Chapter 2
From Metrology to Digital Data

2.1 Measure and Measurements

The operation of measuring an unknown quantity x_o can be modeled as taking an instance—or measurement—x_i at time i with an ad hoc sensor S. Although S has been suitably designed and realized, the physical elements composing it are far from ideal and introduce sources of uncertainty in the measurement process. As a consequence, x_i represents only an estimate of x_o. In extreme cases, the value of x_o might not even exist [109] or simply cannot be measured, e.g., think of the Heisenberg's principle of uncertainty stating that it is not possible to exactly measure both the momentum and the position of a particle [112] with arbitrary accuracy.

As a consequence, despite the intuitive formalization of the measurement process, several major aspects need to be investigated and addressed before claiming that a generic measurement x_i is a an accurate and reliable estimate of x_o. For instance, we would rather require subsequent measurements x_i to be somehow centered around x_o, where centering must be intended according to a chosen figure of merit. In other words, we are requesting an accurate sensor that does not introduce some bias error (*accuracy* property). Then, we hope that the sensor is able to provide a long sequence of correct digits of the number associated with the acquired data. Clearly, a weight sensor able to perceive variations of 1 mg is better than a scale providing a resolution of 10 g (*resolution* property). Finally, each measurement represents only an estimate of the true unknown value, the discrepancy between the two—or error—depending on the quality of the sensor and the working conditions under which the measure was taken (*precision* property). Note that we might have an accurate sensor with a high resolution but a poor precision associated with the measurement process, yielding to a poor measurement. Moreover, we might have a precise measurement acquired with a high resolution sensor, again yielding a poor outcome whenever the sensor is not accurate.

There are other properties we should look at when considering a sensor, e.g., repeatability. Repeatability requires that subsequent measurements acquired in the same operational conditions should be indistinguishable within the uncertainty level

C. Alippi, *Intelligence for Embedded Systems*, DOI: 10.1007/978-3-319-05278-6_2,

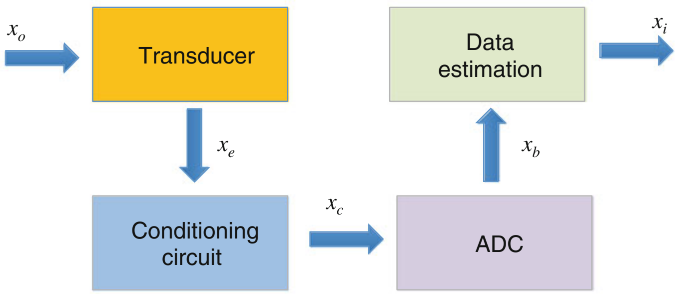

Fig. 2.1 The complete measurement chain of a sensor. The key elements are the transducer, converting an unknown physical entity x_o into the analog electrical entity x_e, the conditioning stage providing an improved analog value x_c, the ADC converting the analog value x_c to a binary codeword x_b, and the final data estimation module leading to the output value associated with the data instance x_i

associated with the sensor. For an in-depth analysis of metrological aspects readers can refer to [180, 182].

In the chapter, we introduce the main actors taking part in the measurement chain which leads, from the physical quantity to be measured x_o, to the final value x_i to be used in the subsequent data processing and decision-making phases. In the following, the measurement framework will be suitably modeled and the properties we expect from the retrieved data formalized.

2.1.1 The Measurement Chain

The main functional elements composing the measurement chain carried out by a sensor are the transduction module, the conditioning circuit, the Analog to Digital Converter (ADC), and the final data estimation module. Figure 2.1 represents a common structure for the measurement chain. The input to the chain is the physical quantity to be measured x_o and the output the digital data x_i.

The functional chain of the figure represents the most common model describing a modern electronic sensor. However, it should be noted that some of the elements composing the chain might be missing in a specific design depending on the cost, the required sophistication level, and where the analog to digital conversion takes place. We will return to these aspects later.

