Università degli Studi di Padova

Scuola di Ingegneria
Dipartimento di Tecnica e Gestione dei sistemi industriali

Corso di Laurea Magistrale in Ingegneria Meccatronica

Tesi di laurea

Measurement and analysis of harmonic emissions of digital inverter refrigerators

Relatore: Prof. Alessandro Sona
Correlatore: Ing. Tristano Dal Canton

Laureando: Valerio Sgnaolin

February 13, 2019

Contents

Abstract

This master's thesis has the purpose of illustrating the work done on a 1-year-long experience at Electrolux Italia S.p.A. The aim was to find and implement a mathematical tool to extract information on the emissions of inverter refrigerators in order to automatically recognize soft anomalies and prevent hard faults. Such information had to be derived by sets of measures of current, sensed by a single on-board chip.

Therefore, starting from the most famous method of spectral analysis, *i.e.* the Fourier's transform, an alternative was found in the Wavelet Transform. Exploiting the Multi-Resolution in time and frequency provided by this kind of analysis, a new modality of looking at signals was achieved, allowing a much more concise view of their spectrum along time. An energy map was then created from such transform and was used as a set of features to be recognized by an algorithm of classification in the future.

The evolution of knowledge eventually led to a minimal experimental setup that was composed by the real-application motherboard with an on-board current sensor, through which the current supplying the inverter flew. Measures of current were acquired with the internal Analog-to-Digital Converter of the microcontroller unit of the board, and the following analysis and classification made online.

Together with a case study as a successful example of this method, also the limitations of the state of things are presented.

Chapter 1

Introduction

This thesis has been elaborated during a 1-year-long period of internship (6th March 2017 – 5 March 2018) at the Research & Development (R&D) department of Electrolux Italia S.p.A., a multinational manufacturer of home appliances. In particular, the workplace has been the plant of Susegana (TV), Italy, where the assembly lines for built-in refrigerators are located, producing for the European market. Consequently, the work of the electronic team - to which the author was assigned - is focused on this particular family of products.

1.1 The context

In the context of refrigeration industry, like in any other field of mass production, the minimization of the costs is one of the major factors that lead to the economic success of a product. Although the electronics embedded in a fridge has a small impact on the final price of the appliance for the customer, the designers must utilize the most suited state-of-the-art techniques for the projects, both to increase the profit margins and to reduce the number of burdensome repair calls. This latter issue, free of charge for the end-user during the warranty period, is exclusively upon the manufacturer and, the more a non-optimized strategy of intervention is employed, the higher is the amount of wasteful costs. Since the Customer Service Technicians (CSTs) often do not have the time or the skills to spot what the real problem of a Printed Circuit Board (PCB) is, they generally end up with substituting the whole board, which is the most expensive choice they could take in terms of material resources. That is why a more focused intervention of the Customer Service (CS) can become a precious source of savings.

1.2 The problem

The focus of this thesis is giving the basis to an automated firmware of fault recognition that, taking advantage of the possibilities of the Wi-Fi connectivity of the new generation household appliances, allows the maintainer to know the cause of a potential malfunction in advance and, therefore, to provide the best solution in terms of resources, times and costs.

Most of the benefits from such a diagnostic system would hopefully apply to the CS. The CSTs often opt for the replacement of the whole PCB when a problem takes place, since that is the most time-saving solution. By the way, it is the most costly one, too. Giving the technicians an instrument to spot the specific problem, which might have a very cheap solution, is the aim of this work.

Specifically, the application of interest is the creation of a ***fault diagnosis technique*** to implement on the MKE04Z128 Microcontroller Unit (MCU) (ARM family, Cortex-M0) using the measurements from the Allegro's ACS718 Current Sensor, which is located between the grid and all the loads on the glsPCB.

1.3 The exit strategy

To overcome the inefficiencies cited in 1.1 and 1.2, an automatic recognition pattern algorithm has been elaborated. Such a sequence of actions can be summarized in the following steps: after acquiring a measurement of the current absorbed by the system, a series of mathematical operations is performed on this sampled signal in order to identify a set of values (the smallest possible) which characterizes it. A classification rule is then applied on this values to assign a known class to this signal within a *pletora* of previously identified possibilities. Such a kind of method is usually referred to as a *supervised learning algorithm.*

In this way, a current signal from the Allegro's ACS718 (*cf.* Sec.1.2) during a specific interval of time can thus be classified as an ordinary behavior of the system or, on the contrary, as some type of predetermined fault and, consequently, notified to the CS. The CSTs would then be prepared for the specific situation before getting to the customer's place, ready to perform a targeted intervention.

The ***purpose of this work*** is showing off the results achieved with this strategy, the limitations encountered and the systematic approach for the identification and the measurement of each fault condition.

The first step (Chap.2) is identifying a way to treat the raw acquired signal in order to see its content from the most favorable point of view. Secondly, the aim will be to extract the most useful information, for the pattern recognition, so that some tools for the analysis are presented together with their possible applications in a classification algorithm (Chap.3). The results of the application of the theoretical considerations of Part I will be presented in Part II, where some of the experimental results are illustrated. In Part III complete lists of references are offered.

The bet of this work is achieving a complex problem like fault recognition by means of a single current sensor.

Part I

Theoretical basis

Chapter 2

The Wavelet Transform Theory

Generally, time-domain signals are difficult to interpret: to overcome this difficulty, mathematical transforms were invented. The aim of transforms is to extract some information that is not readily available in the raw signal [25].

In engineering applications, a "concealed" aspect of major importance is the *frequency spectrum* of a signal. The most popular mathematical tool to extract that kind of information from the time-amplitude representation of the signal is the Fourier Transform (FT). As this chapter will demonstrate, the FT is neither the unique nor the most appropriate tool available to analyze the hidden aspects of a signal. The purpose of this chapter, though, is not to furnish an exhaustive explanation of the underneath mathematics (for that, see [3] and [15], for example), but rather to give an intuitive approach.

2.1 Overtaking Fourier

In this section, starting from a description of the Fourier Transform, with its benefits and shortcomings, the Wavelet Transform will be defined and an explanation of when and why it is a better instrument than the FT will be provided.

2.1.1 Fourier Transform

The Fourier Transform, published by J. Fourier in Paris in the Twenties of the nineteenth century, has become the most utilized mathematical process to extract the frequency spectrum of signals. It translates a signal from its time-amplitude representation to a frequency-amplitude one. The FT achieves this result by computing the coefficients to assign to a new basis of complex exponential orthogonal functions (see Eq.2.1), *i.e.* giving the magnitude of each spectral component of the signal, in order to change the "point of view". With the convention of naming $\hat{g}$ the FT of an integrable function in the time-domain

$g : \mathbb{R} \rightarrow \mathbb{C}$, the formula that defines it is:

$$\hat{g}(f) = \int_{-\infty}^{+\infty} g(t)e^{-j2\pi ft}dt \tag{2.1}$$

for every frequency $f \in \mathbb{R}$.

Although being extremely useful for characterizing the majority of components and equipment, this kind of transform has a substantial deficiency when dealing with a generic signal: its basis functions are periodic and infinite. This implies that also the signal being processed should have such characteristics to be perfectly represented. Assuming that a real-world signal has these peculiarities, while it does not, inevitably leads to some error. For a generic unknown compact-support signal, for instance, such assumptions cannot be made at all.

Citing Strang's musical metaphor in [28], performing the FT of a symphony (a non-stationary acoustic signal) is like assuming an infinite orchestra in which every instrument plays a single steady note. *"Conductor not needed, musicians totally bored".*

In this work, indeed, no *a priori* assumption about the periodicity and duration of the potential fault signals in exam could be made: therefore, a better instrument to describe generic signals should be sought. In short, there is the need of overcoming the ***global*** point of view of the FT and explore new methods of looking ***locally*** at signals to find out their spectral changes along their duration.

2.1.2 The Short-Time Fourier Transform

At first sight, actually, the FT could still be considered a valid instrument for the description of the local spectrum: it can be applied on short intervals of time T and not on the interval $(-\infty, +\infty)$ anymore (*Short-Time Fourier Transform (STFT)*). In other words, the signal is windowed with a function $w(t)$ that gets shifted in time by τ. (Eq.2.2) shows this off:

$$\hat{g}_w(t, f) = \int_{t}^{t+T} \Big(g(t) \cdot \overline{w(t-\tau)}\Big)e^{-j2\pi ft}dt \tag{2.2}$$

where the parameters t and f provide time and frequency, respectively. This formulation is called STFT.

In each segment, the signal gets split into harmonics again, but with different magnitudes for every interval of time covered by the window. In such a way, always with respect to the metaphor in [28], *"The musicians still play one note each, but they change amplitude in each segment".* Nonetheless, there are several disadvantages: for instance, the discontinuities between the segments. Sudden breaks are not avoidable with the STFT and this is one of its troublesome limitations. Moreover, as pointed out in [30], this windowed version of the

FT leads to aliasing phenomena affecting accuracy and efficiency, especially in discrete time: since the FT is performed on a sliding segment of length T on a time series of time step dt (for a total length of Mdt), the range of returned frequencies spans in $f \in [1/T, 1/2dt]$ at each time step. Consequently, aliasing arises for components out of that interval. If one or more dominant components are out of that range, the result gets inconsistent. This dependency on the "response interval" ought to be avoided.

2.1.3 The Wavelet Transform

As demonstrated in this section, a more appropriate tool for analyzing an unknown and possibly time-variant (*i.e. non-stationary*) signal is the family of the Wavelet Transforms. Unlike Fourier decomposition, this transform does not rely on bases of periodic functions: instead, a wavelet (that is a compact-support function, a "little wave") gets scaled in frequency and translated in time, giving a basis of compact-support functions. The possibility of having a local insight of the spectrum, is just one of the major benefits of the Wavelet Transform (WT). Another advantage is the availability of many trade-offs in the resolution in time and frequency in order to adjust the focus on the aspects of interest. Historically, the formalization of such capability of resolution on multiple dimensions was introduced by S. Mallat and Y. Meyer in 1989 with the name of Multi-Resolution Analysis (MRA).

Referring again to Strang's metaphor in [28], the WT reflects a logic similar to the composer's writing down the symphony. In fact, the sub-band subdivision of the WT could be symbolized by an orchestra in which the basses play a passage with the lowest tune, then, for instance, twice as many cellos (which have a higher tune) play the same passage at a doubled frequency, but each starts shifted in time to cover the same time interval as basses (thus with a halved time separation than in-between the basses). Then twice as many violas will in turn speed up by a further factor of two, starting at new time locations, separated by a half of the time step of the cellos. And so on for violins, *etc.*. All of them overlap to compose a fragment of the symphony.

Each instrument is then characterized by two parameters: its tune (related to the fact that it is a bass, cello, viola, violin, *etc.*), interlaced with the speed, *i.e.* the frequency, and the time it starts playing the passage during the whole time interval. Paraphrasing this metaphor, each instrument is like a basis function and the amplitude of the sound of each of them playing the passage is like the series of coefficients that the computation of the WT assigns to each basis function.

To sum up, each musician playing an instrument (*frequency*, in terms of tone or speed) is told the intensity (*coefficients*) and the instant (*time*) to start playing the passage (*wavelet*). All this local information superposed returns the symphony (*signal*).

Briefly, this is the underlying concept of the WT: a "mother" wavelet is scaled in frequency and translated in time so that the sum of all these dilated and shifted version (each given certain amplitude coefficients) is the original signal.

2.1.4 Comparing Fourier and Wavelet Transform

For both the FT and the WT, the mathematical process aims to ***decompose*** the signal into a sum of basis functions multiplying some amplitude coefficients. A further similarity consists in the orthogonality amongst the functions of the basis. Although in the FT such a fact is strictly connected to the complex exponential functions constituting the basis, in the WT this characteristic is not really needed to form a valid base. Nevertheless, this property is usually granted in the construction of the most used wavelets (like, for example, the Daubechies family).

On the contrary, the fundamental difference between these two kinds of transforms resides in the support[1] of the basis functions. Whereas the FT takes into account infinite complex sinusoidal functions, thus providing a global vision of the signal, all of the WTs are based on a set of finite-support functions, therefore returning a local point of view. Looking at the example in Fig.2.1 from [29], the drawback of the global view of the FT becomes glaring: if a signal like a human voice got decomposed using FT, no time localization would be possible. Which frequencies are present would be known, but no clue about the time they appear and their duration would be obtained. On the contrary, the wavelet decomposition, providing local information both on frequency and time, allows to know the extent and the location in time of each note (thus letting us write it correctly on the pentagram).

This difference gets critical when considering finite non-stationary signals: as a matter of fact, the FT can do nothing but committing errors trying to force infinite functions to approximate finite signals, whereas the WT adapts much better to the shape of the original wave, modulating the amplitude of every scaled and shifted version of the mother wavelet constituting the basis.

Another particularly intuitive way of realizing the differences between these two types of decomposition is looking at the so-called “Heisenberg boxes” in the Phase Plane, Fig.2.2. The purpose of these boxes is to illustrate ***uncertainty principle*** that rules the MRA, according to which the frequency and time of a signal at a specific instant cannot be exactly known. Instead, there is a trade-off between the resolution in time and in frequency, resulting in an area of ambiguity on a time-frequency graph. On the so-called Phase Plane, indeed, time is located on the horizontal axis, whereas frequency is posed on the vertical axis. As Fig.2.2 shows, the extreme cases in which we completely lose the resolution either in time or frequency domain are: (i) a pure standard-basis time acquisition — like in Fig.2.2a — where we have no clue of the frequency content; (ii) the FT of the

[1] The support of a real-valued function $f(x)$ is a subset of the domain $\mathcal{D}$ containing the elements that are not mapped to zero, *i.e.* $\operatorname{supp}(f(x)) = \{x \in \mathcal{D} | f(x) \neq 0\}$.

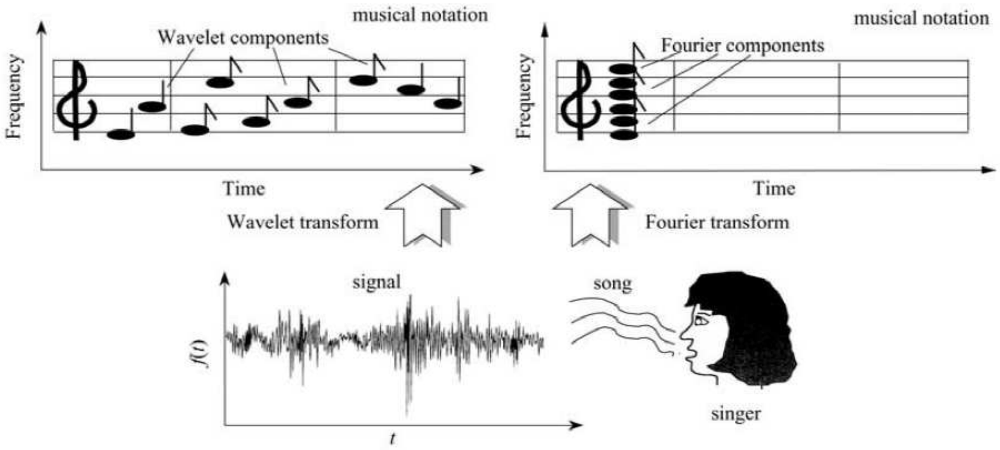


Figure 2.1: Fundamental difference between the signal decomposition by FT and WT: the transcription of a song (signal $f(t)$) can be achieved by wavelet decomposition thanks to its time-frequency localization, but would fail with Fourier decomposition [29].

signal, which loses every information about the time, as in Fig.2.2b. In all the other cases, there is a trade-off between time and frequency resolution.

