular, when designing a sensor for the end-user one needs to provide a calibration function $C(E_i) = s_m$ that for an output signal $E_i$ provides a measurement $s_m$ and which approximates the inverse of the sensor transfer function $F(s)$.



**Table 2.** Contents for Week II

Although it may seem from the last week (and from most of the sensors around us) that the output signal $E$ always corresponds to a voltage, that is not always true. In practice, during the design stage of a sensor[7], $E$ may correspond to any information extracted from the electric signal response of the sensor by using convenient processing techniques that explore its underlying physical principles. To give some practical examples:

[7] usually our work as physicists

> **Doppler radar:** The information of object velocity is encoded in the changes of reflected wave frequency due to the movement of the object.
> **Surface plasmon resonance based sensor:** The information is encoded in the wavelength/angle of highest absorption.

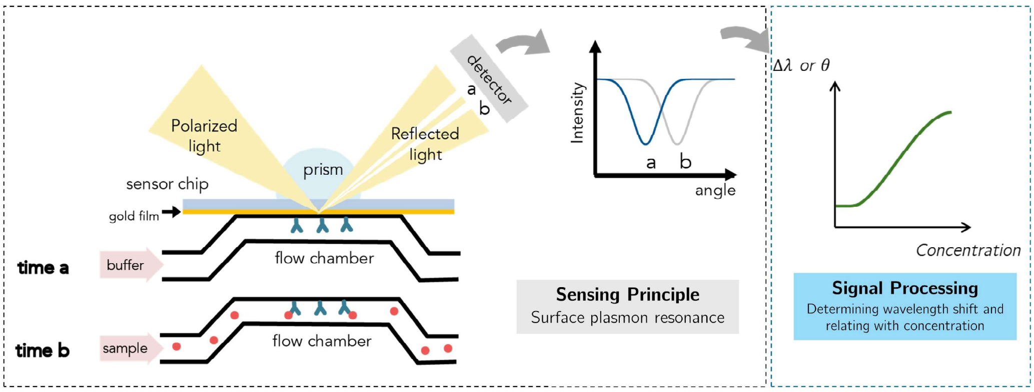



**Figure 10**. Sensing principle of a Surface Plasmon resonance sensor. The sensor is based on the fact that due to light confinement, plasmon coupling is extremely sensitive to the environment, with small changes in the refractive index near the surface translating into major responses. In this case, the presence of a distinct sample in the flow chamber modifies the refractive index and leads to a shift in the absorption band/angle.

Of course, most of the sensors we use every day are highly developed tools

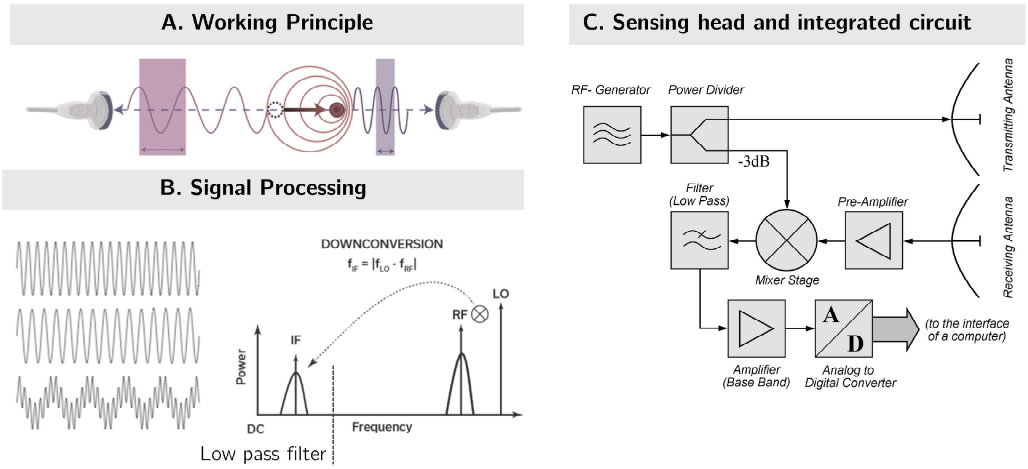



**Figure 11**. Doppler sensor working principle, signal processing, and circuit integration.

designed for agile integration with a processing unit, meaning that preferably all signal processing occurs in-chip (e.g. integrated circuit or even a micro-controller) to translate the output signal of the sensing head $E_i$ into a voltage output signal $E_i'$ for which a direct calibration function can be applied without processing. For example(see figure 11), while the information in the Doppler radar is in the frequency shift, a frequency mixer integrated circuit, followed by a low-pass filter, and a frequency-to-voltage converter can be utilized to convert the frequency shift to an output voltage signal $E_i'$ that is only then communicated to the user. Nevertheless, note that designing a sensor like this involves prior knowledge of the signal processing that is needed to extract the information[8] before being able to retrieve it with the converter.

[8] e.g. mixing and filtering

For this second week, our goal is to introduce some tools of the art of processing data from sensors, with the ultimate aim of extracting the information needed to construct a transfer function and finally a calibration model. First, we focus our attention on analog signals, where the nuances of proper statistical analysis of static and dynamic signals will be explored. Then, we will briefly discuss digital signals, highlighting their advantages and applications. We will describe some analog-to-digital conversion processes, focusing on the most common modulation techniques that are vastly utilized in engineering these days.

# 1 Analog Signals

As we said before, our goal is to extract meaningful information[9] from the output signal of the sensing head. Possible information contained in the signal largely depends on the physical process utilized as a sensing principle

[9] Which strongly relates to the concept of feature extraction in machine learning.

and we can divide it into major families: time-constant signals and time-varying signals.

## 1.1 Static Response

The static response of a sensor refers to its behavior under steady-state conditions, where the quantity being measured does not change over time. Knowing the applied stimulus (normally well-known standards), one can **determine the transfer function** of the sensor and deploy a **calibration model** by curve fitting procedures.

Towards this final goal, we must start by assigning a value $E_i$ for a given stimulus. At first glance, this may seem a simple task, apply $s_i$, register $E_i$, and compute the transfer function. However, such an approach largely overlooks experimental challenges of the real world, such as noise and variability.