2.1.1.1 The Transducer

A transducer is a device transforming one form of energy into another, here converting a physical quantity x_o into an electric or electric-related quantity x_e (in some cases

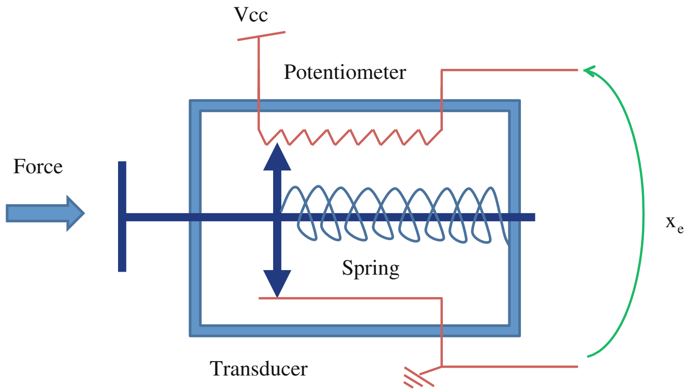

Fig. 2.2 A force transducer composed of a spring and a potentiometer. The force, of intensity x_o, moves a mobile element compressing/releasing the spring. The induced displacement is converted by the potentiometer into voltage x_e thanks to a voltage divider. Vcc represents the reference voltage

the transducer operates with electrical quantities both at the input and output levels). For instance, the temperature of an environment is converted into a voltage (voltage output sensor), the pressure or humidity to a current (current output sensor); the particular target electrical entity depends on the type of the chosen sensor and the way it has been designed. For a detailed analysis of the different typologies of sensors the interested reader can refer to [108]. Clearly, the transduction stage introduces uncertainty on the transduced quantity, which depends on the mechanism used to transform a form of energy into an electrical one.

As an example, and by referring to Fig. 2.2, a sensor of force can be composed, in its transduction principle, of a spring and a potentiometer: the spring converts the force into a displacement and a potentiometer converts the displacement into a voltage variation.

Sensors can be active or passive in their transduction mechanism: an active sensor requires energy to carry out the operation and needs to be powered, whereas a passive sensor does not. Another relevant information is related to the time requested to produce a stable measurement. Such a time depends, for instance, on the dynamics of the transduction mechanisms or the time needed to complete the self-calibration/compensation phase introduced to improve the quality of the sensor outcome.

2.1.1.2 The Conditioning Circuit

The aim of the conditioning circuit [110] is to provide an enhanced electrical quantity x_c of x_e so that the sensitivity of the sensor is amplified, the effect of the noise is mitigated, the interval of definition of the electrical entity is adapted to the requirements of the subsequent ADC. More in detail, the conditioning circuit, which is an analog circuit juxtaposed to the transducer module, at first usually amplifies x_e and then filters its output (e.g., with a low pass filter) to improve the signal-to-noise ratio and the quality of the signal x_c to be passed to the analog to digital conversion stage.

The conditioning circuit might also encompass a module designed to help in compensating parasitic thermal effects, which influence the readout value, as well as introducing corrections to linearize the relationship between the input x_o and x_c. When non-ideal behaviors are compensated by means of a microcontroller, we say that the sensor is enhanced (enhanced sensor). However, it should be pointed out that, in the case of enhanced sensors, the output of the microcontroller is again an analog signal.

In some cases the sensor has an analog output. When this is the case, the output x_i is either x_e or x_c depending on whether the conditioning circuitry is available or not. Analog to digital conversion is carried out later, generally at the microprocessor level, by exploiting the on-chip ADCs. This is a common case in many microprocessors for embedded systems which make available input pins to host analog input signals. Internal on-chip conversion modules are then provided. Clearly, the input signal must be suitably treated and conditioned before it is fed to the microcontroller. For a general presentation of aspects related to embedded system design the reader can refer to [5].