In the STFT, the parameter that lets us vary the resolution is the length of the window: as it is evident in Fig.2.2c and Fig.2.2d, a narrow window privileges time resolution, whereas a wide one allows a finer resolution in frequency, losing some accuracy on the time axis. In both these cases, a homogeneous subdivision of the Phase Plane is obtained, *i.e.* all the frequency intervals have the same sub-band width and all time intervals have the same duration. Nevertheless, as already stated in Sec.2.1.2, aliasing and inaccuracy problems make of the STFT a poorly desirable tool for the purposes of this work.

Differently, the (“classic”) WT in Fig.2.2e provides a sort of “gradient” in time and frequency resolution: while low frequencies are well differentiated (small heights of the boxes), but blurry in terms of time location (wide boxes), higher frequencies are less distinguishable (higher boxes), but are much better located in time (narrow boxes). Since the content of a signal is often mainly in the low-frequency part of such a Phase Plain (which is due to the choice of an adequately high sampling frequency, not to lose information on the useful part of the signal), the flexibility of this last decomposition appears as a major benefit. With the WT the Phase Plane can be divided along such ***octave band subdivision***, providing adjustable compromises in resolution for the analysis of a signal.

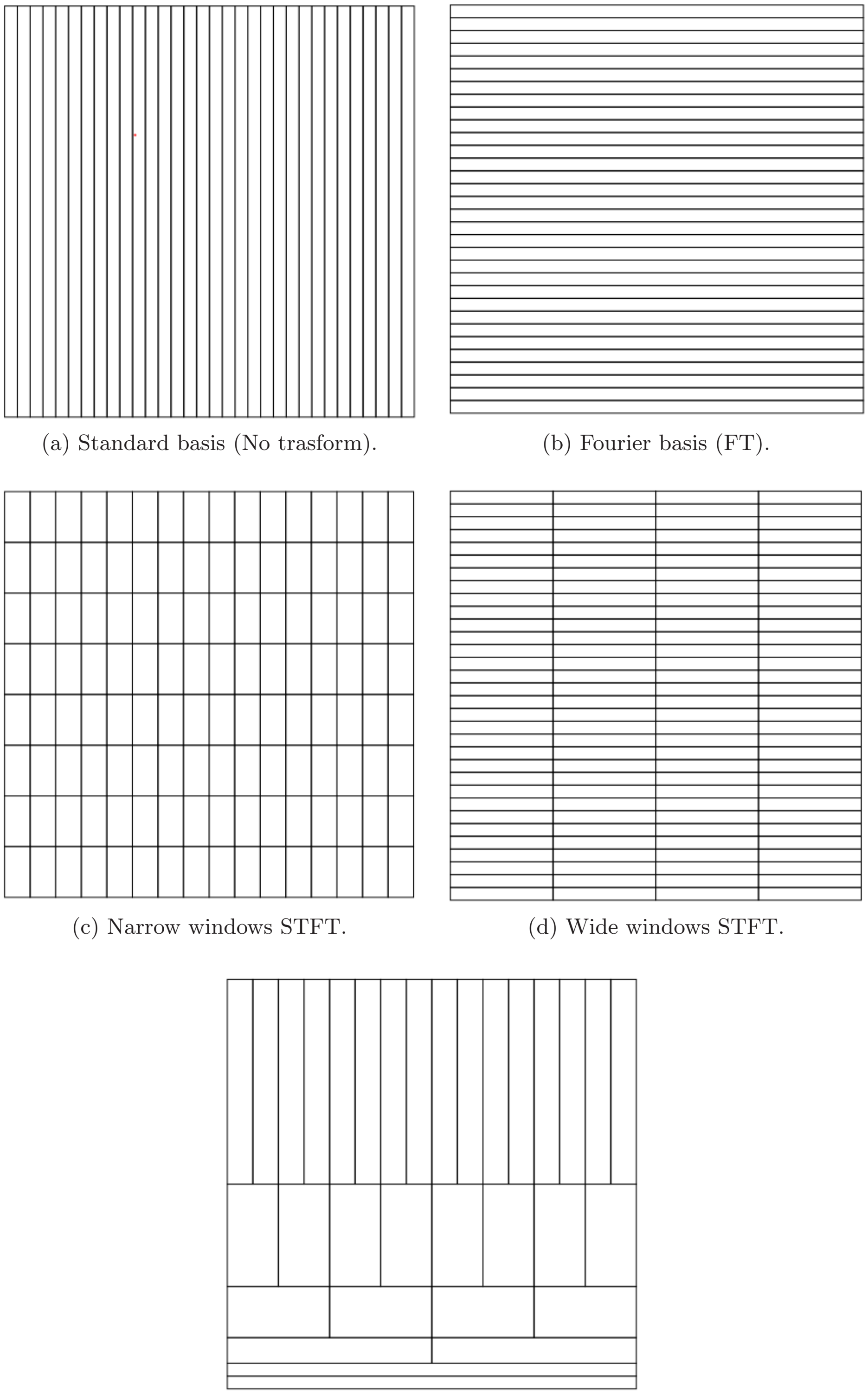
2.2 The Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) was developed as an improvement of the STFT to overcome its resolution problem [25]. In that sense, they are similar, because the signal is multiplied for a function (*cf.* a window in the STFT, a wavelet in the CWT) and the transform is computed separately for different segments of the time-domain signal. The main aspect differentiating them is that the width of the “mother” wavelet is changed as the transform is being computed.

The CWT $\Phi_f^{\psi}(s,\tau)$ of a function $f(t)$, using the mother wavelet ψ (which is a sort of prototype of a “window”), is described by the parameters s of scaling and τ of shifting. It can be expressed by the hermitian inner product[2] in Eq.2.3:

$$\begin{aligned}\Phi_f^{\psi}(s,\tau) &= \langle f(t), \psi_{s,\tau}\rangle \\ &= \int_{-\infty}^{+\infty} f(t)\overline{\psi_{s,\tau}}dt \\ &= \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{s}}\overline{\psi\left(\frac{t-\tau}{s}\right)}dt\end{aligned} \tag{2.3}$$

[2] For clarifications about the inner hermitian product see Sec.2.2.1.



(a) Standard basis (No trasform).

(b) Fourier basis (FT).

(c) Narrow windows STFT.

(d) Wide windows STFT.

(e) Wavelet basis (WT).

Figure 2.2: These Heiseberg boxes let intuitively sense the different uncertainty in time and frequency domains according to several kinds of decomposition [32].

This equation is known as the *analysis* function[3].

The use of the time *translation* term τ is similar to that in the STFT: it describes the position of the wavelet as it is shifted through the signal. This parameter thus corresponds to the time information in the transform domain [25].

However, rather than a frequency parameter as in the STFT, in the CWT there is a *scale* parameter, defined as $s = 1/f$. The *scale* parameter denotes a concept similar to the scale used in maps [25]. As high scales correspond to a global non-detailed view and low ones to a detailed one in maps, in terms of frequency, so low frequencies (*i.e.* high scales) correspond to a a global information and high frequencies (*i.e.* low scales) correspond to a detailed information. In this way, the original one-dimensional signal $f(t)$ is mapped to a two-dimensional $\Phi_f^{\psi}(s,\tau)$ one, thus obtaining multi-resolution observability.

Scaling, mathematically, is an operation that either dilates or compresses a signal. Larger scales correspond to a dilated (or stretched) version of a signal, smaller scales to compressed ones. Indeed, if a generic function $g(t)$ gets scaled by $s > 0$ giving $g(t/s)$, $s > 1$ dilates the signal, whereas $s < 1$ compresses it.

Computing the transform with a small scale — and, thus, a "compressed" wavelet (starting from $s = 1$, for convenience [25]) — allows to highlight high frequencies and locate them in time, since the multiplication between the signal and a narrow window gives relatively high values for high frequencies where they exist. Similarly, large scales — and, thus, stretched wavelets — extract low frequencies, since the multiplication of the signal by a dilated window returns relatively high values for low frequencies.

The largeness (*i.e.* the scale) of the wavelets directly affects the resolution: narrow wavelets provides a finer scale resolution since their narrow support makes less ambiguous to identify the exact value of the scale, which corresponds to a poorer resolution for high frequencies. On the contrary, large wavelets provide a poorer identification of the precise scale, which corresponds to a finer resolution for low frequencies. Anyway, the exact value of a point in the time-frequency plane cannot be known according to *Heisenberg uncertainty principle*: as visible in Fig.2.2e, each box has a non-zero area and all the points falling into a box are represented by one value of the WT. Note that every box has the same area, since there is just a change in the proportions between the heights and the widths of the boxes: at low frequencies, the heights are smaller (that means less ambiguity in frequency), but widths are larger (*i.e.* more ambiguity), while at high frequencies, heights are greater and widths smaller.

To sum up, the CWT coefficients are a measure of the closeness of the signal to the wavelet at the current scale in terms of frequency content.

[3] The multiplication by $\frac{1}{\sqrt{s}}$ after the integration is an energy normalization factor assuring that the signal will have the same energy at every scale [25] [13].

2.2.1 Orthonormality of the CWT

To comprehend some interesting properties of the CWT, there is the need of some reminders. In first place, the notion of inner product must be clarified:

Definition 2.2.1 *The hermitian inner product (extension of the dot product) between two complex vectors* $\mathbf{v}$ *and* $\mathbf{w}$ *is defined as:*

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_i v_i \overline{w_i} \tag{2.4}$$

where $\overline{w_i}$ *is the complex conjugate of the i-th element of vector* w*.*

This notion in Def.2.2.1 can be generalized to complex functions:

Definition 2.2.2 *The inner product between two complex functions* $f(t)$ *and* $g(t)$ *in the interval* $[a, b]$ *is defined as:*

$$\langle f(t), g(t) \rangle = \int_a^b f(t)\overline{g(t)}dt \tag{2.5}$$

where $\overline{g(t)}$ *is the complex conjugate of* $g(t)$*.*

The *orthogonality* is verified when the inner product of two functions (or two vectors) is zero:

$$\langle f(t), g(t) \rangle = 0 \tag{2.6}$$

In second place, we can extend the binary relation of orthogonality to sets of n vectors (*e.g.* a basis) $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$. Besides, when they are pairwise orthogonal and their norm is unit, they are *orthonormal* sets. This is expressed mathematically by the Kronecker delta δ in Eq.2.7:

$$\langle \mathbf{i}, \mathbf{j} \rangle = \delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{2.7}$$

Generalizing to a set of r functions $\{\phi_1(t), \phi_2(t), \ldots, \phi_r(t)\}$, such set is orthonormal when:

$$\langle \phi_k(t), \phi_l(t) \rangle \ = \delta_{k,l} \tag{2.8}$$

where $\delta_{k,l}$ is the Kronecker delta in $k, l = 1, 2, \ldots, r$, and so, equivalently:

$$\begin{aligned} \textstyle\int_a^b \phi_k(t)\overline{\phi_l(t)}dt \ &= 0 \quad \text{if } k \neq l \\ \textstyle\int_a^b |\phi_k(t)|^2 dt \ &= 1 \end{aligned} \tag{2.9}$$

As stated above, the bases for the WT need not be orthonormal, but those bases that have such a property allow computation of the coefficients in a much simpler way than the others. As a matter of fact, for an orthonormal basis, the coefficients μ_k of the CWT are computable as:

$$\mu_k = \langle f(t), \phi_k(t) \rangle = \int f(t)\overline{\phi_k(t)}dt \tag{2.10}$$

2.2.2 Reversibility of CWT

Referring to the notation of Eq.2.3, the reconstruction formula of the CWT is:

$$f(t) = \frac{1}{c_\psi^2} \int_s \int_\tau \Phi_t^\psi(s,\tau) \frac{1}{s^2}\, \psi\left(\frac{t-\tau}{s}\right) d\tau ds \tag{2.11}$$

where c_ψ is called *admissibility constant.* This equation constitutes the *synthesis* function. Actually, the CWT is a reversible transform provided that Eq.2.12 is satisfied.

Theorem 2.2.1 *For the success of the reconstruction in Eq.2.11 the admissibility constant c_ψ must satisfy the condition:*

$$c_\psi = \sqrt{2\pi \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega} < \infty \tag{2.12}$$

where $\hat{\psi}(\omega)$ is the FT of $\psi(t)$. This implies that $\hat{\psi}(0) = 0$, i.e. *the continuous component must be null:*

$$\int \psi(t) dt = 0 \tag{2.13}$$

In other words, the function $\psi(t)$ must be oscillatory.

That is not a very restrictive condition: in fact, there are also non-orthogonal basis functions that satisfy it. Moreover, as far as the solely analysis is concerned, the reversibility of the CWT may seem not needed. It clearly becomes a restriction in the case of the reconstruction of the signal, instead, which is not one of the specific purposes of this work. Nevertheless, since computers need a digitized algorithm, a discretization must be performed: in that case, the original continuous-time signal ought to be reconstructed by its discrete samples to be considered trustworthy. With Eq.2.11, the wavelet grants the reconstruction of the signal from a continuous transform, as a matter of fact.

2.2.3 Discretization of CWT

The CWT cannot be exactly computed automatically since the calculus in a continuous domain is a purely theoretical tool of the analysis. A digitization of the procedure can be executed in order to attain a good approximation of the theory using a computer.

Like it has historically been done for the discretization of the FT and the STFT, the most intuitive way of accomplishing this result is sampling the time-frequency plane with a uniform step. Nevertheless, the scale changes along the decomposition can be used to reduce the sampling rate. At higher scales (low frequencies), the sampling rate (f_s) can be decreased: indeed, if the time-scale plane need to be sampled at a sampling rate $f_{s,1}$ at a scale s_1, the same plane can be sampled at a rate $f_{s,2}$ where $s_1 < s_2$ and $f_{s,2} < f_{s,1}$, according to Nyquist theorem. The actual relation is:

$$f_{s,2} = \frac{s_1}{s_2} f_{s,1} = \frac{f_2}{f_1} f_{s,1} \tag{2.14}$$

This means that at lower frequencies the sampling rate can be decreased, saving a large amount of computational time. With these arrangements, it is possible to define a sort of discretized version of the CWT, called the *Wavelet Series Transform (WST).*

In the first place, then, the scale parameter s is discretized according to a logarithmic grid. Then, the time (or, better, the *translation*) parameter is discretized with respect to the scale parameter, *i.e.* a dyadic sampling scheme is used. The base of the logarithmic grid generally depends on the user, but it usually is $s_0 = 2$. Therefore, if the scale increases by a factor of 2, the sampling rate reduces by the same factor at every step, according to Eq.2.14. In mathematical terms, expressing the scale discretization as $s = s_0^j$ and the translation discretization as $\tau = ks_0^j\tau_0$, where $s_0 > 1$ and $\tau_0 > 0$, the continuous wavelet function:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{2.15}$$

becomes:

$$\psi_{j,k} = s_0^{-j/2}\psi(s_0^{-j}t - k\tau_0) \tag{2.16}$$

where $j, k \in \mathbb{Z}$. Typical values are $s_0 = 2$ and $\tau_0 = 1$.

In case all of the $\psi_{j,k}$ form an orthonormal basis, the WST becomes:

$$\Phi_t^{\psi_{j,k}} = \int f(t)\overline{\psi_{j,k}(t)}dt \tag{2.17}$$

or:

$$f(t) = c_\psi \sum_j \sum_k \Phi_t^{\psi_{j,k}} \psi_{j,k}(t)dt \tag{2.18}$$

Anyway, even though this discretized CWT can be computed automatically, it implies a huge computational burden that can lead to unacceptable time losses not allowing real-time. Therefore a much faster algorithm was optimized to obtain a WT on computers: the Discrete Wavelet Transform.

.