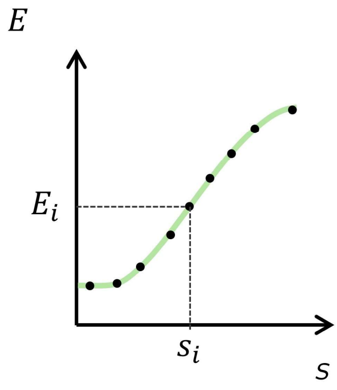
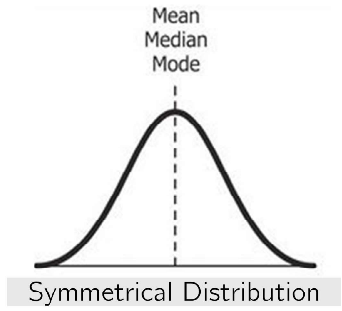



**Figure 12**. Experimental determination of the transfer function.

**Definition. Noise** refers to the random fluctuations that can distort the output signal of a sensor. It can originate from various sources, including **external noise** (when it comes from the environment), **internal noise** (when it comes from within the sensor and electronics itself), or **quantization noise** (from the analog-to-digital conversion procedure).

Solid experimental procedures to determine the transfer function shall then involve recording multiple values of $E_i^{(j)}$ for $j \in \{0, .., N\}$ to a known stimulus $s_i$, preferably over various runs. Having a set of measures for the same stimuli $s_i$ it is now possible to extract a good estimate $E_i$ utilizing statistical procedures.



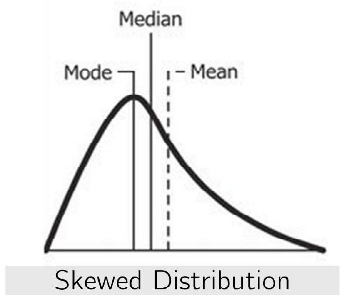




**Figure 13**. Central tendencies in symmetric and asymmetric distributions.

### 1.1.1 Central Tendency

The first information that we can extract is the estimated value $E_i$ itself, which may be obtained in the form of a central tendency[10]. While many measures of central tendency exist, a good rule of thumb is to analyze the distribution of registered $E_i^{(j)}$ values and choose:

[10] *tendency of quantitative data to cluster around some central value*

**Mean:** if the data is normally (or symmetrically) distributed and there are no significant outliers.

**Median:** if the data distribution is skewed (e.g. has a long tail on one side), if there are significant outliers (e.g. if anomalies are common), or if you want a more resilient measurement.

**Mode:** if the data is of categorical type (e.g. On/Off).

Less utilized statistical measures include geometrical, weighted, trimmed, and weighted means. These are utilized in very specific situations which although they may appear in a physics laboratory, usually do not appear in the context of sensors.

### 1.1.2 Uncertainty

Now that we have a central tendency, it would be important also to quantify the lack of exactness of the $E_i$ considering its variation, which relates to the concept of **uncertainty**.

In science, the word **uncertainty** is not necessarily associated with a mistake but rather with the lack of exactness of a measurement. The fact is that it is impossible to measure with infinite accuracy and precision[11] even if we have the best equipment and conditions available.

[11] Accuracy - how close a measured value is to the true value. Precision - how close the measurements of the same item are to each other;

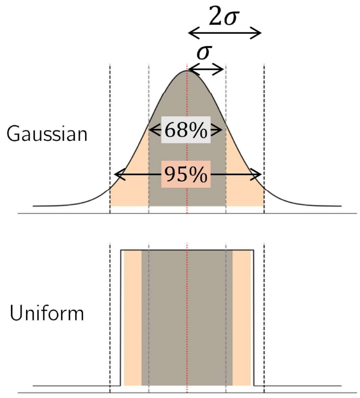



**Figure 14**. Standard deviation and Interquartile range.

Uncertainty encompasses all potential errors in the measurement process, which are categorized into:

- **Type A Uncertainty**: Related to statistical variation of measured data (often associated with random noise) and thus can be estimated by statistical methods.
- **Type B Uncertainty**: Related to all other sources of uncertainty in the measurement process, such as properties of the instrument and environmental conditions. These are not necessarily random and are often evaluated by non-statistical means.

There are multiple manners to estimate the uncertainty of type A, $u_A$ being the most common:

**Standard deviation** is the preferred method for computing Type A uncertainty because it takes into account the variability of all measurements and provides a basis for further statistical analysis. In particular it may define a **prediction interval** for a normal distribution: e.g. the interval $[\bar{E}_i - \sigma_i, \bar{E}_i + \sigma_i]$ contains around 68% of the registered values, whereas $[\bar{E}_i - 2\sigma_i, \bar{E}_i + 2\sigma_i]$ contains around 95%.

**Maximum deviation**, might be used for quick assessments or when the data set is very small, but it does not provide as detailed an understanding of the data variability.

**Percentiles and Interquartile Range**, when the data distribution is significantly non-normal or to mitigate the presence of outliers.

For type B uncertainty $u_B$, one should consider all known or estimated sources of uncertainty other than those derived from statistical variation.

This can include uncertainties due to instrument calibration, environmental factors, assumptions in the measurement process, and more. Each source of Type B uncertainty is quantified, usually as a standard deviation equivalent, based on the best available information.

Once you have quantified Type A and Type B uncertainties, you combine them to calculate the combined standard uncertainty. Assuming that all uncertainties are independent, the combined standard uncertainty is calculated using the square root of the sum of the squares of the individual standard uncertainties:

$$u_i = \sqrt{u_A^2 + u_B^2}. \tag{1.1}$$

Your output signal is then represented as $E_i \pm u_i$[12].

[12] Use 2 significant figures, according to SI regulations.

### 1.1.3 Signal-to-noise Ratio

Signal-to-noise Ratio (SNR) is another common measure used in science and engineering in the context of sensors. It calculates the ratio of signal power to noise power, **being an indicator of how much the signal may be corrupted by noise**. A higher SNR indicates a cleaner signal with less background noise, whereas a lower SNR indicates more noise in comparison to the signal.