2.1.1.3 The Analog to Digital Converter

The third stage of the functional chain is the conversion module, also known as ADC. The input to the module is the analog electrical signal x_c and the output is a codeword x_b represented in a binary format. There is a large variety of architectures for ADCs [107], all of them having in common the resolution (the number of bits of the codeword) and the sampling rate as target outputs. During the conversion phase, the input x_c must be kept constant, operation carried out by the "sample and hold" mechanism (the analog value is sampled and kept to avoid dangerous fluctuations in the input signal). The conversion introduces an error associated with the quantization level, whose statistical properties may depend on the specific ADC architecture. The source of uncertainty is here variegate and depends, to name a few examples, on the quality of the reference signal (which can change with fluctuations of the powering source), the speed and quality of the conversion step, and the presence of thermal variations that shift the working point of the electronics from a reference ideal one into a different state. The interested reader can refer to [107, 111].

2.1.1.4 The Data Estimation Module

The final module (when present) introduces further corrections on x_b by operating at the digital level. In particular, it generally carries out a further calibration phase aiming at improving the quality of the final data x_i. When a microprocessor is present to address the data estimation module needs, the sensor is defined to be a "smart sensor." The microprocessor can carry out a more sophisticated processing relying on simple but effective algorithms, generally aimed at introducing corrections and structural error compensations. For instance, a thermal sensor can be onboard, in addition to the principal sensor, to compensate the thermal effect on the principal sensor readout. The microprocessor carries out the thermal compensation by reading the temperature value, comparing it with the rated working temperature defined at design time and introducing a correction on the readout value, mostly by considering a polynomial correction function of the discrepancy between the nominal temperature and the current one. The final value x_i shows better properties being closer to x_o. When the dynamics of the signal are known not to change too quickly (compared with the time requested by the ADC to convert a value) or the signal is constant, the microcontroller can instruct the sensor to take a burst of n readings over time. The outcome data sequence $x_{b,j}$ $j = 1, \ldots, n$ can be used to provide an improved final estimate of x_o by averaging

$$x_i = \frac{1}{n} \sum_{j=1}^{n} x_{b,j} \tag{2.1}$$

When the data estimation module is not available, the best estimate of x_o at this level is the value provided by the ADC, i.e., $x_i = x_b$. The designer of the embedded application might decide to carry out this operation later within the application by implementing it in software.

2.1.2 Modeling the Measurement Process

Following the functional description of the sensor given in Sect. 2.1.1 the whole measurement process can now be seen as a black box, suitably described by an input–output model whose simplest, but generally effective form, is

$$x = x_o + \eta \tag{2.2}$$

where $x \in X \subset \mathbb{R}$ is a generic acquired instance, x_o its the ideal, noise-free unknown value, and $\eta = f_\eta(0, \sigma_\eta^2)$ is an independent and identically distributed (i.i.d) random variable with zero mean and finite σ_η^2 variance drawn from probability density function f_η and corrupting the measurement. The additive signal plus noise model (2.2) represents a simple but realistic model describing the measurement process as carried out by the sensor with η accounting for the uncertainty associated with the measurement process. The model implicitly assumes that the noise does not depend on the working point x_o.

Despite the fact that the i.i.d hypothesis is commonly assumed and in fact holds in many circumstances, it might not be satisfied a priori for a specific sensor/application. In fact, we have seen that several sources of uncertainty affect the sensor components and the independency assumption might be violated. It is one of the tasks of the application designer to verify the appropriate model for a sensor as well as determine the existing metrological properties. This is done by first inspecting the sensor datasheet, the operating conditions afterwards, and carrying out suitable acquisitions and metrological analyses whenever requested.

Another common model for the sensor is the multiplicative one where

$$x = x_o + \eta x_o = x_o(1 + \eta). \tag{2.3}$$

In this way, the noise depends on the working point x_o. In absolute terms, the impact of the noise on the signal is $x_o\eta$, but the relative contribution is η and does not depend on x_o. The type of model to be considered depends on the structure of the instrument/sensor available and the way it has been designed and implemented. Working conditions might also have an impact on the selection of the proper model.