2.3 The Discrete Wavelet Transform

Although the WST enables the computation of an approximation of the CWT by computers, it is not a truly discrete transform, it is just a sampled version. It is redundant in information (as for a possible reconstruction) and expensive in terms of computational time and resources.

The Discrete Wavelet Transform (DWT), instead, provides only the sufficient information both for analysis and synthesis, with a significant reduction in computational time.

As in the CWT, the purpose is to provide a time-scale representation of the signal. Whereas in the CWT the analysis formula was achieved by changing the scale of the window, shifting it, multiplying it by the signal and then integrating, in the DWT the signal is passed through a series of high-pass and low-pass digital filters to analyze separately high and low frequencies. This technique is called ***sub-band coding***.

The resolution of the signal is changed by filtering operations, whereas the scale is modified by down-sampling (or sub-sampling) operations. The DWT coefficients could be seen as samples from the CWT, taken following a dyadic scheme with $s_0 = 2$ and $\tau_0 = 1$, and so $s = 2^j$ and $\tau = k2^j$, according to the notation of Sec.2.2.3.

2.3.1 Definition of discrete wavelet

A formal re-definition of a wavelet function from continuous to discrete time can be made in three ways [2]:

Definition 2.3.1 *A wavelet is a function $\psi(t)$ of $L^2(\mathbb{R})$[4] whose Fourier transform $\hat{\psi}(\omega)$ satisfies the condition (continuous sense):*

$$\int_0^{+\infty} |\hat{\psi}(\omega t)|^2 \frac{dt}{t} = 1 \tag{2.19}$$

Definition 2.3.2 *A wavelet is a function $\psi(t)$ of $L^2(\mathbb{R})$ whose Fourier transform $\hat{\psi}(\omega)$ satisfies the condition (discrete sense):*

$$\sum_{-\infty}^{+\infty} |\hat{\psi}(2^j \omega)|^2 = 1 \tag{2.20}$$

Definition 2.3.3 *A wavelet is a function $\psi(t)$ of $L^2(\mathbb{R})$ such that*

$$\psi_{j,k} = 2^{-j/2}\psi(2^{-j}t - k) \tag{2.21}$$

is an orthonormal basis for $L^2(\mathbb{R})$.

From Def.2.3.2 and Def.2.3.3, each subspace is in fact defined by an integer power of two. The resulting sequence of nested subspaces of $L^2(\mathbb{R})$ is known as a *multi-resolution approximation* or (*analysis*).

[4] $L^2(\mathbb{R})$ is the set of square integrable functions in a measure space X.

2.3.2 Multiresolution Approximation

The effectiveness of the multiresolution analysis consists in splitting the space $L^2(\mathbb{R})$ into a sequence of subspaces V_j, where $j \in \mathbb{Z}$ each spanned by a an orthonormal basis of translates of a single function $\phi(t)$, described by the expression:

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \tag{2.22}$$

where $k \in \mathbb{Z}$. These subspaces must satisfy the following properties:

Property 2.3.1 *The j-th subspace of the sequence is nested into the (j+1)-th subspace:*

$$V_j \in V_{j+1} \qquad \forall j \in \mathbb{Z}$$

Property 2.3.2 *The union of all the subspaces is dense*[5] *in $L^2(\mathbb{R})$, whereas their intersection is the empty set:*

$$\begin{aligned} \bigcup_{j=-\infty}^{+\infty} V_j &= L^2(\mathbb{R}) \\ \bigcap_{j=-\infty}^{+\infty} V_j &= \{\varnothing\} \end{aligned}$$

Property 2.3.3

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1} \qquad \forall j \in \mathbb{Z}$$

Property 2.3.4 *V_0 has an orthogonal basis of translates $\phi(t-k)$, $\forall k \in \mathbb{Z}$*

If all properties hold, the approximation of a function $f(t) \in L^2(\mathbb{R})$ at the resolution 2^j is the orthogonal projection of $f(t)$ onto V_j[6]. The construction of the orthogonal projection starts by finding a unique function $\phi(t) \in L^2(\mathbb{R})$. A *wavelet basis*, related to $\phi(t)$, evolves from the additional information carried by an approximation of resolution 2^{j+1} compared with the resolution 2^j: this extra information resides in the projection W_j of the orthogonal complement of V_j in V_{j+1}; *i.e.* $V_{j+1} = V_j \oplus W_j$. This space W_j will be spanned by the wavelet orthonormal basis defined in Def.2.3.3. The relation among the resulting subspaces is, then:

$$V_{j+1} = V_j \oplus W_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_j \tag{2.23}$$

[5] A subset A of a topological space X is dense in X if and only if the only closed subset of X containing A is X itself.

[6] It is an orthogonal projection because each basis function of the approximation is orthogonal to the others.

2.3.3 The construction

The construction of the functions needed for the decomposition starts with the solution $\phi(t)$ of the ***dilation equation***, also known as ***scaling function*** or ***father wave***:

$$\phi(t) = \sum_k c_k \phi(2t-k) \tag{2.24}$$

where the c_k constitute a set of coefficients. Imposing some *requirements*, those coefficient c_k get some restraints that will lead to some useful properties of the functions to obtain.

The first requirement is the *normalization*: $\int \phi(t)dt = 1$. Multiplying by 2 and integrating Eq.2.24 brings the first condition on c_k:

$$2\int \phi(t)dt = \sum_k c_k \int \phi(2t-k)\, d(2t-k)$$

which yields:

$$\sum_k c_k = 2 \tag{2.25}$$

A further condition comes from the degree of *accuracy* $p-1$ with which the polynomials are meant to be reproduced exactly by the approximating functions (thus presenting an error of order h^p). As found by Daubechies and recalled by Strang in [27], this is related to the vanishing moments of the FT of ϕ. In [27] the condition on the c_k is referred to as “Condition A” and yields:

$$\sum (-1)^k k^m c_k = 0 \qquad m = 0, 1, \ldots, p-1 \tag{2.26}$$

Another condition, always in [27], is called “Condition O” and is related to the *orthogonality* of the scaling function with any of its shifts by an even number of coefficients:

$$\sum c_k c_{k-2m} = 2\delta_{0,m} \tag{2.27}$$

where $\delta_{0,m}$ is the Kronecker Delta in Eq.2.7.

The corresponding ***wavelet*** or ***mother wave*** is then defined as:

$$\psi(t) = \sum_k (-1)^k c_{1-k}\, \phi(2t-k) \tag{2.28}$$

Satisfying all of these conditions leads to families of orthonormal wavelets like the ones that will be used in this work.

2.3.4 Wavelet and scaling function coefficients

Let the attention be focused on the coefficients of the scaling function and the wavelet in order to find the relation between them. Let the recursive definition of the scaling function from Eq.2.22 be applied onto its other definition in Eq.2.24, declining it in the case of the first stage of the transform. In such a way, a new set of coefficients h_k can be spotted:

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{+\infty} h_k \phi(2t-k) \qquad t \in \mathbb{R} \tag{2.29}$$

and so:

$$\sum_{k=-\infty}^{+\infty} c_k = \sqrt{2} \sum_{k=-\infty}^{+\infty} h_k \tag{2.30}$$

Imposing the *normalization* of the scaling function in the form of Eq.2.29, like in Sec.2.3.3, a condition on the coefficients h_k is then obtained:

$$\sum_{k=-\infty}^{+\infty} h_k = \sqrt{2} \tag{2.31}$$

Likewise, $\psi(t)$ can be re-written as:

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{+\infty} g_k \, \phi(2t-k) \tag{2.32}$$

with coefficients g_k that must satisfy the condition $\langle \mathbf{h}, \mathbf{g} \rangle = 0$[7] in order for $\psi(t)$ to be orthogonal to $\phi(t)$. This means that:

$$g_k = (-1)^k \, \overline{h_{1-k}} \tag{2.33}$$

Besides, since the wavelet zero moment must vanish (or, equivalently, its mean is null, its nature is oscillatory) as seen in Thm.2.2.1, the condition $\int_{-\infty}^{+\infty} \psi(t) dt = 0$ leads to:

$$\sum_{k=-\infty}^{+\infty} g_k = 0 \tag{2.34}$$

In discrete time, $\{h_k\}$ and $\{g_k\}$ constitute the filter bank that will achieve the decomposition of the signal. Actually, these filters are of finite length L and their construction formulas will be adapted consequently (see Sec.2.3.6).

To sum up, starting from continuous time wavelet transform with its father and mother wave, a re-definition has been operated in order to finally find a truly Discrete Wavelet Transform of practical implementation.

[7] $\mathbf{h}$ and $\mathbf{g}$ are the column vectors of the h_k and g_k coefficients, respectively.

2.3.5 The DWT algorithm

Once the signal is sampled, the domain becomes discrete and, then, the terms *function* and *sequence* can be used interchangeably for $f(n)$, where $n \in \mathbb{Z}$.

As anticipated, the procedure is also made of discrete filtering operations. The mathematical formula describing the action of a filter is the discrete convolution, defined as in Def.2.3.4.

Definition 2.3.4 *The convolution operation between the discrete-time sequences $x(n)$ and $y(n)$ is defined as:*

$$x(n) * y(n) = \sum_{k=-\infty}^{+\infty} x(k) \cdot y(n-k) \tag{2.35}$$

This operation in the time domain is expressed in the frequency domain by Propty.2.3.5.

Property 2.3.5 *If a convolution in the time domain (Eq.2.35) between two sequences $x(n)$ and $y(n)$ gives the sequence $w(n)$, this corresponds to a multiplication of the FTs of those sequences, such that:*

$$w(n) = x(n) * y(n) \quad \rightarrow \quad \hat{w}(\omega) = \hat{x}(\omega) \cdot \hat{y}(\omega) \tag{2.36}$$

The DWT algorithm combines this filtering with a down-sampling at each step of decomposition. To clarify the consequences of these operations, let us consider the first level of decomposition, *i.e.* the application of the first pair of filters, in particular the low-pass one. If a sequence $f(n)$ is sampled with a rate f_s, the maximum observable frequency component is $f_{max} = f_{nyq} = f_s/2$. Passing the signal through a low-pass filter halves such maximum frequency[8]. According to Nyquist rule, since the signal now has a highest frequency of $\pi/2$, a half of the samples can be discarded because redundant, *i.e.* the signal can be *sub-sampled*.

The resolution, which is related to the quantity of information in the signal, is affected only by the filtering: removing half the frequencies generally means discarding a half of the information (*i.e.* a half of the resolution). Anyway, removing a half of the spectral components by down-sampling a filtered signal in which a half of the samples is redundant, therefore means getting no loss of

[8] Recall that, talking about discrete time, the proper unit of frequency is the radian. The rule for conversion from a continuous-time frequency f to discrete-time radial frequency ω is: $\omega = 2\pi f t_s$, where t_s is the sampling time. Accordingly, the sampling rate is equal to 2π radians in terms of radial frequency and Nyquist rate is π radians (considering $f = f_s = 1/t_s$). Nonetheless, common use and readability allow to talk about frequency in terms of hertz when preferred.

information. The down-sampling, instead, doubles the scale, letting the process of decomposition proceed similarly to the CWT. Note that the sub-sampling after filtering does not affect the resolution.

To sum up: (i) the filtering halves the resolution, but leaves the scale unchanged; (ii) the down-sampling by 2 doubles the scale, but does not affect the resolution.

The procedure of the DWT consists of repetitions of the same filtering operations followed by a sub-sampling by a factor of 2. This results in several *levels of decomposition*. At the first stage, the low-pass filtering through $h(n)$ provides the coarse ***approximation*** of $f(n)$ ($a(k)$, Eq.2.37), which is associated to a *set of scaling functions*. The high-pass filtering through $g(n)$, instead, returns its ***detail*** ($d(k)$, Eq.2.38) and is associated to a *set of wavelet functions*. In mathematical terms, according to the definitions of convolution in Def.2.3.4 and of down-sampling by 2 (see Def.2.4.1, in Sec.2.4.4), this first step of the algorithm is:

$$a(k) = \sum_n f(n) \cdot h(2k - n) \tag{2.37}$$

$$d(k) = \sum_n f(n) \cdot g(2k - n) \tag{2.38}$$

This operations can be applied recursively on each level of approximation a_j. Therefore, Eq.2.37 and Eq.2.38 can be generalized:

$$a_{j+1}(k) = \sum_n a_j(n) \cdot h(2k - n) \tag{2.39}$$

$$d_{j+1}(k) = \sum_n a_j(n) \cdot g(2k - n) \tag{2.40}$$

Iterating this “filtering & down-sampling” scheme as in Fig.2.3, a finer resolution in frequency is obtained increasing the level of decomposition since the sub-bands get narrower. On the contrary, time resolution worsens because at each level the signal is described with a half of the samples of the previous level. Summarizing, this so-called *sub-band algorithm*, schematized in Fig.2.3 offers this benefits:

- good time resolution at high frequencies;
- good frequency resolution at low frequency.

The spectral components that are most prominent in the original signal will appear with the highest magnitudes (coefficients of the transform) in the sub-band containing them.

The signal can then be reconstructed with the inverse process: the decomposed signal gets up-sampled by two, passed through the synthesis filters

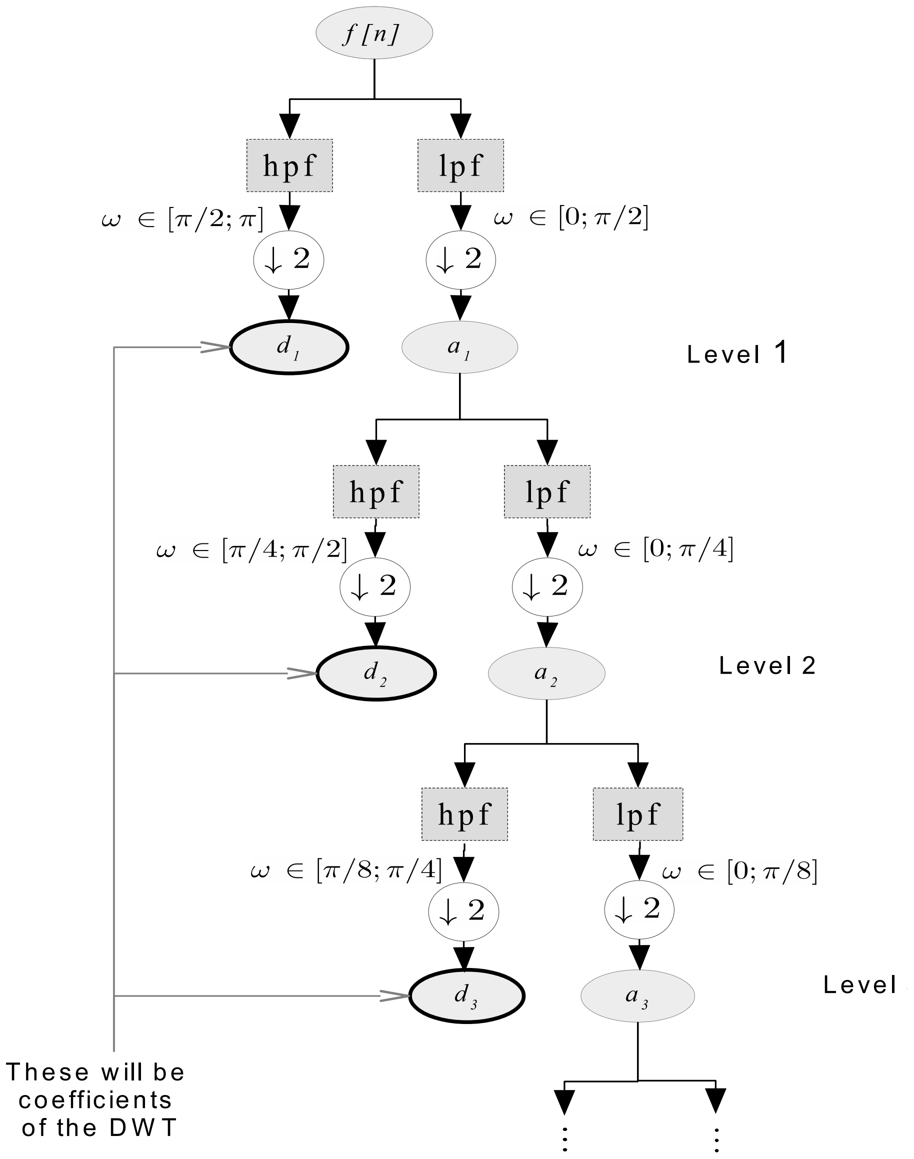


Figure 2.3: The sub-band coding algorithm generates an octave band subdivision of the spectrum by successive filtering and down-sampling by 2 the approximation and keeping the detail.