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{1.2}$$

for power or

$$SNR = \frac{A^2_{signal}}{A^2_{noise}} \tag{1.3}$$

for amplitude signals. Normally, as it covers it may be expressed in decibels (dB) being

$$SNR = 10 \log_{10} \frac{P_{signal}}{P_{noise}} \text{dB} \tag{1.4}$$

when involving powers, or

$$SNR = 20 \log_{10} \frac{A_{signal}}{A_{noise}} \text{dB} \tag{1.5}$$

when involving amplitudes.

So you can utilize our previous estimate $E(s_i)$ obtained from central tendency as the numerator for any of these formulas. **But what should you use in the denominator?**

The most common is to utilize one of two methods:

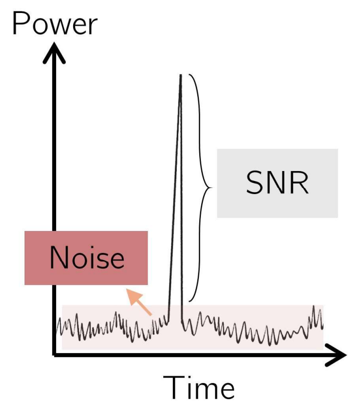



**Figure 15**. Signal-to-noise Ratio.

**Direct Measurement**: For electronic sensors, you can, when possible, directly register the signal in a quiet (noise-free) environment. This measurement gives you the peak-to-peak noise voltage or current, from which you can calculate the RMS (root mean square) or the standard deviation for the noise signal.

**Statistical Analysis**: if you cannot perform direct measurement of the noise level, and you already estimated type A uncertainties, you can usually utilize the standard deviation as a measure of the sensor random noise.

To choose what to use in real-world scenarios you must analyze your situation and see what choices you have. But more importantly than that is to **to present the utilized formula**. This is usually sufficient to maintain the scientific accuracy of your works.

### 1.1.4 Analyzing the Transfer function and Calibration Function

With the values $E(s_i) + u_{E(s_i)}$ calculated for a given stimulus set $\{s_i\}$ we are now ready to construct the **Transfer function** for our sensor. As we saw last week, the transfer function allows us to characterize our sensor in terms of meaningful quantitative and qualitative parameters namely:

**From direct graphical analysis:**

- Span, Full-scale Output, Dynamic Range, and Deadband;
- Hysteresis, linearity, and saturation;

**From numerical analysis:**

- Sensitivity, by computing numerical derivatives;

**From multiple runs:**

- Repeatability, Stability (if you measure various runs over a large period of time).

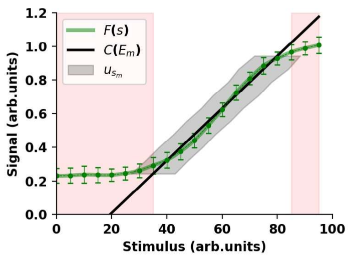



**Figure 16**. Experimental determination of a transfer function and calibration.

Besides, having the transfer function set we are now in conditions to build the calibration curve, i.e. a mathematical model that we will use in the real world to estimate a measurement $s_m$ if we record the value $E_m$ at the output signal. Again, is up to you to decide which analytical model is best. Yet it is always a good practice[13] to prefer **linear models** when possible,

$$C(E) = mE + b. \tag{1.6}$$

[13] and following Occam's razor principle

If you need to utilize a nonlinear model, opt for the **lowest number of**

**fitting parameters**[14].

To estimate the value of parameters you need to perform what it is called in statistics and data science a numerical **regression**. A regression task provides estimates for the parameters of a given mathematical model by minimizing a loss function[15].

These days you have multiple computational and statistical libraries available (in *python* you have for example *scipy curve_fit* function or the *llmfit* library) that may provide you with convenient parameter estimation and related uncertainty. Besides, most of the tools allow you to provide uncertainty for each point, performing a weighted regression which exploits the variability of your data[16]. Then, if you want to estimate the uncertainty on your final measurement $s_m$ you may utilize the propagation of uncertainties (assuming uncorrelated variables)

$$u_{s_m} = \sqrt{\left(u_{E(s_m)}\frac{\partial C}{\partial E}\right)^2 + \sum_n \left(u_{a_n}\frac{\partial C}{\partial a_n}\right)^2} \tag{1.7}$$

where $a_n$ are the parameters of the model estimated during curve fitting. Note that you do not only the uncertainty of the parameters of the fit but also of the output signal. This is expected but often disregarded as it is not trivial to compute the uncertainty $u_{E(s_m)}$ if $s_m$ is outside the calibration set $\{s_i\}$. In that case, you may either construct an uncertainty model (e.g. spline interpolation), take the maximum uncertainty, or resort to some more advanced statistical methods.

Finally, you can compare the calibration curve and the transfer function to estimate the **accuracy** $\Delta$ of your sensor. As we saw previously, a good[17] estimate is to **provide the maximum distance between the calibration curve and the transfer function**. Nevertheless, you should also consider the uncertainty previously determined as $u_{s_m}$. If the uncertainty is larger, it would be preferable to opt for this value as an estimate of the accuracy. Note that we are now in a condition to better estimate other parameters of our sensor such as **sensitivity**.

**Exercise 3. Computational Activity:** Complete the activity in the Jupyter notebook for Week II that revolves around time-constant signals and constructing a transfer and calibration function.

**Exercise 4. Hands-on Activity:** With an Arduino, a CNY70 distance sensor, and a linear stage to precisely monitor the distance, follow the same strategy to create a transfer function and calibration function, and provide some specifications, for this simple yet real-world scenario.

[14] With four parameters I can fit an elephant. - John Von Neumann

[15] such as the sum of squared residuals $\sum(C(E(s_i)) - s_i)^2$, as in the least squares method, but not exclusively

[16] e.g. data with lower uncertainty are benefited

[17] and safer!

## 1.2 Time-Varying Signals

Time-varying signals or simply time signals correspond to all the situations when the amplitude of a signal varies with time. Time signals can be separated into **deterministic** and **non-deterministic** signals. Deterministic signals follow an analytical formula or equation and can be either periodic or non-periodic. Non-deterministic are stochastic in nature and can be either stationary or non-stationary.