In the sequel, we focus on the additive model and introduce other models whenever appropriate. Details related to the validity of the above "signal plus noise" model will be discussed later in the chapter. Despite the particularities of each sensor, we expect some basic properties to hold. The main ones are formalized in the sequel for historical reasons and for their intuitive and common use. However, whenever possible, we should speak about sensor measurement uncertainty. In particular, we need to provide the model adopted for the noise affecting the signal and the pdf function fully associated with the uncertainty. The interested reader can deepen the study of these issues by referring to [180].

2.1.3 Accuracy

Consider the additive signal plus noise model of (2.2). We say that a measure is accurate when the expectation taken w.r.t. the noise satisfies

$$E[x] = x_o. \tag{2.4}$$

In order to have an accurate measurement, the instrument and the measurement process need not introduce any bias contribution. However, this is not always the case: in real-life we all experience problems with sensors providing wrong measurements despite several acquisition attempts, e.g., a room temperature or a badly deployed scale. When this is the case, the simplest model for the sensor becomes

$$x = x_o + k + \eta \tag{2.5}$$

being k the bias value associated with the measure. By taking expectation of (2.5) we have that

$$E[x] = x_o + k \tag{2.6}$$

and, even if we are able to remove the measurement uncertainty, the acquired value is wrong, introducing an unknown offset (bias) value k. When a measurement process is biased we need to subtract the expected value (or its estimate) from the read value. However, since k is unknown, we must rely on a reference value to estimate it. For instance, if we are able to drive the sensor to a controlled state where the expected value is known, say x_o, then, from (2.6) $k = E[x] - x_o$. This phase is called sensor calibration [109, 182].

Accuracy is a main property a measurement system should have since we would like our measurements not to contain any bias error. If we have an accurate measurement system, (2.6) states that, by taking expectation w.r.t. the noise, we remove the impact of noise on the specific value x. During this phase, the value x need not change: in practice, we have to sample at a frequency rate much higher than the dynamics of the signal the sensor is acquiring. This operation is done by the data estimation module if the sensor is smart; otherwise, we have to do it in software with an ad hoc code at the primary microcontroller of the embedded system.

It is always a good practice to take the average of a sequence of n repeated measurements $x_1, x_2 \cdots, x_n$ of the same quantity x to provide a better estimate, $\hat{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, of x_o compared to that obtainable by using a single instance x_i, leading to $\hat{x} = x_i$. The number of samples n we should consider as well as the convergence properties of the average to the expectation are studied in Sect. 4.2.

Example: Sensor Calibration

We bought a low-cost temperature sensor and are not sure about its accuracy. We wish to quantify the potential bias value so as to zero center subsequent measurements.

For this purpose, we drive our sensor to operate at a known reference value x_o (e.g., set by a laboratory-grade temperature standard) and wait until the dynamics effect associated with the change of state vanishes. In the steady state the sensor shares the same temperature as the environment. We then take n samples, say $n = 40$, from the sensor. An estimate $\hat{k}$ for the bias k is

$$\hat{k} = \frac{1}{n}\sum_{i=1}^{n} x_i - x_o. \tag{2.7}$$

If we iterate the process for different x_o values so as to explore the input domain, we can construct a curve that passes through these points and get a very good calibration curve specialized for the given sensor.

Despite the intuitive example, we comment that calibration is a more complex problem if we look at it closely, in its inner mechanisms. For instance, for an integrated temperature sensor, the read value depends on the value of the power voltage, which is contrasted toward a reference value to identify the change in temperature. Any structural discrepancy between these two values introduces a bias error on the final output. Moreover, the measured voltage is not the voltage powering the sensor since the conditioning and the ADC electronics modify it. In addition, the relationship between the measured voltage and the sensed temperature is nonlinear, depending on the transducing mechanism. Compensations of the above phenomena are known in the literature as offset, gain, and linearization.

2.1.4 Precision

Under the signal plus noise framework and the above assumptions, each taken measurement is seen as a realization of a random variable. Measurements will then be spread around a given value (x_o in the case of accurate sensors, $x_o + k$ in case of an inaccurate one), with the standard deviation defining a scattering level index (other indexes can be defined, e.g., as proposed in [181]). In the sequel, precision is a measure of such scattering and is a function of the standard deviation of the noise σ_η, in the case of both accurate and inaccurate sensors.