$h'(n) = h(-n)$ and $g'(n) = g(-n)$ and then added:

$$f(n) = \sum_{k=-\infty}^{+\infty} (d(k) \cdot g(-n+2k)) + (a(k) \cdot h(-n+2k)) \tag{2.41}$$

Except for a time reversal, the synthesis and analysis filters are equal.

2.3.6 Adaptation to finite-length signal

Practical signals (like acquisitions of tension, current, *etc.*) are discrete-time datasets of finite length N coming from measures of continuous signals passed through an Analog-to-Digital Converter (ADC) during an interval of time $\Delta T = N\, t_s$, where t_s is the sampling period. There are some constraints on the length of such signals, as treated in Sec.2.3.7. A possible adaptation of the signal due to these conditions will return a new sequence of length M equal to the closest power of two respect to N. The filters, too (better analyzed in Sec.2.3.8), will be finite sequences of even length $2L$. Therefore, the formulas found since now cannot be directly applied to a actual case as they are, they must be adapted. The algorithm will then be expressed by modified versions of Eq.2.39 and Eq.2.40:

$$a_{j+1}(k) = \sum_{n=0}^{M-1} a_j(n) \cdot h(2k-n) \tag{2.42}$$

$$d_{j+1}(k) = \sum_{n=0}^{M-1} a_j(n) \cdot g(2k-n) \tag{2.43}$$

where $h(n)$ and $g(n)$ are visited in a circular way, *i.e.* if $m \in [0; 2L]$, $g(-m) = g(2L-m)$.

2.3.7 The number of samples

Considering the generic sequence of length N as the function to decompose, there is no assurance that a signal of such length can be properly handled. Actually, the choice of the factor 2 in Sec.2.3 has a huge impact on the number of samples that can be taken into account. Since the number of samples at each level is a half with respect to the previous level, there is a logarithmic decrease through the levels. In order not to lose information and consistency for the algorithm, the only samples discarded must be due only to the sub-sampling because redundant and not due to a disarrangement in the number of samples. Therefore, the number of samples taken into account must be a power of 2, *e.g.* $M = 2^{l_{max}}$, where $l_{max} \in \mathbb{N}$ is, therefore, the maximum level of decomposition.

If $M \neq N$, the original sequence of N samples must be adapted to the closest power-of-two number of samples M in one of the following methods:

Cancellation ($N > M$) Portions of the signal are canceled because of an exceeding number of samples. Roughly, the datasets that can be discarded are chosen among three main options: (i) the first $|N-M|$ samples; (ii) the last $|N-M|$ samples; (iii) the first n_1 and the last n_2 samples, such that $n_1+n_2 = |N-M|$. Note that, in order not to alter the spectral content, the samples deleted must all be consecutive (considering the sequence stored in a circular array). Discarding some random values, instead, would alter the spectrum. A decimation in order to reduce the number of samples now would have the same effect and is, thus, an option to be avoided.

Zero-Padding ($N < M$) A $|N-M|$ number of zeros is put either (i) at the rear; (ii) at the front; (iii) somehow distributed between the rear and the front. No zeros ought to be put within the signal, otherwise altering it. With this method, the signal is like a sudden manifestation of an event, ending in turn in a sudden way.

Periodization ($N < M$) The signal is considered as if it was periodic. Then, three possibility are to take into account: (i) the last $|N-M|$ samples are repeated at the beginning of the signal; (ii) the first $|N-M|$ samples are repeated at the end of the signal; (iii) the first n_1 samples are repeated at the end of the original signal and the last n_2 samples are repeated at the beginning, such that $n_1 + n_2 = |N-M|$.

Before the transform, the (somehow obtained) M samples describe the signal in the time domain, whereas, after a complete decomposition (*i.e.* up to a level $l_{max} = \log_2 M$), the resulting M samples describe the signal components with a specific resolutions in time and frequency, according to the level attained. Indeed, the best frequency resolution is related to the highest possible level, *i.e.* $\pi/2^{l_{max}}$, among the $(l_{max}+1)$ available sub-bands (l_{max} of detail and one of coarsest approximation).

2.3.8 The DWT filters

The low-pass ($h(n)$) and high-pass ($g(n)$) filters used in the DWT are commonly known as *Quadrature Mirror Filters* and correspond to the scaling and wavelet functions of the DWT. They are defined as in Def.2.3.5 [33]:

Definition 2.3.5 *In discrete signal processing, a Quadrature Mirror Filter pair is a set of two filters $(h_0(n), h_1(n))$ whose magnitude responses $(H_0(z), H_1(z))$ are the mirror image around $\pi/2$ of one another,* i.e. *they satisfy Eq.2.44 and Eq.2.45:*

$$H_1(z) = H_0(-z) \tag{2.44}$$

$$|H_1(e^{j\Omega})| = |H_0(e^{j(\pi-\Omega)})| \tag{2.45}$$

where the frequency is $\Omega = \pi/2$.

For orthogonal wavelets, a further property is added to the analysis filters in order not to alter the distribution of the energy of the signal:

Property 2.3.6 *The power complementarity property of a filter bank imposes that the power sum of the filters is the unit,* i.e.*:*

$$|H_0(e^{j\Omega})|^2 + |H_1(e^{j\Omega})|^2 = 1 \tag{2.46}$$

where the sampling rate is normalized to 2π *[33].*

Let us consider the filter pair $h(n)$ and $g(n)$ of the analysis functions in Eq.2.37 and Eq.2.38. For orthogonal wavelets they are Finite Impulse Response (FIR) of length $2L$ in number of points so that they can be called *Conjugate Mirror Filters* as recalled in [15]. Their expression in discrete time can be seen in Eq.2.47 [25]:

$$g(n) = (-1)^n \, h(2L - 1 - n) \tag{2.47}$$

Note that they are the odd index alternated reversed version of one another. The conversion from low-pass to high-pass is provided by the factor $(-1)^n$.

2.4 The Wavelet Packet Transform

The “classic” DWT, as described in Sec.2.3, chooses a particular basis for the decomposition, that is the coarsest level of approximation and all the details at every other level. This approach can be abridging for the analysis of the characteristics of a generic unknown signal: actually, no assumption can be made *a priori* on the spectrum of the signal and the frequency or time resolution needed for each part of it. There is no certainty that high frequency resolution is needed only at low frequency, for instance.

A more flexible and reliable method for signal analysis is a modification of the DWT algorithm, obtained performing the sub-band coding not only on the approximation a_j, but also on the detail d_j of every level j of decomposition. This adjustment of the DWT is called Wavelet Packet Transform (WPT) [32]. The same limitations on the number of samples of Sec.2.3.7 are still effective. The WPT provides a sort of transition from a view of the signal purely in the time domain to multiple views of it with an increasing resolution in frequency up to $\pi/2^l$, where l is the level chosen to stop the decomposition (the finest uncertainty available is still of $\pi/2^{l_{max}}$ with a complete decomposition). Even if the information may be redundant, this kind of decomposition provides several possible trade-offs between the resolutions in time and frequency in every part of the spectrum.

To provide a graphical interpretation of peculiarities of the DWT and the WPT, a representation of the history of the coefficients of every level along the decomposition was made in Fig.2.4a, Fig.2.5a and Fig.2.6a of Ex.2.4.1.

Another intuitive representation of the WPT algorithm is the so-called Wavelet Packet Tree (WP Tree): this is a binary tree that describes the splitting of the spectral and time content of the WPT throughout the levels of decomposition. Starting from the root (*i.e.* the top node in Fig.2.4b, Fig.2.5b and Fig.2.6b of Ex.2.4.1), where the original signal resides, each node gives birth to two other nodes representing its low-pass (left child) and high-pass (right child) filtered versions, (down-sampled by a factor of two, according to the sub-band coding algorithm). Consequently, the *number of nodes* within a level j of decomposition is 2^j. The *number of samples* λ_j contained *in each node* of level j, accordingly, is equal to:

$$\lambda_j = M/2^j = 2^{l-j} = 2^\eta \tag{2.48}$$

where $M = 2^{l_{max}}$ is the total amount of samples and $\eta \in \mathbb{N}$. Each node spans a range of frequencies that can be calculated as:

$$\left[p\frac{\pi}{2^j}; (p+1)\frac{\pi}{2^j}\right] \tag{2.49}$$

where j is the level of decomposition and p the progressive number of the node within that level (*cf. Sec.2.4.1.* For attaining a more intuitive formula to understand, the discrete frequency in radians can be rewritten in hertz (though improperly), so that the intervals spanned by each node are[9]:

$$\left[p\frac{f_s}{2^{j+1}}; (p+1)\frac{f_s}{2^{j+1}}\right] \tag{2.50}$$

Each level, thus, provides an equal-width sub-band division of the spectral content, furnishing a finest resolution of $f_s/2^{l+1}$ at the highest level chosen. Nevertheless, as will be examined in Sec.2.4.4, such sub-bands do not respect a pleasingly intuitive frequency order like the STFT, due to the effect of aliasing. Before adjusting the tree order, though, its notation (Sec.2.4.1), internal relations (Sec.2.4.2) and peculiarities Sec.2.4.2) must be analyzed in depth.

2.4.1 The notation of the WPT

The nodes of the WP Tree are named after their level j and their progressive number p in the level, thus being described by a couple (j, p_j), where $j = 0, 1, \ldots, l$ (with $l_{max} = \log_2 M$) and $p_j = 0, 1, \ldots, (2^j - 1)$. Though the number p_j has a maximum value depending on the level in exam, for the sake of readability, it will be denoted just by the notation p. This notation can be substituted by a serial numbering $m = m(j, p)$, where $m(0, 0) = 1$ is the root. Given the notation of the node by its couple (j, p), indeed, the serial number of the node can be obtained by Eq.2.51:

$$m(j, p) = 2^j + (p+1) \tag{2.51}$$

[9] Recall that $2\pi[rad] \rightarrow f_s[Hz]$.

Inversely, given the serial number m of a node, it is always possible to get its level and progressive number within that level. Let the discrete Heaviside step function be adapted to this case as:

$$\Theta(m, 2^i) = \begin{cases} 1, & \text{if } m \geq 2^i \\ 0, & \text{if } m < 2^i \end{cases} \tag{2.52}$$

where $m \in \mathbb{N}_0 : m = 1, \dots, (2^l - 1)$ and $i \in \mathbb{N} : i = 0, \dots, l$. The level j of a node given in the serial notation m can be found as:

$$j(m) = j_m = \sum_{i=1}^{l} \Theta(m, 2^i) \tag{2.53}$$

and the progressive number in that level j_m is:

$$p(j_m) = m - 2^{j_m} \tag{2.54}$$

where $p \in \mathbb{N} : p = 0, \dots, (2^{j_m} - 1)$.

2.4.2 The Parent-Child Relations in the WP Tree

According to the notation (j, p) introduced in Sec.2.4.1, it is possible to relate each node of the WP Tree with its parent and its children. The relations that can be achieved constitute two useful instruments for an easier interpretation of the spectral distribution along the tree. The following properties (Propty.2.4.1 and Propty.2.4.2) clarify these interconnections:

Property 2.4.1 *Let the a node be denoted by a couple (j, p), where $j \in \mathbb{N} : j = 0, \dots, (l_{max} - 1)$* [10] *and $p \in \mathbb{N} : p = 0, \dots, (2^j - 1)$. Its children nodes can be computed as:*

$$(j, p) \rightarrow \begin{cases} (j+1, 2p) \\ (j+1, 2p+1) \end{cases} \tag{2.55}$$

Property 2.4.2 *Let the a node be (j, p), where $j \in \mathbb{N} : j = 1, \dots, l_{max}$* [11] *and $p \in \mathbb{N} : p = 0, \dots, (2^j - 1)$. Its parent can be computed as:*

$$(j, p) \rightarrow \left(j - 1, \left\lfloor \frac{p}{2} \right\rfloor\right) \tag{2.56}$$

This knowledge is useful while performing search methods on the tree, like in Sec.2.4.6.

[10] Note that the excluded level l_{max} cannot have children.

[11] Note that the excluded level 0 cannot have a parent.

2.4.3 The differences between DWT and WPT

To explain the benefits of using a more computationally expensive method like the WPT instead of the classic DWT, let an example with a simple generic signal $f(n)$ be considered, Ex.2.4.1.

Example 2.4.1 *Let us consider the simple case of an acquisition of a continuous wave $f(t)$ at the sampling frequency $f_s = 1/t_s$. Let the signal in exam be a sequence $f(n) = f(nt_s)$ with a number of samples $N = 8$, so that, for the sake of simplicity, $n \in \mathbb{N} : n = 0, 1, \ldots, 7$. According to Sec.2.3.7, being N a power of two, no adaptation of the number of samples is needed, thus the number of considered samples is $M = N = 8 = 2^3$.*

*The signal is then a discrete function $f(n) = \{f_0, f_1, \ldots, f_7\}$. Referring to the results of Sec.2.3.6 for the DWT, passing the sequence a_j at a certain level j of decomposition (where $a_0(n) = f(n)$) through the low-pass filter $h(n)$ gives an approximation $a_{j+1}(n)$, whereas passing it through the high-pass filter $g(n)$ returns the detail d_{j+1}. As stated in Sec.2.4, in the WPT the sub-band coding is not applied solely on the approximation, but also on the detail, with a consequent proliferation of the nodes of the WP Tree. Therefore, the notation has to become slightly more articulated. In order to give a track of the history of each node, every time a portion of the signal passes through the low-pass filter, an "a" is placed before its current name, whereas a "d" is put if it passes through the high-pass filter (*cf. *Fig.2.4a, Fig.2.5a and Fig.2.6b, for instance).*

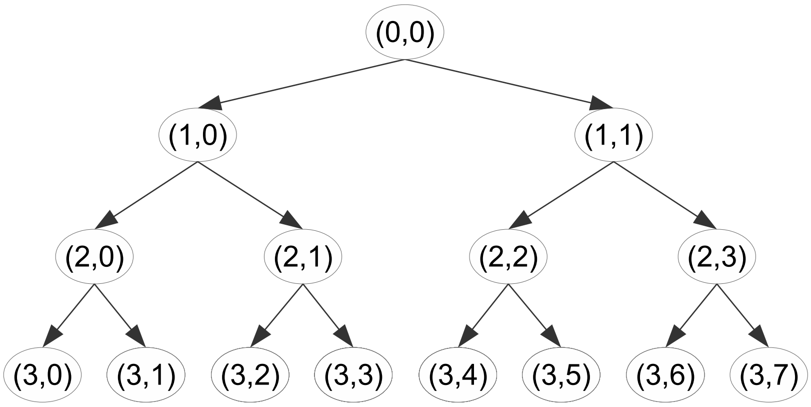
According to Sec.2.4, the WPT gives a full decomposition of the signal up to a maximum level $l_{max} = \log_2 M = 3$, as shown in Fig.2.4a. At level $l = l_{max}$, the time resolution in lost, but the uncertainty in frequency is the smallest obtainable $\Delta\omega = \pi/2^{l_{max}} = \pi/8$ (or equivalently $\Delta f = f_{nyq}/8 = f_s/16$). One of the major benefits of the WPT in comparison to the DWT is that the finest frequency resolution is available all through the spectrum of the signal, not only at low frequencies. Reciprocally, a fine time resolution is possibly disposable also for low frequencies.