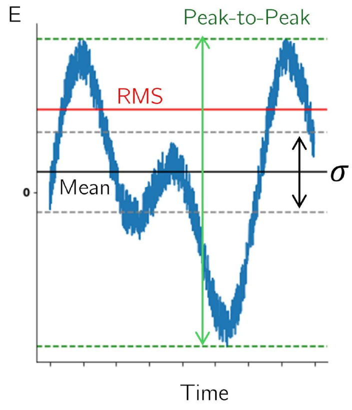



**Figure 17.** Some time-domain features.

### 1.2.1 Time-domain Features

**Mean and median:** The average and middle value of the signal.

**Standard Deviation and variance:** measures the dispersion of the signal, with the variance being the square of the standard deviation.

**Root Mean Square (RMS):** Square root of the average of squared values. It is useful for assessing the power of the signal.

**Peak-to-Peak:** Difference between the maximum and minimum values in the signal.

**Skewness:** Measure of the asymmetry of the signal distribution.

### 1.2.2 Frequency-domain Features

Fourier transform is one of the most important and universal tools of mathematicians, engineers, and physicists. In general, it can be utilized for analyzing the frequencies contained within a signal or a function and its essence lies in its ability to decompose a complex signal into its constituent sinusoidal waves of varying frequencies, amplitudes, and phases.

Mathematically, the Fourier transform $\mathcal{F}$ of a signal $f(t)$ is a function $F(\omega)$ given by the integral

$$\mathcal{F}[f(t)] = F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{1.8}$$

and transforms a time-domain signal into its dual frequency-domain representation. The signal can be recovered using the inverse Fourier transform

$$\mathcal{F}^{-1}[F(\omega)] = f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \tag{1.9}$$

The Discrete Fourier Transform (DFT) is a variant of the Fourier Transform that is utilized for analyzing discrete signals. Its computational implementation is the Fast Fourier Transform (FFT), one of the most impactful computer algorithms of all time[18]. There are also two-dimensional Fourier Transforms used for image processing and many more specialized forms.

Performing the Fourier Transform of the signal we can extract a few important features of our time signal which may contain relevant information for our sensor:

**Spectral Centroid:** The center of mass of the spectrum, being defined as $c = \sum_f fP(f) / \sum_f P(f)$ is the spectral centroid and $P(f)$ is either the magnitude (i.e. absolute value) of the Fourier Transform or its Power Spectral Density (see below).

**Fundamental Frequency:** The lowest frequency of a periodic waveform, determines its pitch.

**Spectral Bandwidth:** Usually the width of the band at one-half the peak maximum but in signal processing is common to define the spectral bandwidth as

$$BW = \sqrt{\frac{\sum_f (f-c)^2 P(f)}{\sum_f P(f)}} \tag{1.10}$$

where $c = \sum_f fP(f) / \sum_f P(f)$ is the spectral centroid and $P(f)$ is either the magnitude of the Fourier Transform or its Power Spectral Density (see below). This definition gives a more precise measure of the bandwidth as it accounts for the distribution of energy across the spectrum.

**Spectral Entropy:** Entropy of the spectral distribution, indicating its randomness. Usually computed as the Shannon entropy of the power spectral density (PSD).

At this point, we should note that in real-world applications noise may be present in the signal. In this scenario the Fourier transform and its magnitude may be insufficient to get a clear picture of the signal under analysis. In this context, the Power spectral density is a much more resilient tool:

**Power Spectral Density (PSD):** Measure of signal power in function of the frequency. The PSD is mathematically equivalent to the Fourier Transform of the autocorrelation function of the signal

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-2\pi i f \tau} d\tau \tag{1.11}$$

and can be considered as a normalized or averaged version of the squared magnitude of the Fourier Transform over multiple instances or realiza-

18 A nice video can be found here.

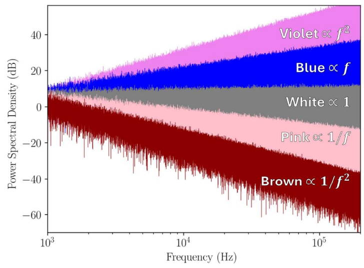



**Figure 18.** Analyzing the type of noise with Power Spectral Density.

tions of a process, especially in the context of random or time-varying signals. It is important to stress that **PSD is not equivalent to the magnitude of the Fourier transform**. It is rather utilized to analyze the power distribution over frequency of signals that are often considered over longer durations or as processes, being thus particularly useful when there is noise in the signal, as well as understanding noise properties and its characteristics.

Magnitude of the Fourier Transform corresponds to the Energy, not the Power.

As a rule of thumb, you may utilize the magnitude spectrum from the Fourier transform for identifying *what frequencies are present*. Yet, in the presence of noise, estimating the above quantities may benefit from using PSD as it answers *how is the signal's power distributed across frequencies*.

### 1.2.3 Time-frequency Features

The Fourier transform considers the signal as an infinite duration series and decomposes it into sinusoidal components. Yet, sometimes the frequency signature of the output signal may vary in time[19] which can be analyzed in the form of a **spectrogram** - frequency *vs* time.

[19] e.g. a Doppler sensor.

To construct a spectrogram we may use **short-time Fourier Transforms** (STFT) which is a mathematical technique used to analyze signals whose frequency components change over time. It combines the concepts of the Fourier Transform with a sliding time window to provide a two-dimensional representation of a signal, showing how its frequency content evolves. There are however two major problems of STFT, the computational load and the Heisenberg Uncertainty, which limits the simultaneous precision of measuring both time and frequency properties of a signal.

A possible workaround appears as **wavelet transforms**, mathematical tools that decompose a signal into components that are localized in both time and frequency. Wavelets are some kind of a **mathematical microscope** that you can utilize to analyze signals in detail, in particular those that have non-stationary characteristics (such as audio signals or images just to name a few examples). Wavelets are also very useful for analyzing **very large datasets** in an efficient manner.