Given a confidence level δ, precision defines an interval I for x_o within which all values are indistinguishable due to the presence of uncertainty η. In other words, all values $x \in I$ are equivalent estimates of x_o. The amplitude of the interval depends on the confidence level δ, i.e., $I = I(\delta)$, as it will be immediately clear.

To ease the understanding, let us consider at first η as drawn from a Gaussian distribution $f_\eta(0, \sigma_\eta^2)$ of zero mean and variance σ_η^2. The Gaussian hypothesis holds in many off-the-shelf integrated sensors and can be safely introduced unless differently specified by the sensor data-sheet. Under the Gaussian assumption [181] and by setting a confidence level $\delta = 0.95$, we have that a realization x_i of x_o lies in $I = [x_o - 2\sigma_\eta, x_o + 2\sigma_\eta]$ at least with probability 0.95. With the choice of the confidence interval $I = [x_o - 3\sigma_\eta, x_o + 3\sigma_\eta]$ the confidence level raises to 0.997 (acquired x_i belongs to I with at least probability 0.997). The interval defines the precision (interval) of the measure at a given confidence δ. In this last case, the precision of the sensor (sensor tolerance) is defined as $3\sigma_\eta$, so that $x = x_o \pm 3\sigma_\eta$.

When f_η is unknown, we cannot use the strong results valid for the Gaussian distribution. In this case, we need to define an interval I function of δ within a pdf-free framework. The issue can be solved by invoking the Tchebychev theorem [2] which, given a positive λ value and a confidence δ, grants that inequality

$$\Pr\left(|x_o - x| \leq \lambda\sigma_\eta\right) \geq 1 - \frac{1}{\lambda^2} = \delta$$

holds. By selecting a wished confidence δ, e.g., $\delta = 0.95$, we select the consequent value $\bar{\lambda}$. The precision interval I is now $x = x_o \pm \bar{\lambda}\sigma_\eta$. Clearly, the lack of priors about the distribution is a cost we pay in terms of a larger tolerance interval. This can be clearly seen in Table 2.1 where we compare results provided by a "compact" distribution such as the Gaussian one with those obtainable with a distribution-free approach based on Tchebychev's inequality. By having a priori information about the noise distribution, the precision interval can be easily characterized with a better precision.

Table 2.1 The confidence achievable with precision interval $I = [x_o - \lambda\sigma_\eta, x_o + \lambda\sigma_\eta]$ in the Gaussian and the distribution-free case (Tchebychev inequality)

Distribution	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$
Gaussian	0.682	0.954	0.997	1
Distribution-free	n.a.	0.750	0.889	0.938

2.1.5 Resolution

Whereas precision is a property associated with a measure, resolution is associated with an instrument/sensor and represents the smallest value that can be perceived and differentiated by others given a confidence level.

If our instrument has a resolution of 1 g, we will not be able to measure values of 1 mg due to the limits of the instrument: the scale will make sense in steps of 1 g (and all values in such interval will be equivalent and indistinguishable). However, having a high resolution neither implies that the measure is accurate nor precise. In fact, the scale can be badly mounted, hence introducing a $k = 100$ g fixed error in the readout (the scale is not accurate). Moreover, if the scale is analog, we might not be able to perceive changes affecting the gram for visualization insufficiency but only something around 10 g (precision error): the size of the pointer might well exceed the gram!

Since our final interest is the accuracy and precision of a sensor, sensor designers mostly provide the precision level (by automatically also considering the resolution impact on the measure in there). That said, the reader must be aware of the confusion present in the market and attention should be paid before selecting a sensor. Moreover, a metrological analysis phase should be carried out if we are not sure about the provided figures.