As can be seen in Fig.2.5a, the classic wavelet basis of the DWT itself can be extracted as a particular selection from the Wavelet Packet Decomposition (WPD). The orthogonality of this basis can intuitively be spotted thanks to the absence of "vertical superposition" for the selected coefficients, which means there is no redundancy and the basis spans the whole space $\mathbb{R}^8$.

From the WPT, many other orthogonal basis can be extracted according to the necessities of each single case. An example can be seen in Fig.2.5, called "sub-band basis" in [32]. The chosen nodes are all at the same level, so that the result is similar to what we could obtain from STFT, with fixed-duration interval of time on which the transform is performed. In this case, the advantage is that each level presents a different trade-off between the resolutions in time and frequency: the choice is up to the user.

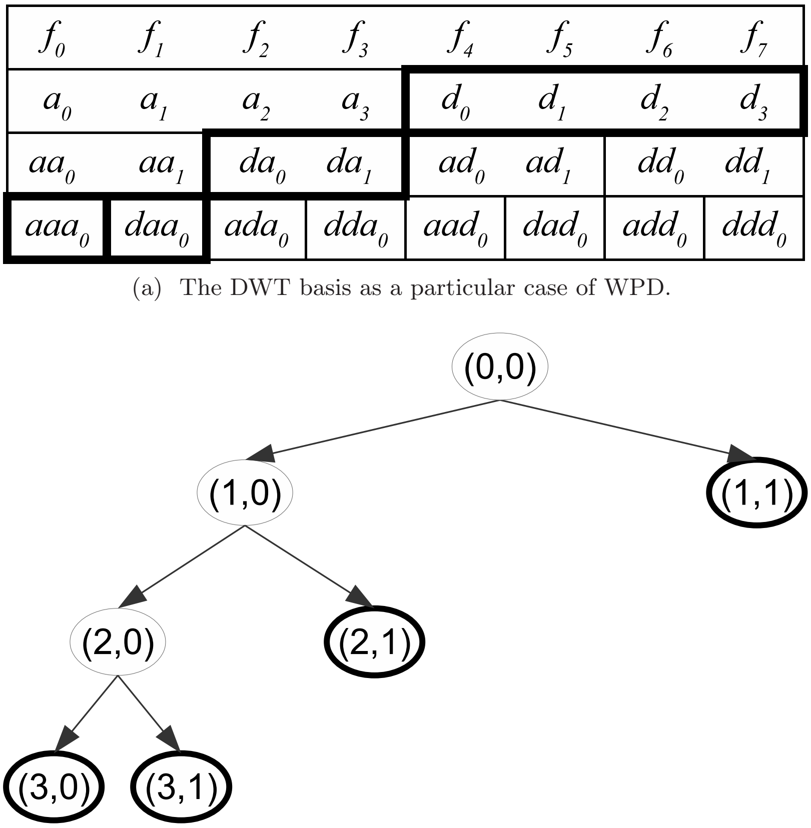
f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
a_0	a_1	a_2	a_3	d_0	d_1	d_2	d_3
aa_0	aa_1	da_0	da_1	ad_0	ad_1	dd_0	dd_1
aaa_0	daa_0	ada_0	dda_0	aad_0	dad_0	add_0	ddd_0

(a) WPT full decomposition. No particular basis is selected.



(b) Full WP Tree. No particular basis is selected.

Figure 2.4: The WPT provides a full decomposition of every part of the signal, not only of the approximation.

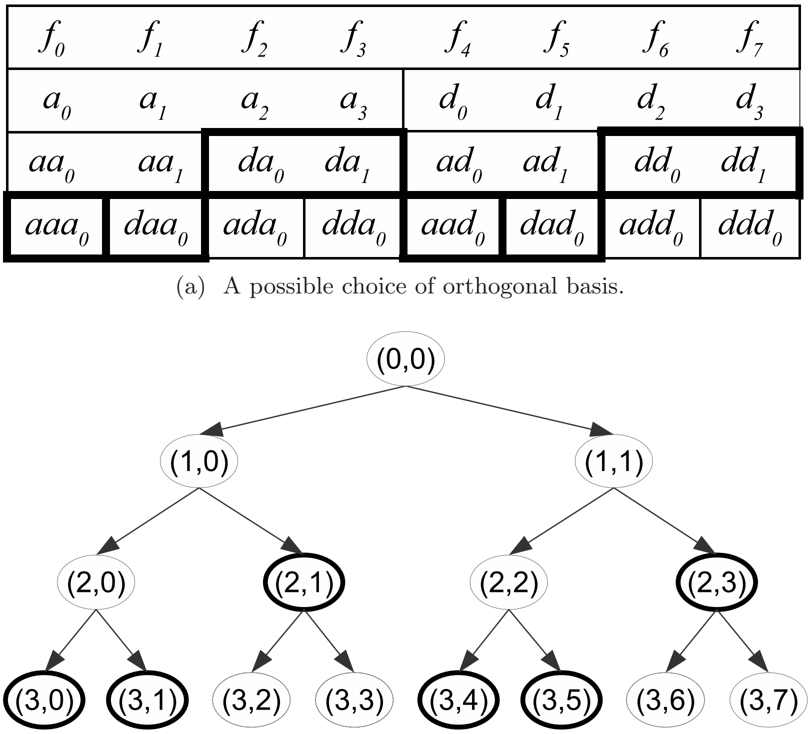


f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
a_0	a_1	a_2	a_3	d_0	d_1	d_2	d_3
aa_0	aa_1	da_0	da_1	ad_0	ad_1	dd_0	dd_1
aaa_0	daa_0	ada_0	dda_0	aad_0	dad_0	add_0	ddd_0

(a) The DWT basis as a particular case of WPD.

(b) Particularization of the WP Tree with the classic DWT basis.

Figure 2.5: The WPT provides a full decomposition of every part of the signal. The choice of a particular basis is up to the user. The classic DWT is just a possible choice.



f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
a_0	a_1	a_2	a_3	d_0	d_1	d_2	d_3
aa_0	aa_1	da_0	da_1	ad_0	ad_1	dd_0	dd_1
aaa_0	daa_0	ada_0	dda_0	aad_0	dad_0	add_0	ddd_0

(a) A possible choice of orthogonal basis.

(b) WP Tree of a generic orthogonal basis.

Figure 2.6: The WPT provides a full decomposition of every part of the signal. Many bases can be chosen, some of them orthogonal. This is just an example.

Moreover, many combination of nodes forming an orthogonal basis are possible performing the WPD. In Fig.2.6 there is an example. There must not be redundancy, i.e. *if a child node is chosen, its parent must not be picked.*

2.4.4 The effects of aliasing

As seen in Sec.2.4, the WPT follows the sub-band coding algorithm passing the signal through discrete-time filters, which are then down-sampled by a factor of 2. A well-known collateral effect provoked by this chain of actions is the phenomenon of ***aliasing***.

The aliasing is a distortion that causes different signals to become indistinguishable (or “alias” of one another) due to a “slow” sampling rate. Filtering itself does not cause aliasing, since the filters $h(n)$ and $g(n)$ are designed to split the spectral content of the signal exactly in two parts without side-effects. Down-sampling the high-passed signal, instead, may cause undesired effects. To explain this by-product of the WPT, first of all, *down-sampling* itself must be defined:

Definition 2.4.1 *Down-sampling a sequence $x(n)$ by a factor α means generating a sequence $y(n)$ such that [4]:*

$$y(n) = x(\alpha n) \tag{2.57}$$

In the frequency domain, this means:

$$\hat{y}(z) = \frac{1}{\alpha} \sum_{m=0}^{\alpha-1} \hat{x}(e^{-j\frac{2\pi}{\alpha}m} z^{\frac{1}{\alpha}}) \tag{2.58}$$

where $\alpha \in \mathbb{N}$.

To better understand what this means, two properties of the Z-Transform must be recalled [31]:

Property 2.4.3 *Time scaling for a signal $x(n)$ by a factor $\alpha \in \mathbb{N}$:*

$$\mathcal{Z}[x(\alpha n)] = \hat{x}(z^{\frac{1}{\alpha}}) \tag{2.59}$$

Property 2.4.4 *Time shifting for a signal $x(n)$ by a quantity $n_0 \in \mathbb{Z}$:*

$$\mathcal{Z}[x(n - n_0)] = z^{-n_0}\hat{x}(z) \tag{2.60}$$

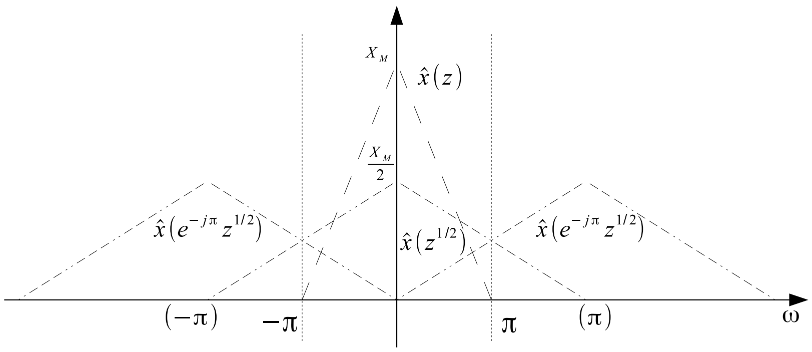


Figure 2.7: Let $\hat{x}(z)$ be the spectrum of a discrete signal $x(n)$ and its maximum amplitude be X_M. Down-sampling by 2, which means scaling by a factor 2 and shifting by π, according to Eq.2.61, causes the summation of the stretched version $\hat{x}(z^{1/2})$ and $\hat{x}(e^{-j\pi}z^{1/2})$ of the original spectrum, normalized by a factor $\frac{1}{2}$. Note that shifting the dilated signal by π is equivalent to a translation of 2π on the ω-axis of the original signal (this is indicated by means of the parentheses).

In the case of the sub-band coding algorithm, according to the notation of Def.2.4.1, the under-sampling factor is $\alpha = 2$.

As already recalled, at each step of the decomposition, the sequences in the nodes of the WP Tree pass through a low-pass filter $h(n)$ and a high-pass $g(n)$ filter. Then, let us refer to the filtered versions of a sequence at node (j, p), $\varphi_{(j,p)}(n)$, as φ_h and φ_g, respectively (where $\varphi_{(j,p)}(n)$ can either be the approximation $a_{(j,p)}(n)$ or the detail $d_{(j,p)}(n)$ in a generic node (j, p)). In particular, let the high-passed signal $\varphi_g(n)$ be taken into account. Then let it go through the down-sampling stage, so that its detail $d_{(j+1,p')}(n) = d(n)$ is extracted (with $p' = 2p + 1$, as seen in Sec.2.4.2), purged of redundant information. Mathematically, it means that $d(n) = \varphi_g(2n)$. In the frequency domain, this implies that:

$$\begin{aligned}\hat{d}(z) &= \frac{1}{2}\Big[\hat{\varphi}_g(e^{-j\frac{2\pi}{2}\cdot 0}z^{\frac{1}{2}}) + \hat{\varphi}_g(e^{-j\frac{2\pi}{2}\cdot 1}z^{\frac{1}{2}})\Big]\\ &= \frac{1}{2}\Big[\hat{\varphi}_g(z^{\frac{1}{2}}) + \hat{\varphi}_g(e^{-j\pi}z^{\frac{1}{2}})\Big]\end{aligned} \tag{2.61}$$

Therefore, the meaning of Eq.2.61 is that the down-sampling causes the stretching and the superposition of two copies of the signal (the factor $\frac{1}{2}$ is a normalization coefficient), thus provoking aliasing. In Fig.2.7 there is a simple example of this phenomenon.

To comprehend the side effects of the down-sampling along the decomposition, Ex.2.4.2 is provided. Such example visually shows how aliasing affects the WPT in Fig.2.8[12] [10]: not causing a loss of information due to

[12] Note that these figures do not take into account the normalization factor 1/2, since their

indistinguishable signals, but rather the *inversion* of the sequentiality of the spectral content spanned by some sibling nodes.

Example 2.4.2 *Let the spectrum obtained by the FT of the original signal $f(n)$ be $\hat{f}(\omega)$ (Fig.2.8a). Passing it once through the high-pass filter returns its high-frequency half, $\hat{f}_g(\omega)$ (Fig.2.8b). Then, the down-sampling changes the scale of the ω-axis by a factor of 2, getting a sequence whose spectrum will be referred as $\hat{d}(\omega)$ (Fig.2.8c). The result of this down-sampling is the reversed version of the high-pass signal [10]. Going on with the decomposition, therefore, the spectrum of the low-passed sequence $\hat{d}_h(\omega)$ (Fig.2.8d) and, as a consequence, of its decimated version $\widehat{ad}(\omega)$ (Fig.2.8e) is of higher frequency than the high-passed sequence $\hat{d}_g(\omega)$ (Fig.2.8f) and, thus, its down-sampled version $\widehat{dd}(\omega)$) (Fig.2.8g).*

This inversion must be corrected to give a proper description of the frequency content of the signal.

2.4.5 The sequency order

As seen in Ex.2.4.2, not all of the nodes of the WP Tree are ordered according to frequency: the disposal they get after the WPT is called *"natural"* or *Paley's order*. By either slightly modifying the algorithm of the WPT or re-ordering the tree later, a more pleasingly intuitive ***frequency*** or ***"sequency" order*** can be achieved, so that the inversions seen in Sec.2.4.4 are adjusted. For the purposes of this work, ***frequency ordering must always be achieved***.

The reversals in the sequentiality of sub-bands occur when filtering and down-sampling specific nodes, according to a precise scheme. To identify where the corrections must be brought, first of all it is necessary to define what the "sequency" of a node is.