To briefly grasp the working principles of wavelet transforms, the continuous wavelet transform of $f(t)$ may be defined by the coefficients

$$[W_\psi f](a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \cdot \psi\left(\frac{t-b}{a}\right) dt \qquad (1.12)$$

with $a$ a scale parameter and $b$ a translation parameter. The mother wavelet $\psi$ is a function localized in time and frequency, and thus, by varying scale $a$ (scanning in frequency) and time with $b$(scanning in time), it is possible to

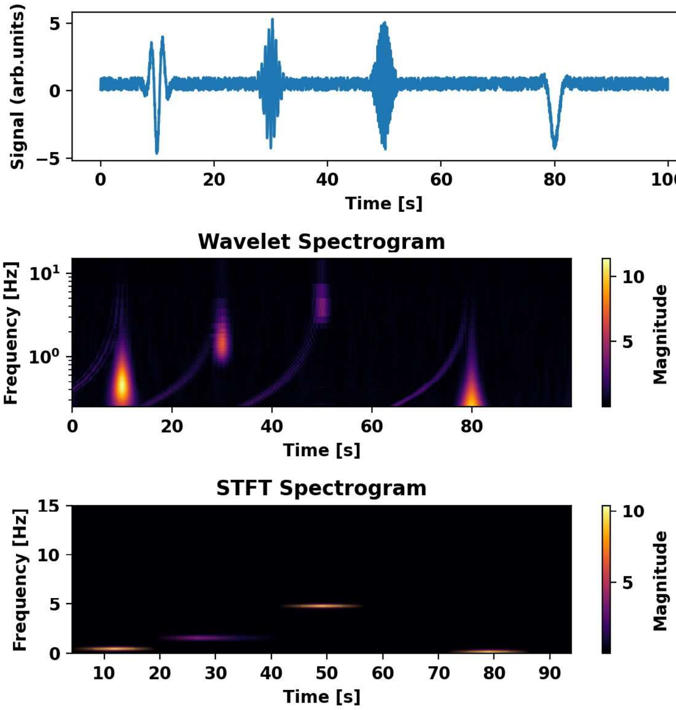



**Figure 19**. Spectrograms analysis with wavelets and short-time Fourier transform, highlighting the major problems of both: wavelets provide better time characterization, especially at higher frequencies, but do that at the cost of additional artifacts and non-linear scale in the frequency.

transform a signal into a localized distribution of frequencies in function of time via a **multi-resolution analysis**.

In practice, computational implementations utilize a discrete version - discrete wavelet transform - which utilizes a discretized computation of the integral to obtain the wavelet coefficients

$$c_{jk} = [W_\psi f](a = 2^{-j}, b = k2^{-j}) \tag{1.13}$$

with $a = 2^{-j}$ known as binary dilation and $b = k2^{-j}$ the binary position. This formulation is particularly useful in the case of an orthonormal wavelet, for which it is possible to recover the original signal from

$$f(t) = \sum_{j,k=-\infty}^{\infty} c_{jk} 2^{j/2} \psi(2^j x - k). \tag{1.14}$$

As $k$ works as a linear translation in time, analyzing the variation of these coefficients at each $k$ (e.g. centroid, bandwidth) may allow to extract of usable information in the context of sensing.

### 1.2.4 Statistical and correlation Features

So far we have focused on properties that are more useful in deterministic signals, whether these are periodic or non-periodic. If the signal is stochastic, however, the tools introduced previously may lack the necessary information. It is in this particular context that correlation appears.

Correlation, especially in the context of time-varying signals, is a fundamental tool in signal processing that measures the similarity or relationship between two signals as a function of the time-lag applied to one of them.

**Correlation**: Measures the similarity and relationship between two signals, mathematically defined

$$R_{fg}(t) = \int_{-\infty}^{\infty} f^*(t)g(t+\tau)dt \tag{1.15}$$

**Auto-correlation:** if $f = g$ and useful to identify periodic signals or the fundamental frequency of a signal;
**Cross-correlation:** if $f$ and $g$ are distinct, and useful for detecting if the signals are similar with a given time delay between them(similarly to pattern recognition).

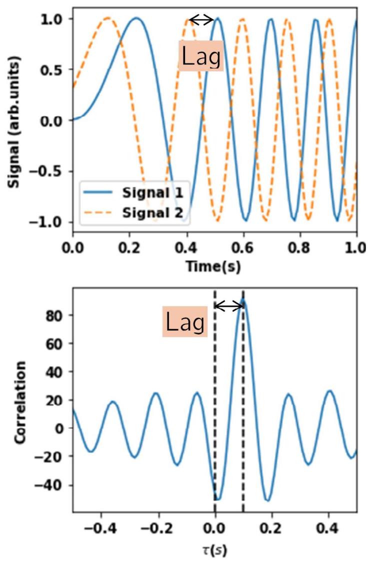



**Figure 20**. Determination of Lag using cross-correlation.

Some key information that can be extracted:

- **Time delay estimation:** if two signals are similar with a lag, the time delay can be extracted as a maximum appearing in the correlation signal;
- **Signal Detection and Pattern recognition:** correlation is used to detect known signals within noisy environments, which is crucial for demodulating transmitted data in communications. Here, a known signal pattern (such as a **preamble** or **pilot** signal) is correlated with the received signal to detect the presence of data or to synchronize the receiver.
- **Noise reduction:** In periodic signals, correlating the signal with itself (autocorrelation) allows to enhance signal features while reducing noise, which does not correlate over time.

## 1.3 Signal Pre-Processing

Week

III

Signal pre-processing is a critical step in the analysis and interpretation of data obtained from sensors. The goal of signal pre-processing is to improve the quality of the signal for further analysis, making it easier to extract useful information. We will focus on the most common challenges and important techniques, focusing on those that can be implemented both at the digital level as well as those that shall be applied at the analogic level.

### 1.3.1 Sampling Rate adjustment

Choosing a correct sampling frequency, i.e. frequency at which data is acquired from a sensor, is of major importance for correctly assessing time-varying data. To warrant good results, the criteria you should follow is given by the **Nyquist-Shannon theorem** that states that for recovering a signal at frequency $f_0$ you should sample at least at frequency $f_s \geq 2f_0$. Any value below this number will give an estimate of a frequency below the true frequency, an effect that is called **aliasing**.