Example: A Real Sensor

Table 2.2 presents the main features of a temperature sensor for aquatic measurements. The resolution of the instrument is high, but the impact of the noise on the readout value is high as well. The sensor provides values within a [−4 °C, 36 °C] interval with an additive error model influencing the read value up to ±0.3. We immediately derive that $\sigma_\eta = 0.1$ since the sensor is ruled by a Gaussian distribution from data-sheet information and we consider $\lambda = 3$. Otherwise, we should have invoked the Tchebychev's inequality, set a confidence level, e.g., 0.997 (so as to be in line with the Gaussian case) leading to $\lambda = 5.77$. The sensor requires a warm-up time up to 2 s: any value read before the warm-up time has elapsed would produce erroneous data (repeatability is not granted). Engineers should pay attention to this issue.

Table 2.2 A temperature sensor for aquatic measurements

Features	Value
Range	−4–36 °C
Resolution	0.01 °C
Accuracy	±0.3 °C
Response time	≤2 s

2.2 A Deterministic Versus a Stochastic Representation of Data

A common problem we face when designing an embedded application is related to the number of significant digits available within the given codeword. The uncertainty aspect introduced by the binary representation will be studied in detail in Sect. 3.1. Differently, here, we focus on the fact that uncertainty exists and affects somehow the data. We ask ourselves the question: if the output of the data estimation module x_i is represented by means of n bits and hence uncertainty affects the readout, how many bits p are relevant out of the n? The answer to the question requires a deeper analysis and can be addressed by considering two relevant scenarios depending on the nature of the available data, as it will become clear in the sequel.

Consider $x_o = x_o(t) \in X \subset \mathbb{R}$ to be a signal evolving over time and assume that the measurement process is much faster than the dynamics of the signal so that sample $x = x(t)$ can be considered constant during each data acquisition.

2.2.1 A Deterministic Representation: Noise-Free Data

The case covers the situation in which digital data x_i are confined within a deterministic domain, i.e., the feasible values of acquired data are error-free and belong to the closed interval $[a, b]$. If n bits are made available to represent the data and no noise affects them, then each of the 2^n available codewords are worth to be used. By considering a reasonable uniform assignment codeword-information, the distance Δx between two subsequent data instances is

$$\Delta x = \frac{b-a}{2^n - 1}$$

if we also wish to represent both extremes of the interval. In this way the 2^n codewords are respectively assigned as $x_1 = a, x_2 = a + \Delta x, \ldots, x_{2^n} = b$. Clearly, different assignments can be made, also depending on the specific application. Given a value x_o, the maximum representation error is $\frac{\Delta x}{2}$ and the average error is zero. If values are uniformly distributed in the interval $[x_o - \frac{\Delta x}{2}, x_o + \frac{\Delta x}{2}]$, then the variance of the error representation is $\frac{\Delta x^2}{12}$.

Differently, if the data we wish to represent are affected by noise, as it is generally the case, then not all the codewords are meaningful and less than n bits are relevant and should be kept.

2.2.2 A Stochastic Representation: Noise-Affected Data

As we have seen in the measurement chain, data acquired from a sensor are noise-affected. Obviously, we are not interested in spending bits to represent the noise when writing a number. At the same time, precision introduces a constraint on the indistinguishable values we can acquire. In fact, two data are distinguishable and deserve distinct codewords only if their distance is above the precision interval I which, as we have seen, depends on a predefined confidence level δ and acts as the deterministic Δx of the Sect. 2.2.1. The number of independent values can be written as the ratio between the domain interval of the data and the value $I_m = 2\lambda\sigma_\eta$ of the probabilistic indistinguishability interval, σ_η being the uncertainty standard deviation associated with the measurement process. Finally, the number of independent points I_p is

$$I_p = \frac{b-a}{2\lambda\sigma_\eta} + 1$$

if we require the interval extremes to be represented. As before, a straight assignment would be $x_1 = a, x_2 = a + I_m, \ldots, x_{I_p} = b$. The number of significant bits is now