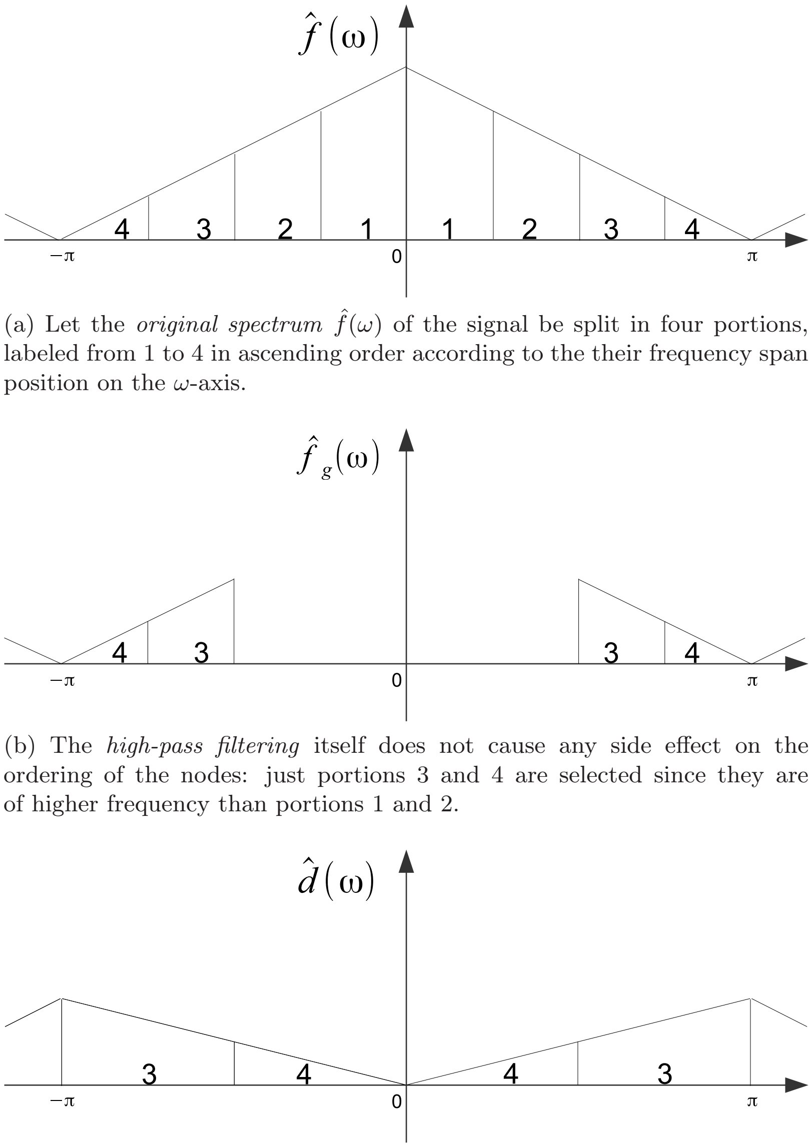
Definition 2.4.2 *The sequency of a node is the progressive number associated to its position on its level j of the WP Tree*[13]*, starting from the (even) sequency* 0 *at the leftmost node, up to the (odd) sequency* $(2^j - 1)$ *at the rightmost node, where $j \in \mathbb{N} : j = 0, \dots, l$.*

It can be proved that the children of the odd-sequency nodes have to be switched to obtain the frequency ordering of the tree, like in Fig.2.9[14]. As a

aim is rather to show the effect of inversion of the order of the frequencies spanned by the nodes.

[13] It corresponds to p using the notation in Sec.2.4.1.

[14] As a matter of fact, in Ex.2.4.2, the problematic node was that referred to as $d(n)$, in position $(1, 1)$, which has the odd sequency 1. Its children (occupying the positions $(2, 2)$ and $(2, 3)$) must be swapped with respect to the natural ordering. Besides, also the children of the node of coordinates $(2, 3)$ will have to be switched, since its sequency (3) is odd.

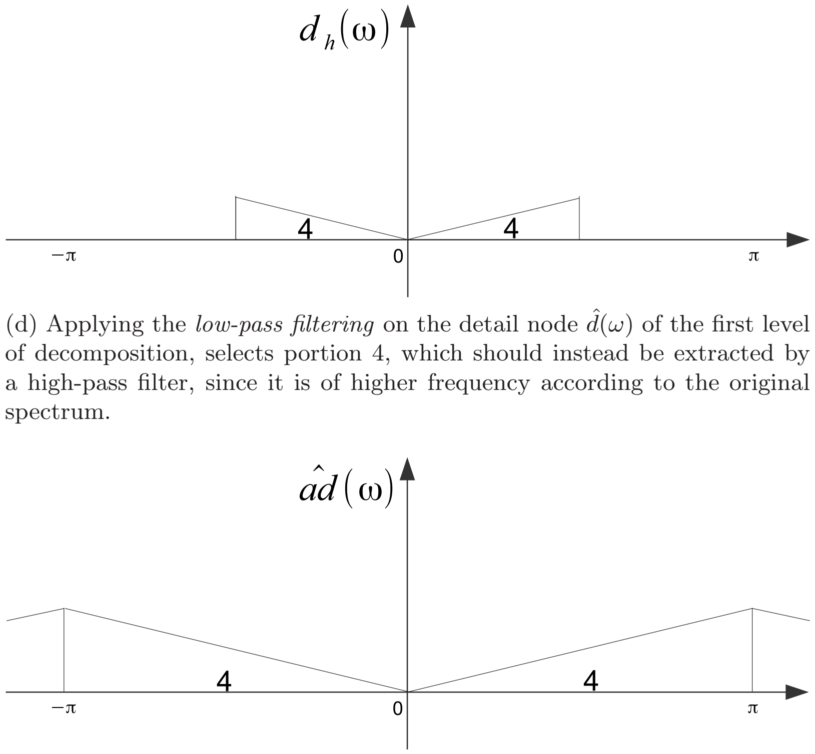


(a) Let the *original spectrum* $\hat{f}(\omega)$ of the signal be split in four portions, labeled from 1 to 4 in ascending order according to the their frequency span position on the ω-axis.

(b) The *high-pass filtering* itself does not cause any side effect on the ordering of the nodes: just portions 3 and 4 are selected since they are of higher frequency than portions 1 and 2.

(c) The scaling and shifting effects in the spectrum of the high-pass filtered signal due to the *down-sampling* provoke the overturning and the expansion of $\hat{f}_g(\omega)$. Now, portion 4 is located at lower frequencies than portion 3. This commonly known as *aliasing*.

Figure 2.8: (This example continues with figures Fig.2.8d, Fig.2.8e)



(d) Applying the *low-pass filtering* on the detail node $\hat{d}(\omega)$ of the first level of decomposition, selects portion 4, which should instead be extracted by a high-pass filter, since it is of higher frequency according to the original spectrum.

(e) *Down-sampling* $\hat{d}_h(\omega)$ gives the node containing $\widehat{ad}(\omega)$.

Figure 2.8: (This example continues with figures Fig.2.8f, Fig.2.8g)

rule of thumb, given a node (j,p), p can be seen as the sequency of the node. Therefore, the procedure of reordering can be synthesized as:

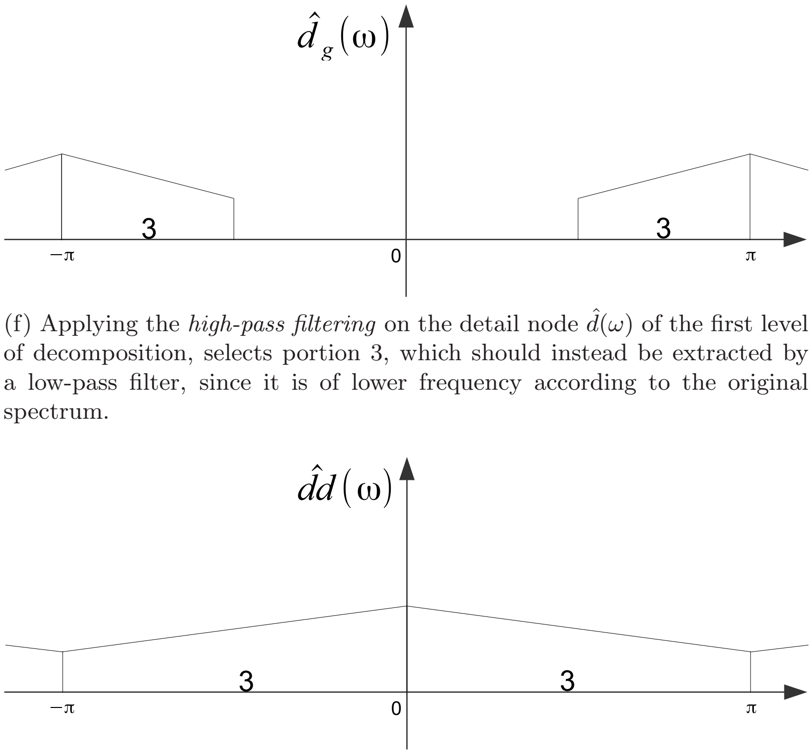
$$\text{if } p \text{ is } \begin{cases} even & \text{then low-pass on the left, high-pass on the right} \\ odd & \text{then high-pass on the left, low-pass on the right} \end{cases} \tag{2.62}$$

This can either be done *while* calculating the WPT in order to optimize the computational time or after computing the Paley ordered tree, if that form could possibly be of any interest, too.

2.4.6 The Best Basis Decomposition

As mentioned in Sec.2.4.3, once performed a full WPD, the choice of the basis is purely discretionary. By choosing a particular criterion, anyway, the choice of the basis can be optimized for selecting the nodes that best represent the signal according to such a criterion. In this way, a ***best basis decomposition*** is performed to highlight certain characteristics of the signal.

The instrument for the choice is a cost function that maps each node to a real value calculated from the elements $\mathbf{x}$ of the node itself by a specific law. Such map functions are defined as [32]:



(f) Applying the *high-pass filtering* on the detail node $\hat{d}(\omega)$ of the first level of decomposition, selects portion 3, which should instead be extracted by a low-pass filter, since it is of lower frequency according to the original spectrum.

(g) *Down-sampling* $\hat{d}_g(\omega)$ gives the node containing $\widehat{ad}(\omega)$.

Figure 2.8: The effect of aliasing is an inversion in the spectral sequentiality due to high-pass filtering and down-sampling. Portion 3 of the original spectrum of $\hat{f}(\omega)$ spans a lower-frequency range of the ω-axis than portion 4 does. Nonetheless, passing $\hat{f}(\omega)$ through filtering and down-sampling up to the second level of decomposition, portion 3 gets extracted by the high-pass filter, so appearing to be of higher frequency, instead. This is why the "natural" order of the WPD must be adjusted.

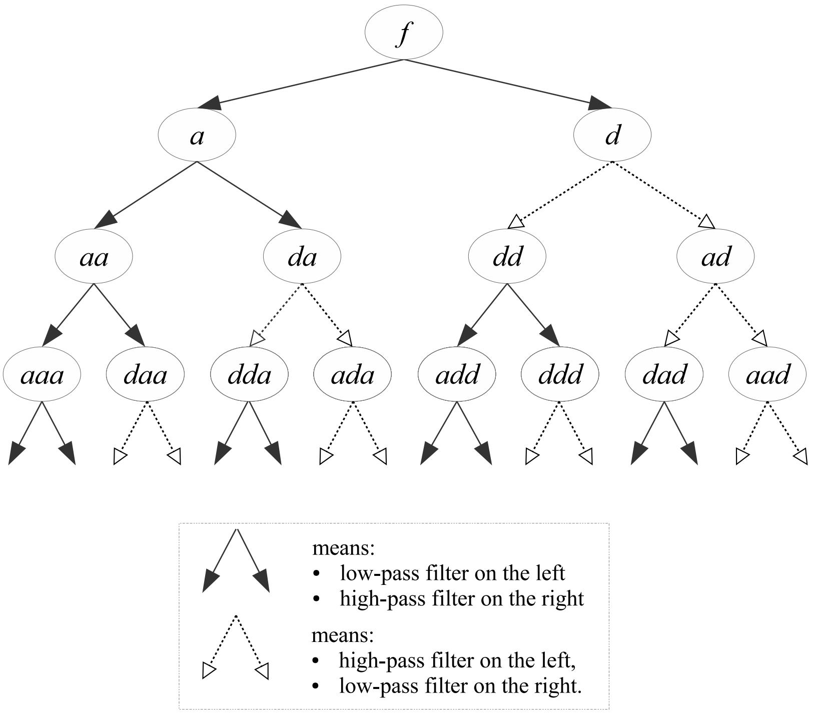

f
a
d
aa
da
dd
ad
aaa
daa
dda
ada
add
ddd
dad
aad
means:
• low-pass filter on the left
• high-pass filter on the right
means:
• high-pass filter on the left,
• low-pass filter on the right.

Figure 2.9: The children of the odd-sequency nodes must be swapped with respect to the natural order of the Wavelet Packet tree, due to the effects of aliasing. The left child, then, will be the high-pass-filtered and down-sampled version of their own coefficients, whereas the right child will be the low-passed-filtered and down-sampled version, inversely with respect to Paley ordering.

Definition 2.4.3 *A map $\mathcal{M}$ from sequences $\mathbf{x} = \{x_n\}$ to $\mathbb{R}$[15], is called an additive information cost function if:*

$$\begin{cases} \mathcal{M}(0) & = 0 \\ \mathcal{M}(\mathbf{x}) & = \sum_n \mathcal{M}(x_n) \end{cases} \tag{2.63}$$

The typical functions used to find out the orthogonal basis that best describes a particular aspect of a signal $f(n)$ are *entropy* functions [18][32]:

Shannon Entropy The (non-normalized[16]) Shannon Entropy for a single

[15] In the WPT, $n \in \mathbb{N} : n = 0, \ldots, (\lambda_j - 1)$, where $\lambda_j = \frac{M}{2^j}$ is the length of the sequence of a node at level j.

[16] The normalized Shannon-Weaver Entropy for a single element is defined as $\mathcal{M}_{shan-weav}(x_n) = \rho_n^2 \log \rho_n^2$, where $\rho = \frac{|x_n|^2}{\|\mathbf{x}\|^2}$, and, thus, for a sequence, it is $\mathcal{M}_{shan-weav}(\mathbf{x}) = -\sum_n \rho_n^2 \log \rho_n^2$. Even if it is set that $0 \log 0 = 0$, this is a non additive cost function. That is why the non-normalized version is necessary.

element is defined as:

$$\mathcal{M}_{shannon}(x_n) = x_n^2 \log x_n^2 \tag{2.64}$$

so that, set $0 \log 0 = 0$, for a sequence $\mathbf{x}$ it is:

$$\mathcal{M}_{shannon}(\mathbf{x}) = -\sum_n x_n^2 \log x_n^2 \tag{2.65}$$

As a matter of fact, $\exp\left(\mathcal{M}_{shannon}(\mathbf{x})\right)$ is related to the number of coefficients needed to represent the signal.

Logarithm of energy The so-called "log energy" entropy for a single element is defined as:

$$\mathcal{M}_{log\,en}(x_n) = \log x_n^2 \tag{2.66}$$

so that, set $\log 0 = 0$, for a sequence $\mathbf{x}$ it is[17]:

$$\mathcal{M}_{log\,en}(\mathbf{x}) = -\sum_n \log x_n^2 \tag{2.67}$$

Concentration in l^p norm The concentration in l^p norm entropy, chosen $p \in \mathbb{R} : p \in [1, 2[$, for a single element is defined as:

$$\mathcal{M}_{norm}(x_n) = |x_n|^p \tag{2.68}$$

so that, for a sequence $\mathbf{x}$ it is:

$$\mathcal{M}_{norm}(\mathbf{x}) = \sum_n |x_n|^p = \|\mathbf{x}\|_p \tag{2.69}$$

The smaller is the l^p norm of a function of unit energy, the more concentrated it is into a few coefficients.

Number above a threshold The threshold entropy, chosen an arbitrary threshold $\epsilon \in \mathbb{R} : \epsilon > 0$, for a single element is defined as:

$$\mathcal{M}_{thresh}(x_n) = 1 - \Theta(x_n, \epsilon) = \begin{cases} 1 & \text{if } |x_n| > \epsilon \\ 0 & \text{if } |x_n| \leq \epsilon \end{cases} \tag{2.70}$$

where the auxiliary step function $\Theta(x_n, \epsilon)$ is similar to Eq.2.52. In this way, for a sequence $\mathbf{x}$ it is:

$$\mathcal{M}_{thresh}(\mathbf{x}) = \sum_n \left(1 - \Theta(x_n, \epsilon)\right) \tag{2.71}$$

This entropy gives the number of coefficients needed to transmit the signal with precision ϵ.

[17] Minimizing this function finds the best apporoximation of the Karhunen Loeve basis for the process, which attains the global minimum for $\mathcal{M}_{log\,en}$ over the whole orthogonal group [32].