An important thing you should note is that if additional frequencies are present, fake alias peaks coming from higher-frequency signals may appear in the spectrum. As this is not possible to verify prior the only way to prevent this effect is to apply spectral filtering (a low-pass, Which needs to be performed by hardware, before sampling).

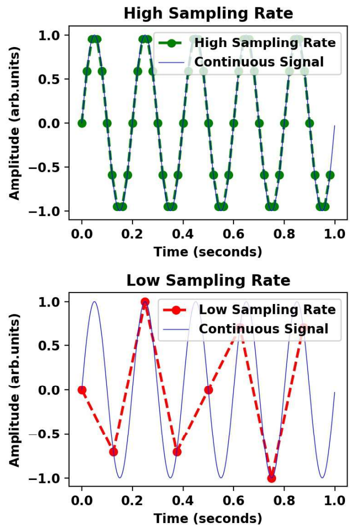



**Figure 21**. Aliasing.

### 1.3.2 Signal Filtering in the spectral domain

Filters are utilized to remove variations of the signal that do not contain relevant information to the sensor operation. Usually, filtering unwanted frequencies is one of the most common tasks in signal processing, in particular when we know the spectral range where the signal information is located. In this context low-pass, high-pass, band-pass, and band-reject filters are of particular importance. After sampling the data applying any kind of spectral filter in the Fourier domain is straightforward, and ideal filters can be created on-demand.

Yet, it can be inefficient in some cases, like removing aliasing effects as seen in the previous section, or if one needs real-time processing. In this context, one needs to apply filtering in the electrical domain. The Butterworth low-pass filter (see figure 22) is perhaps one of the most popular electronic filters, designed to filter out signal components above a certain frequency. Its gain function is given by

$$G(f) = \frac{1}{\sqrt{1 + (2\pi f)^{2N}}} \tag{1.16}$$

with $N$ the order of the filter[20]. Having higher $N$ means sharper and better filtering, but comes at the cost of higher complexity of implementation.

[20] which translates into the number of reactive elements - inductors and capacitors - in the circuit.

### 1.3.3 Smoothing in time domain

Smoothing is a type of filtering that is applied in the time domain and thus may impact the whole frequency range of a signal. The most used smoothing filters include:

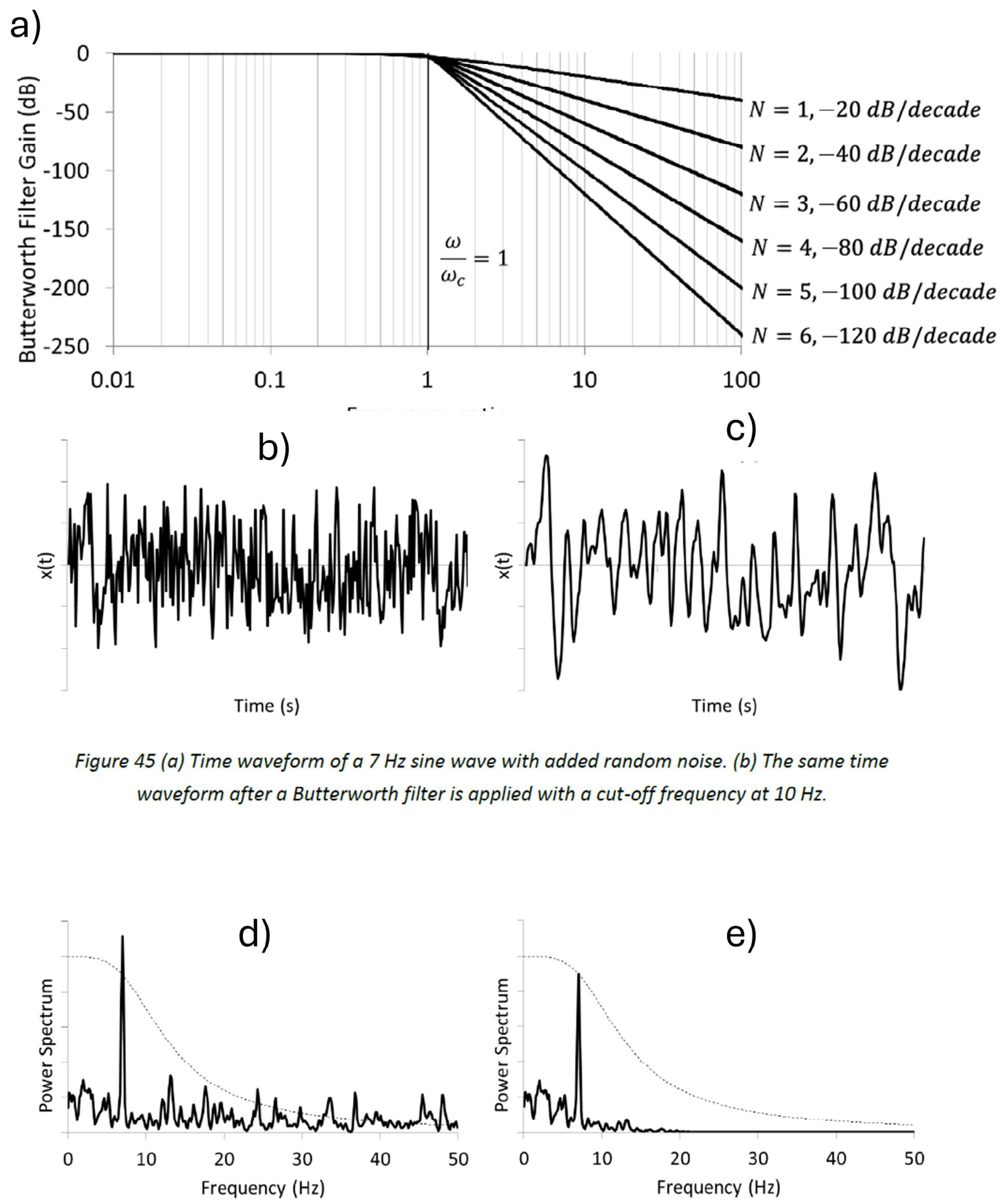



**Figure 22**. a)Butterworth filter gain function. b) Unfiltered and c) Filtered signal. Spectral analysis of the d) Unfiltered and e) Filtered signal.