$$p = \lceil log_2\left(I_p\right)\rceil$$

where $\lceil\cdot\rceil$ is the ceiling operator. We have that $p \leq n$ represents the significant bits within the n with the statement holding at least with probability δ. Figure 2.3 shows how values around $x_o = 6$ are affected by noise under the assumption that the noise is normal (zero mean, unitary standard deviation, and $\lambda = 3$). As we get further from x_o, the probability of having a value wrongly assigned to x_o diminishes. Codewords are $x_o = 0, 6, 12, 18$ but the error distribution is shown only in correspondence to codewords $x_o = 6$ and $x_o = 12$. Given the tails of the distribution, we might erroneously assign with probability $1 - \delta$ a wrong codeword to a given value. Such a probability is 0.003 for the example given in Fig. 2.3, e.g., see numbers around 9 which can be assigned both to $x_o = 6$ and $x_o = 12$, though with different probabilities. However, in most reasonable distributions (and in most embedded applications), the introduced error is contained since the mis-assignment probability rapidly diminishes.

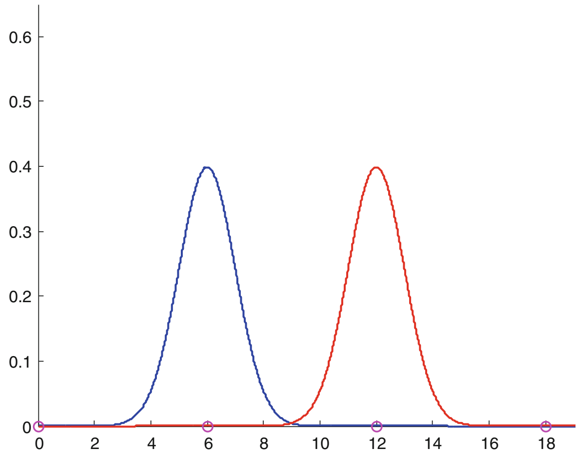

Fig. 2.3 The impact of a normally distributed noise as a function of the distance from $x_o = 6$. The presence of a distribution tail implies that we might wrongly assign the codeword of x_o to a value x which should be assigned to a different codeword instead. As an example, value $x = 9.1$ associated with codeword $x_o = 12$ might have been generated by $x_o = 6$ as well

2.2.3 The Signal-to-Noise Ratio

Consider now the case where the signal is not bounded in deterministic terms and measurements are modeled as instances drawn from a stationary—possibly unknown—pdf. A probabilistic interval can be identified for x_o whose probabilistic extremes are associated with $\lambda_x \sigma_x$, being σ_x the standard deviation of the signal and λ_x the term modulating the width of the interval, chosen to grab confidence level δ. As in previous sections, the number of independent values I_p depends on the interval between two distinct codewords, which are distinguishable according to confidence level δ

$$I_p = \frac{2\lambda_x \sigma_x}{2\lambda \sigma_\eta} + 1$$

By considering the same λs both for the noise and signal we define the Signal-to-Noise ratio (SNR) as

$$SNR = \log \frac{\sigma_x}{\sigma_\eta}$$

where the logarithm base can either be 2 or 10 depending on the subsequent use. Interestingly, $2SNR$ represents the logarithmic ratio of the energy of the signal compared with that of the noise. The SNR is pdf-free and applies to any distribution thanks to the Tchebychev inequality, provided that the same λ value is considered. The number of relevant bits p of the binary codeword finally becomes

$$p = \left\lceil \log_2 \left(\frac{\sigma_x}{\sigma_\eta} + 1 \right) \right\rceil \leq \lceil SNR_2 \rceil + 1. \tag{2.8}$$

If $p \geq n$, all bits present in x_i are statistically relevant; otherwise, only p out of n are relevant and $n - p$ are associated with noise. We comment that (2.8) holds for $\frac{\sigma_x}{\sigma_\eta} > 1$, i.e., in all meaningful applications. Alippi and Briozzo [37] show how the SNR can be used to dimension a digital architecture implementing the scalar product between two vectors and then the processing requested by an artificial neuron.

http://www.springer.com/978-3-319-05277-9

Intelligence for Embedded Systems
A Methodological Approach
Alippi, C.
2014, XIX, 283 p. 81 illus., 73 illus. in color., Hardcover
ISBN: 978-3-319-05277-9