Sure entropy The sure entropy [18], chosen an arbitrary threshold $\epsilon \in \mathbb{R} : \epsilon > 0$ and introduced the same auxiliary function $\Theta(x_n, \epsilon)$ of Eq.2.70, for a sequence $\mathbf{x}$ of length $\lambda \in \mathbb{N}_0$ is defined as:

$$\mathcal{M}_{sure}(\mathbf{x}) = \lambda - \sum_n \Theta(x_n, \epsilon) + \sum_n \min(x_n^2, \epsilon^2) \tag{2.72}$$

Chosen one of this functions, it can be used all through the *sequency ordered WP Tree* to compute the entropy of each node. Then, by the ***Best Basis Algorithm***, some nodes are *marked.* Then the closest to the root constituting the Best Basis are extracted in order to ***minimize*** the total entropy. The steps of this procedure are shown in Alg.2.4.1 [32].

Algorithm 2.4.1 *The Best Basis Algorithm states that whenever a parent is of lower information cost than the children, the parent is marked. If the children have lower information cost, the parent is not marked, instead the (lower) total information cost of the children is assigned to the parent.*

Once marked the nodes[18], starting from the terminals $(l, 0)$ and $(l, 1)$ and their parent, a Depth First Search is performed to extract the marked nodes closest to the root from left to right, which form the Best Basis.

Since the tree has been sequency ordered, the concatenation of the sub-bands spanned by each node leads to no corruption of the sequentiality of the spectrum. Eventually, the signal is described with a different resolution in time and frequency (*i.e.* different Heisenberg boxes in the Phase Plane), according to the level of decomposition of each node selected.

The number of chosen nodes is not predictable and varies from signal to signal. Therefore, such Best Basis procedure can be fruitfully exploited for analysis purposes, on one hand, but, on the other hand, it is unlikely to use it in a classification strategy: in such a case, in fact, a fixed base is preferable in order to suitably compare some signals.

[18] The marking process can be done at the same time as the coefficients in the nodes are computed, as noted in [32].

Chapter 3

The Classification

The WPT is able to provide a wide range of possible ways of looking at the information contained in a signal both in time and frequency. It can be considered one of the most powerful tools from the MRA. Anyway, there could be little point in just obtaining the coefficients of each node: some elaboration of the data should be performed to create an efficient instrument of analysis and classification. In fact, there are a few particularly useful approaches to treat the decomposed signals on the nodes of the tree: the Wavelet Packet Spectrum (WPS) and the Energy Map.

The purpose of the classification is to extract the most reduced *feature set* of data that allows the recognition of a pattern with an acceptable percentage of error. In the case of the WP Tree, the features are the energies of some particularly relevant nodes.

3.1 Tools for Signal Analysis from the WPT

The WPS is the most handy and flexible tool to analyze a signal spectrum along the time axis: basically, it lets visualize the amplitude of the coefficients of some selected nodes, properly coloring the Heiseberg Boxes they represent of the Phase Plane (supposing a sequency ordered WP Tree). The Energy Map, instead, gives a focus, level by level, on the concentration of energy in the spectrum of the signal. It is particularly useful for extracting a reduced set of representative information for classification purposes.

3.1.1 The Wavelet Packet Spectrum

The ***Wavelet Packet Spectrum*** (WPS) is the representation of the information contained in each node on the Phase Plane, with time t on the horizontal axis and frequency f on the vertical axis. This plane is partitioned in rectangular patches of size Δt-by-Δf, called Heisenberg boxes in honor of the uncertainty principle [32], as anticipated since Sec.2.1.4. These boxes are disposed according

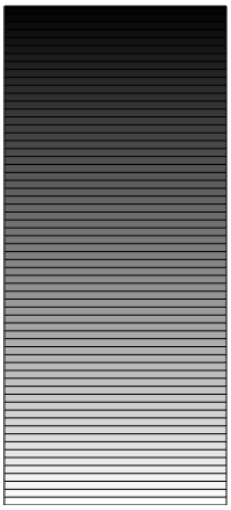


Figure 3.1: This is an example of color palette with 64 shades. In particular, this is a so-called flipped gray palette, associating the highest value to black and the lowest to white.

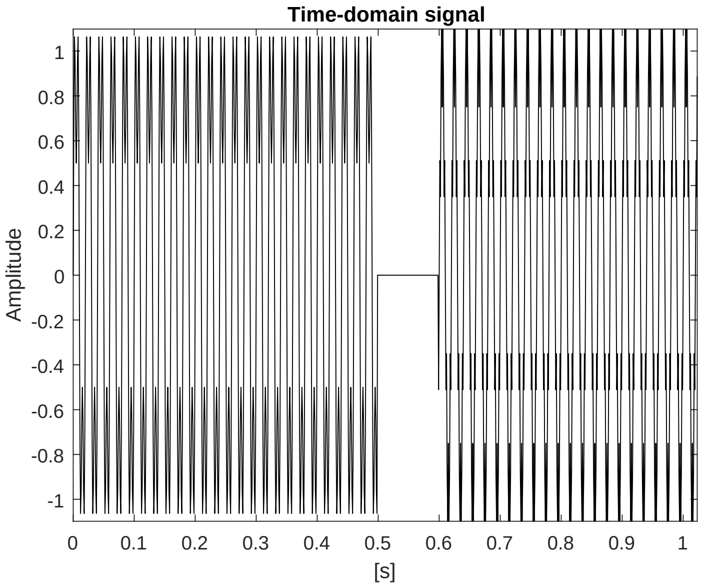
to a grid where the rows represent the nodes of the basis, so that the leftmost node on the tree at a specified level corresponds to the lowest row (low frequency) and the rightmost node is the top row (high frequency).

The information considered is in the form of the amplitude of the coefficients $|x_n|$ of the nodes of the chosen basis. The *color palette* used to assign the corresponding tonality to the amplitude of a coefficient is built so that the minimum value assumed by the coefficients of the nodes corresponds to the bottom of such a palette and the maximum value corresponds to the top of it. Between the intermediate shades and amplitudes there is a quasi-linear relation, except for some quantization in the color-map. An example can be seen in Fig.3.1.

The most direct way of using the WPS is to select all and only the nodes of a certain level j of decomposition of a sequency ordered WP Tree. In such a way, an orthogonal basis is taken into account and there is a uniform partition of the Phase Plane: the number of coefficients per node ($\lambda = M/2^j$), the time resolution ($\Delta t = \Delta T/2^j$) and span ($\Delta\omega = \pi/2^j$ or equivalently $\Delta f = f_s/2^{j+1}$) of every chosen node are the same, since the level is identical. Therefore, the WPS is made of a grid of $2^j \times \lambda$ rectangular patches all of the same size Δt-by-Δf.

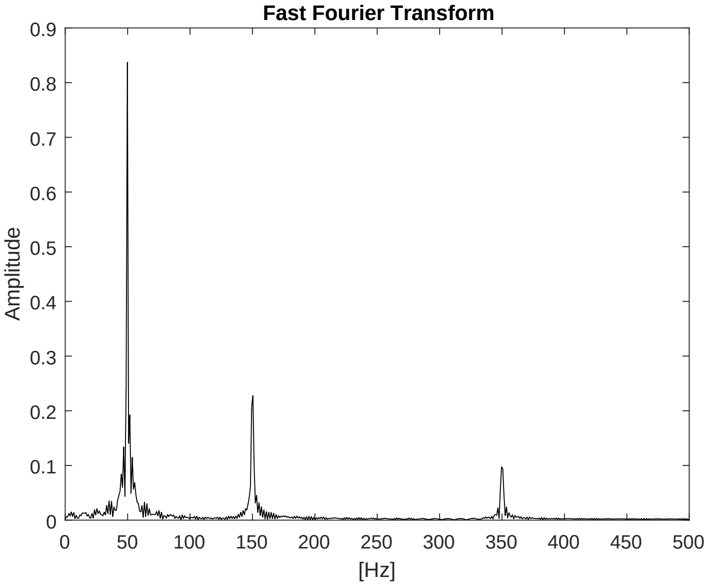
If another basis is selected, including nodes from several levels, the size of the boxes of each row generally differs from that of another one. For instance, the Alg.2.4.1 generally leads to this latter kind of basis, leading to non-uniform boxes. An example is shown in Fig.3.2.

Though the WPD deals with a huge amount of scattered data and, therefore, is not suitable to a fast real-time method of classification, it is essential for a



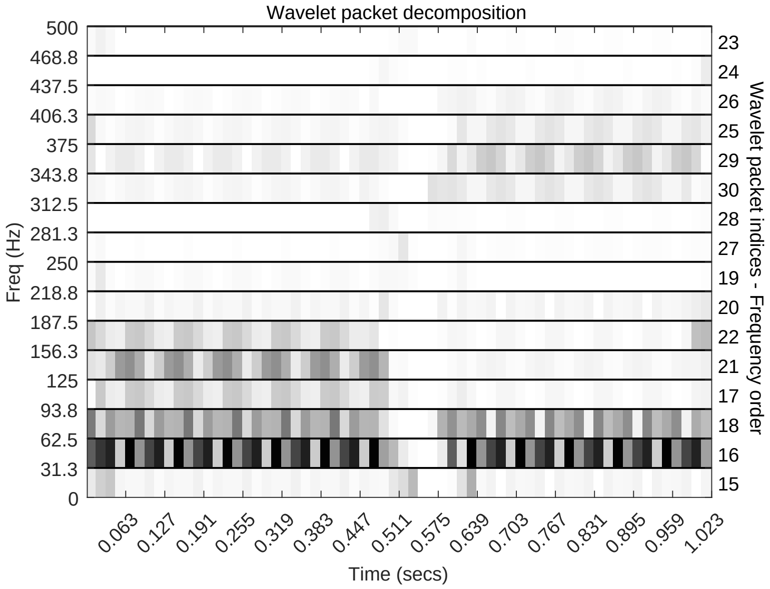
(a) Example of a non stationary signal made of a sinusoid at 50 Hz, with a second harmonic superposed at 150 Hz (a half of the amplitude) in the first trait and another harmonic at 350 Hz (a quarter of the amplitude) added in the second trait. In the middle of the acquisition there is a section of constant null signal separating the two traits. The sampling frequency is 1 kHz. The number of samples is $N = M = 1024 = 2^{10}$, thus covering an interval $\Delta T = 1.024\,\mathrm{s}$.

Figure 3.2

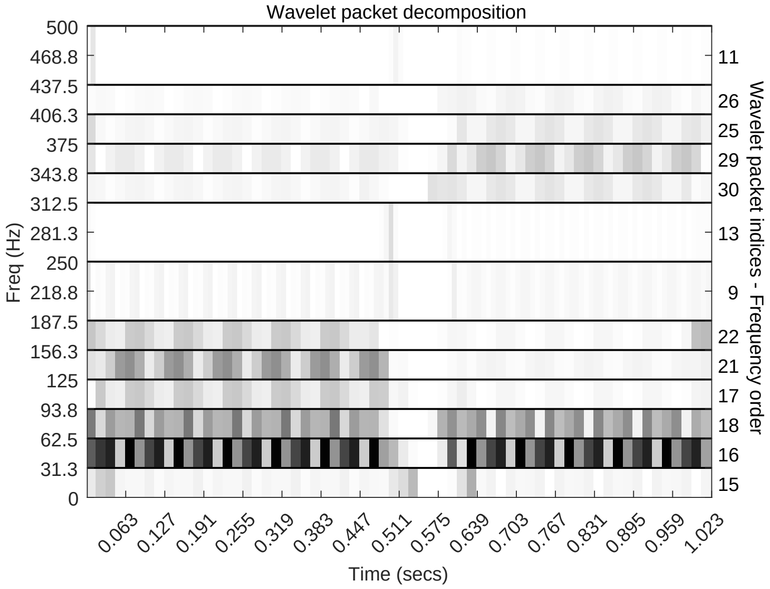


(b) The Fast Fourier Transform performed by MATLAB highlights the presence of the three harmonics, but does not give any information about their duration and the instant when they appear. Besides, this fact distorts the real amplitude of those component: the spectral component in the FFT is then proportional to the real amplitude of the signal component, but also to its duration. Analyzing non stationary signals with this tool is not wise.

Figure 3.2



(c) WPS of the signal in Fig.3.2a with a level of decomposition 4. Here only the equal-width sub-bands of the fourth level are chosen.



(d) WPS of the signal in Fig.3.2a with a level of decomposition 4 using the Best Basis from Alg.2.4.1 with the non-normalized Shannon Entropy (Eq.2.65). The sub-bands spanned by the nodes selected are not equal.

Figure 3.2: Example of a non-stationary signal Fig.3.2a decomposed to a level $l = 4$ by the WPT, whose WPS is shown using an equal-width sub-band representation in Fig.3.2c and a Best Basis Tree selection in Fig.3.2d.

preliminary analysis of the kind of signals that are to be taken into account. For instance, it could be useful to know whether a spectral component is always present during the acquisition of a generic non-stationary signal, or if it appears only during some transients. This could be extremely hard or ambiguous to spot in the standard basis (*i.e.* the time domain) and surely cannot be done by the FT, which gives a global view on the whole ΔT. Only the STFT could give some hints, but with all the limitations cited in Sec.2.1.2.

3.1.2 The Energy Map

The ***Energy Map*** [10] of a WPD is a grid containing the average energy e_m of every node vector of coefficients $\mathbf{x}_m = \{x_{m,0}, x_{m,1}, \dots, x_{m,(\lambda_j-1)}\}$ of level j, serial number m (see Sec.2.4.1) and length $\lambda_j = 2^\eta$ (*cf.* Eq.2.48), so that:

$$e_m = \frac{1}{\lambda_j} \sum_{n=0}^{\lambda_j - 1} [x_{m,n}]^2 \tag{3.1}$$

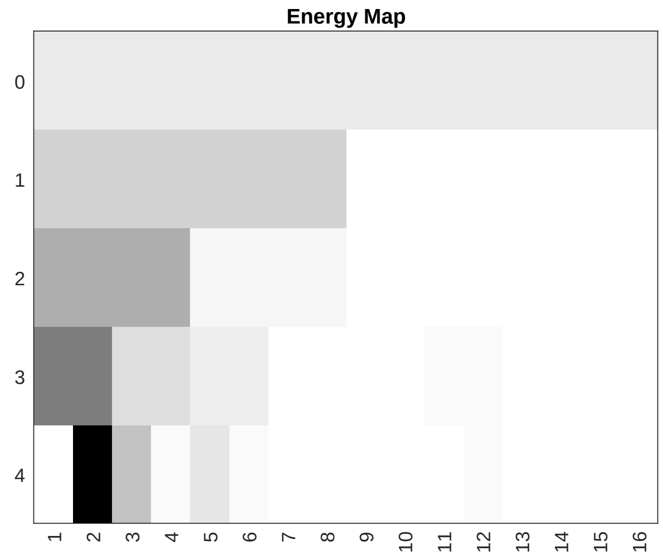
The choice of the *average energy* as the representative parameter is due to the intention of spotting the parts of the spectrum containing the highest concentration of information during the time interval of observation ΔT.

Each row of the Energy Map represents a level of decomposition and each box represents a node. On a row, then, there are 2^j boxes, where the original signal is located at $j = 0$, represented by the top row. The order in which the energies of the nodes are presented is the same of the sequency-ordered WP Tree, as a convention.

The main advantage of this mode of visualization of the signal decomposition is the reduction of the terms that describe the signal. For instance, the acquired signal can have a large amount of samples N. Its adapted version (in order to fit the requirements of the Wavelet Transform, see Sec.2.3.7) has a number of samples M which is the power-of-two number closest to N. The information is scattered all through the WP Tree and possibly involves a lot data. The Energy Map, instead, has as many terms as the total number of nodes in the tree, that is $P_l = \sum_{j=0}^{l} 2^j$. Therefore, if the chosen decomposition level is not too high, P is "small"[1]. The major reasons why the level of decomposition l is limited are the computational time and memory needed for the WPT and also the progressive larger uncertainty in time.