**Moving Average:** Each point is replaced by an average of the neighboring data points, resulting in an effect that resembles a low-pass filter but that may attenuate the amplitude of the existing harmonics.

**Savitzky-Golay:** Each point is replaced by a generalized weighted average of the neighboring data points, whose coefficients (an odd number of them) are pre-computed and relate to the local polynomial fit. Compared

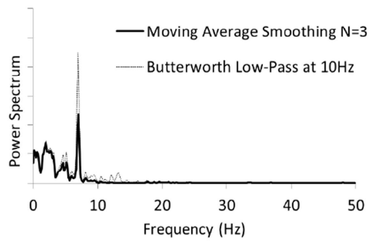



**Figure 23**. Moving average filter performance.

to the moving average it may preserve more of the original amplitude of the signal, but its performance in filtering noise may be lower.

### 1.3.4 Windowing and Leakage

Fourier analysis assumes that the signal is infinite, i.e., even if the signal is limited in time to a maximum number of samples, Fourier transform works as if that signal repeats over time. This introduces some problems when the measurement period is a non-integer multiple of the period of the signal frequency, which happens in typical conditions as perfect synchronization is not possible.

In this scenario, for the FFT algorithm the signal is no longer a pure harmonic signal and the true peak power tends to **leak** to other frequencies.

A way to mitigate leakage is to previously multiply the signal by a function - **window** - which forces the decay of the signal to zero in the limits of the acquisition period. There are multiple options for windowing, being the most utilized the Hamming, Hanning, and Blackman-Harris.

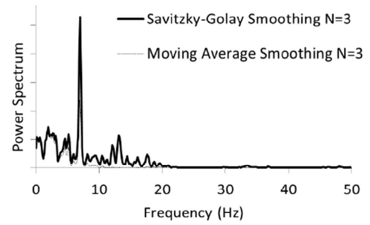



**Figure 24**. Savitzky-Golay filter performance.

## 2 Digital Signals

As we saw in the first week some sensors output analog signals while others prefer digital. This choice depends on various intertwined design preferences, from application requirements, to complexity, and even cost.

As a rule of thumb, **digital signals** may be preferable when

- **Communication:** digital signals are more resilient to degradation and thus easier to transmit over longer distances.

- **Integration:** If your sensor needs to interface with computers or digital systems, a digital output might be more straightforward to integrate.

- **Cost and Complexity of Digital Circuitry:** Digital circuitry can sometimes be cost-effective and compact, offering smart features like built-in calibration, error detection, and communication protocols. Besides, bypassing the need for an analog-to-digital converter is an important step in reducing the overall cost of the solution.

Analog signals may be preferable for the **design stage**, as well as **simpler applications**, **real-time applications**, or **high-resolution and sensitive applications**.

## 2.1 Typical conversion schemes

There are many analog-to-digital conversion schemes and they are not the main scope of this curricular unit. As such, we will just briefly mention two of the most used strategies: Pulse Width Modulation (PWM) and Pulse Density Modulation (PDM).

**Pulse Width Modulation (PWM):** PWM controls the power to devices by varying the width of the pulses in a fixed frequency signal. The average power delivered is proportional to the pulse width and thus a wider pulse increases the average voltage supplied to the load, while a narrower pulse decreases it. It is used in the control of motors or LEDs, simpler to control but re-conversion to analog may suffer higher levels of distortion.

**Pulse Density Modulation (PDM):** PDM represents an analog signal with the relative density of pulses in a sequence. Instead of varying the pulse width, PDM varies the number of pulses in a given time frame. A higher density of pulses represents a higher signal level, and a lower density represents a lower signal level. It is widely used in audio applications, such as digital microphones, as it offers higher resolution and is directly compatible with digital signal processing techniques. Yet is also more complex to implement and decode.

# 3 From sensors to automation: a brief note on controllers

Controllers are integral components of control systems used to regulate the behavior of other devices or systems. They interpret input data, often from sensors, and provide appropriate output commands to achieve desired system behavior. The objective is to maintain the system's output within a desired range, despite disturbances or changes in the environment. Among various types of controllers, Proportional-Integral-Derivative (PID) controllers, Fuzzy Logic controllers, Model Predictive Controllers (MPC), and Neural Network Controllers, just to name a few. For this curricular unit, we will focus on the most common PID controller.

A PID controller is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications that require continuous monitoring and control. The working principle relies on the computation of an error value $e(t) = S - V$ as the difference between a desired setpoint $S$ and a measured process variable $V$, leading to an output value

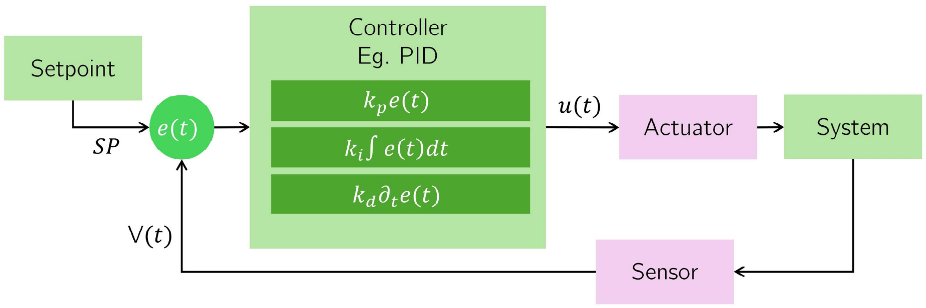



**Figure 25**. Conceptual role of a controller (e.g. PID) in a system monitored by a sensor.

to be passed to the actuator based on proportional, integral, and derivative terms, denoted by $P$, $I$, and $D$ terms:

- **Proportional (P)**: a proportional term to the current error value, $P = K_p e(t)$, acting as an immediate response to the deviation.
- **Integral (I)**: The integral term is concerned with the accumulation of past errors, being given by $I = K_i \int_0^t e(t)dt$. It acts as an accelerator of the movement depending on the accumulated error allowing a softer response closer to the setpoint.
- **Derivative (D)**: The derivative term is a prediction of future errors, based on the rate of change of the error and given as $D = K_d \partial_t e(t)$. It can provide a damping effect that reduces overshoot and improves system stability.

The output of the PID controller is thus given by $u(t) = P + I + D$. Note that the gain values shall be optimized for each system to reach the setpoint as fast as possible without significant overshoot or oscillations, and with minimal steady-state error. This can be done using methods from manual tuning, for tuning these parameters, including manual tuning, Ziegler-Nichols heuristics, or software-based optimization techniques.

# 4 Machine Learning and Sensors

The recent advent of machine learning has fostered significant developments in the way that we collect and analyze data gathered from sensors. In particular, machine learning[21] offers an interesting potential for dealing with more complex scenarios in particular:

[21] to be precise, numerical regression is already a type of machine learning!

- **Highly nonlinear and multiparameter responses:** When you have multiple parameters you can have highly complex and nonlinear transfer

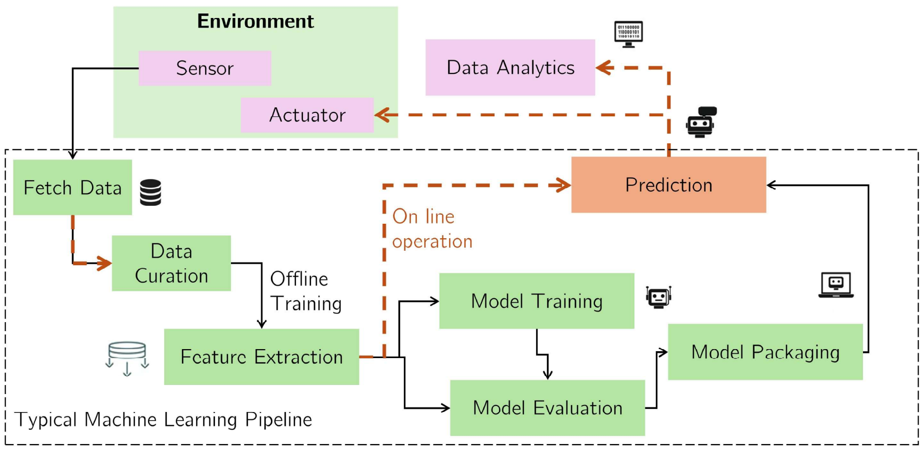



**Figure 26**. Typical Pipeline of Machine learning and a conceptual example of its application with a sensor.

functions. In this case, function fitting may be challenging to achieve [22]and data-driven regressions such as K-Nearest Neighbors regression, Support vector machines, or multilayer neural networks may be helpful to construct reliable calibration models.

[22] you would need a good estimate of a prior mathematical model

- **Sensor fusion:** Combining the information from multiple sensors is a particularly interesting topic of research. You can combine multiple measures with similar sensors (like taking a photo from distinct angles) but you can also combine measurements from very distinct sensors (e.g. combining distinct techniques, as different as sound and reflectance).
- **Data Preprocessing:** It is possible to train multiple algorithms to learn preprocessing tasks, from noise filtering to feature extraction that can be applied directly to the raw sensor data.
- **Automated Pattern Recognition:** Sometimes the information provided by sensors is extremely complex (e.g. unusual vibrations detected by installed accelerometers) to recognize complex patterns. This has many applications in industry fields such as predictive maintenance, where active maintenance strategies may be implemented to reduce downtime, predict failures, and ultimately save costs. Other interesting applications are healthcare and wearable sensors.

Before ending the discussion it is important to leave some considerations so you can learn from previous mistakes of others. Machine learning is a great tool but it is not a silver bullet and should be used only in a very informed way[23].

[23] *Little learning is a dangerous thing.*

First, regarding models, it is always preferable to, when possible, use the standard calibration curve procedure as presented before this week. Machine learning virtues can often be deceiving, and being easy to train a simple regression may seem. Yet this approach may overlook some problems (e.g. outliers) that can be circumvented otherwise. If not learning models that relate to the sensing principle to black-box machine learning models.

Secondly, regarding data, take into consideration the most important saying of Machine learning: **Garbage in, Garbage out**. This saying synthesizes that the effectiveness of ML models heavily depends on the quality and quantity of the data collected by sensors. So, when possible, informed feature extraction procedures are always preferable to naive brute-force approaches. Besides, take into consideration the size of the input data that you use to train the model, which should be much larger than the number of free parameters to avoid overfitting.

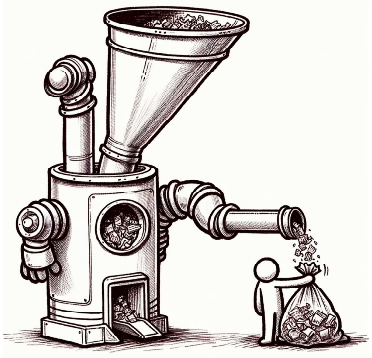


**Figure 27**. Garbage In, Garbage Out: perhaps the most important concept of Machine Learning.

Finally, consider that complex machine learning models may require the use of microprocessors or even processors. In this case, be careful to use energy-efficient algorithms and consider the possibility of performing the computing closer to the sensing head[24], avoiding data transmission and associated latency.

[24] usually referred to as *edge computing*

**Exercise 5. Hands-on Activity:** In addition to the signal analysis procedures there is a Jupyter notebook that concerns sound signal analysis and classification. There, we explore how to use a simple sensor and a machine learning system to deploy a simple classifier of spoken words.

# 5 Concluding Remarks

In this second week, we focused on signal processing, i.e., extracting good and meaningful information out of the output signal of a sensor. We covered the analysis of static signals, in particular focusing on the topic of distributions and uncertainties, allowing us to measure a transfer function from the extracted information and to build a calibration model. Then, focusing on time-varying signals, we introduced tools to extract information that can be used for the purpose. We briefly mentioned some topics on signal filtering, digital signals, and machine learning, so you can have a wide perspective of their role in modern sensors.

So far you already know:

- What is a sensor and how to characterize one;
- How to extract information from the output signal of a sensor and construct a calibration function;

and therefore we are ready to start building some of the most advanced and sensitive sensors: interferometric optical sensors.