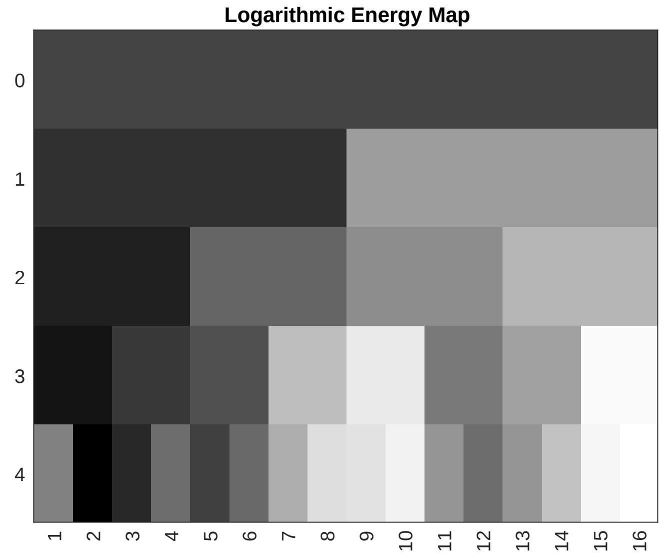
The computation of the average energy of a whole node can encounter some difficulties in the repeatability of the measurement, since it is subjected to a strong dependence on the trigger instant: if the trigger is not properly chosen, a time shift between two measures of the same phenomenon causes a discrepancy in the distributions of the coefficients of the nodes between their trees. To stem

[1] Common values for l can be in the interval between 4 and 8 (thus leading at a maximum of $P_8 = \sum_{j=0}^{8} 2^j = 511$ values of energy).



(a) This is the Energy Map of the signal in Fig.3.2a up the fourth level of decomposition. The darkest boxes represent the nodes with the highest energy. On the top row there is only the average energy of the node $(0, 0)$ (level 0 of the WP Tree), whereas going down to the bottom row the average energies of the nodes of the various levels appear.

Figure 3.3



(a) This is the same Energy Map of Fig.3.3a, but seen in a logarithmic scale log. This point of view may let some possibly relevant nodes emerge, since in the linear scale the different orders of magnitude among the energies could have hidden them (*i.e.* they may have remained white in the case of this flipped gray color-map). This is particularly useful when comparing several Energy Maps to try to spot some changes among possible different classes.

Figure 3.4:

this incompatibility, it is possible to use a window [10] in order to calculate the average energy on several portions of the coefficient vectors of the nodes, so that the dependency on the trigger instant is diminished, being the energy distributed among many near elements.

Let W_j be the window length for level j, with $W_{j+1} = \frac{1}{2}W_j$ and $i \in \mathbb{N} : i = 0, \dots, \left(\frac{\lambda_j}{W_j} - 1\right)$ the generic index of the resulting energies for a node with serial number m. The length of the window W_j should be a power-of-two fraction of the node vector length, so that an integer number of windows per node $\frac{\lambda_j}{W_j} \geq 1$ is applied. The i-th energy $e^w_{m,i}$ calculated on the node m, using a window which extracts a fraction $\frac{W_j}{\lambda_j}$ of the samples of a sequence on level j, would then be:

$$e_{m,i} = \frac{1}{W_j} \sum_{n=i \cdot W_j}^{i \cdot W_j + W_j - 1} [x_{m,n}]^2 \tag{3.2}$$

In Eq.3.2 it is supposed that the windows are applied in series through the coefficients of a node. Nevertheless, it is also possible to employ them with overlapping. Set $\gamma \in \mathbb{Q} : \gamma > 1$ as the inverse of the dimensionless fraction of W_j giving the distance between two consecutive windows (*e.g.* if there is a shift of $\frac{1}{2}W_j$ between two consecutive overlapping windows, then $\gamma = 2$), accordingly modifying Eq.3.2 gives that:

$$e^{1/\gamma}_{m,i} = \frac{1}{W_j} \sum_{n=i \cdot W_j/\gamma}^{i \cdot W_j/\gamma + W_j - 1} [x_{m,n}]^2 \tag{3.3}$$

Here, $i \in \mathbb{N} : i = 0, \dots, \gamma\left(\frac{\lambda_j}{W_j} - 1\right)$.

Nevertheless, even if computing such kinds of windowed energies can possibly loosen the dependency on time shifts, a sensible proliferation of the terms that are used to describe the signal is caused. In fact, applying adjacent windows like in Eq.3.2 provokes an increase of the Energy Map size from $P_l = \sum_{j=0}^{l} 2^j$ to $P'_l = P_l \frac{\lambda_j}{W_j}$. Employing overlapping windows even leads to a size $P''_l = P_l\left(\gamma\frac{\lambda_j}{W_j} - (\gamma - 1)\right)$, where $P_l < P'_l < P''_L$.

A proliferation of the terms is counter-productive if the main purpose is to extract some concise information to feed the classification algorithm with. This is why a simple average on the whole node vector is preferred for the feature extraction (Sec.3.2) and the following classification algorithm (Sec.3.3). On the other end, greater effort must be made to trigger the signal as repeatably as possible.

3.2 Feature extraction

Let a single acquisition be taken into account for now. Let it be characterized by a sample rate f_s and a duration ΔT, thus giving $N = \Delta T \cdot f_s$ samples.

Performing the WPT with a level of decomposition l on an adapted signal of length M according to Sec.2.3.7 and, then, calculating the Energy Map of the sequency ordered WP Tree, returns a collection of scalar energies e_m, where $m \in \mathbb{N} : m = 1, \dots, P_l$ is the serial number of a node (according to Sec.2.4.1). For a matter of practicality, those energies can be put in a vector $\mathbf{e} = \{e_1, e_2, \dots, e_{P_l}\}$[2].

Let a whole campaign of K_1 acquisitions of the same phenomenon be considered. If the procedure of decomposition and extraction of the Energy Map is performed with the same parameters, an energy vector $\mathbf{e_{k_1}}$ can be computed for every acquisition, where $k \in \mathbb{N} : k_1 = 1, \dots, K_1$. Knowing *a priori* that these measurement are about the same type of event, a ***class*** C_1 can be assigned to this set of energy vectors.

In such a set of vectors, there should be some particularly relevant coefficients in some fixed position[3] that seems able to characterize the class.

Taking further campaigns in a similar way highlights other classes C_c of finite length K_c (where c is the generic index of a class) and other possible sets of "characterizing" coefficients for these new classes. If the trigger was properly set, those coefficients should be in some fixed positions within their own energy vectors $\mathbf{e_{c,k_c}}$ (of class c and number of acquisition k_c within the campaign, in the following it will be $k = k_c$ for the sake of readability).

As a consequence, a reduction of the nodes (*i.e.* of the dimensionality of the energy vector) is prospected. A set of ***features*** can be built, as an example of a general case, from the union of the positions of the "characterizing" coefficients identified. Extracting the energies in such positions from all of the energy vectors generates a ***Train Set***, that is a stack of reduced-dimensionality classified energy vectors (considered as row vectors), so that:

$$Train\,Set = \begin{bmatrix} \mathbf{e_{1,1}} & C_1 \\ \mathbf{e_{1,2}} & C_1 \\ \vdots & \vdots \\ \mathbf{e_{c,k}} & C_c \\ \vdots & \vdots \end{bmatrix} \tag{3.4}$$

This Train Set is the knowledge on which the classification algorithm will decide the class of a new acquisition. Assuming that a brand new set of unclassified energy vectors, called ***Test Set***, is taken so that: (i) f_s and ΔT are the same

[2] Let it be recalled that $P_l = \sum_{j=0}^{l} 2^j$, *i.e.* each term of $\mathbf{e}$ is the average energy of a node of the frequency ordered WP Tree, where the root has $m = 1$ and the others are numbered left-to-right, from the lowest to the highest level of decomposition.

[3] If the trigger was properly set, the pattern of energy distribution along the nodes should not be shifted between the various acquisition of the same event.

of the acquisitions of the vectors of the Train Set (ii) its events can be classified like those of the Train Set., So, it is possible to assign each vector of the Test Set to one of the known classes of then Train Set, according to its "closeness". If κ is the generic index for an acquisition of the Test Set, then it can be seen as:

$$Test\,Set = \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \\ \vdots \\ \mathbf{e}_\kappa \\ \vdots \end{bmatrix} \tag{3.5}$$

3.3 Classification algorithm

Once provided a Train Set, it is possible to operate the classification of the vectors of the Test Set $\mathbf{e}_\kappa$ on the basis of the computation of the their *distance* from each classified vector $\mathbf{e_{c,k}}$ of the Train Set itself.

Taking into account that the final application of this algorithm is on a MCU, the computational burden of the classification must be restrained. That is one of the reasons why the algorithm chosen is the *Nearest Neighbor Rule* (*NN rule*, Sec.3.3.1): an unclassified ***feature vector*** can be associated to a class on the basis of the majority of vectors of a specific class among its nearest K ones (*i.e. K-NN rule*). A preeminent role is thus played by the chosen ***distance function*** (Sec.3.3.2).

3.3.1 Nearest Neighbor Algorithm

The *K Nearest Neighbor Rule* (***K-NN rule***) is a ***supervised learning algorithm*** that provides the classification of the feature vectors of a Test Set according to the prevalence of those of a specific class among the nearest $K \geq 1$ ones. The steps of the algorithm are listed in Alg.3.3.1:

Algorithm 3.3.1 *The stages of the K Nearest Neighbor Algorithm (K-NN) are:*

i) arranging a Train Set of classified feature vectors according to Eq.3.4;

ii) arranging the Test Set so that it is made of the feature vectors representing the same nodes of the WPT as the ones of the Train Set;

iii) computing the distances between each vector of the Test Set $\mathbf{e}_\kappa$ *and all the vectors of the Train Set* $\mathbf{e_{1,1}}, \ldots, \mathbf{e_{c,k}}, \ldots;$

iv) for each vector of the Test Set, searching the K closest vectors of the Train Set, i.e. *those with the smallest distance (see Sec.3.3.2);*

v) each vector of the Test Set is finally assigned to the class which the majority of its K nearest neighbors of stage iv) is associated to.

The simplest application of this algorithm consists in setting $K = 1$, *i.e.* the class of a vector of the Test Set is decided upon its closest vector of the Train Set. This implementation is summarized in Fig.3.5.

The determiner of the success of the algorithm is choice of the distance function: a simple euclidean distance might not be the cleverest option. The success of the algorithm must be tested choosing, as members of the Test Set, some feature vectors that are actually known to belong to some specific class of the Train Set. The percentage of error on this known entries will give reliable indications on the goodness of the application of the algorithm in the particular case.

3.3.2 Distance Functions

The distance functions that may be taken into account to quantify the nearness of two feature vector are several, each one with peculiar characteristics that may facilitate or not the success of the classification according to the specific case. In the following, a series of functions of distance is presented [6]. Though these ones are computed all along the experiments of Part II, the whole Sec.3.3.2.1 is dedicated to the χ^2 distance, considered the most reliable one.

To describe the different kinds of distance, let two sample units i and h of the same length p be considered as row vectors, where the j-th element of the sample unit i is described by a_{ij} and the j-th element of h by a_{hj}, where $j \in \mathbb{N} : j = 1, \ldots, p$ [4] [20]:

$$\begin{array}{rl} i: & [a_{i1} \quad \ldots \quad a_{ip}] \\ h: & [a_{h1} \quad \ldots \quad a_{hp}] \end{array} \tag{3.6}$$

According to this notation, the following distance functions d_{ih} are:

Euclidean The normalized Euclidean distance between two sample units i and h is:

$$d_{ih}^{eucl} = \sqrt{\frac{\sum_{j=1}^{p}(a_{ij} - a_{hj})^2}{p}} \tag{3.7}$$

Cosine similarity The cosine similarity between two sample units i and h is:

$$d_{ih}^{sim} = \frac{\mathbf{a_i} \cdot \mathbf{a_h}}{\|\mathbf{a_i}\| \, \|\mathbf{a_h}\|} \tag{3.8}$$

Minkowsky The Minkowsky similarity between two sample units i and h, chosen a value $r > 1$, is:

$$d_{ih}^{sim} = \left(\sum_{j=1}^{p} |a_{ij} - a_{hj}|^r \right)^{1/r} \tag{3.9}$$

[4] Here the numeration of the elements of the vectors starts from 1 rather than 0 merely for the sake of clarity utilizing the subscripts.

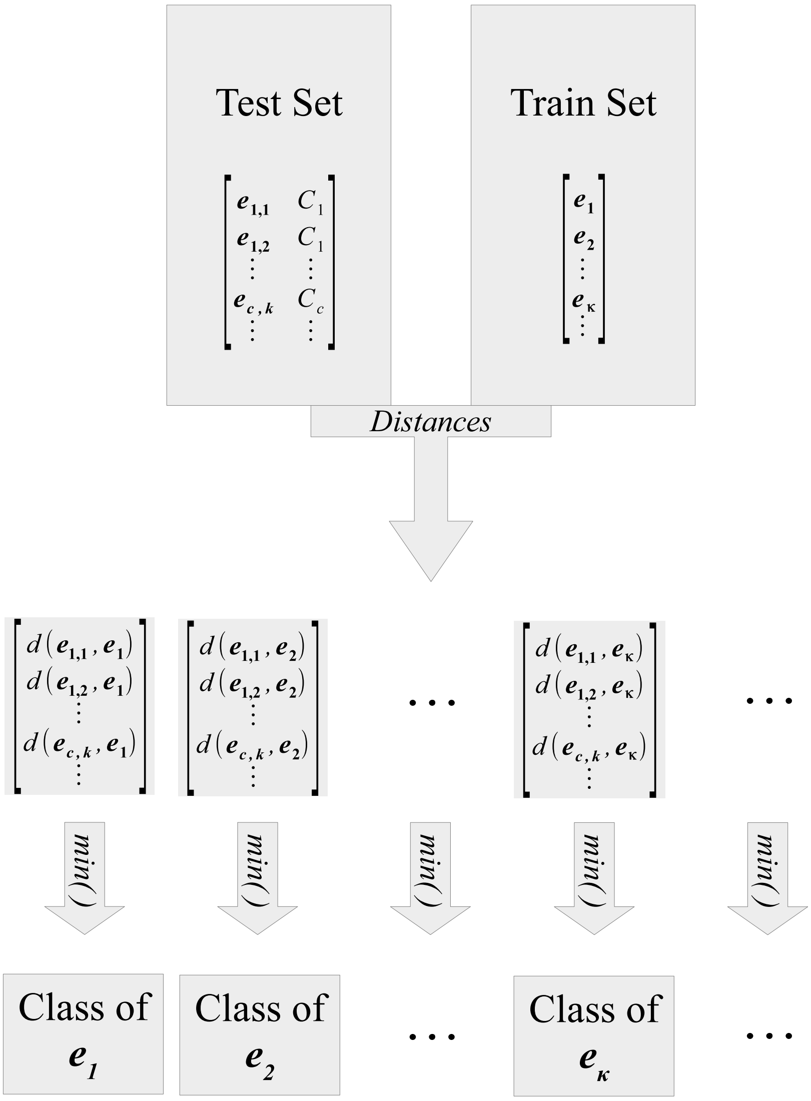

Test Set
$\begin{bmatrix} e_{1,1} & C_1 \\ e_{1,2} & C_1 \\ \vdots & \vdots \\ e_{c,k} & C_c \\ \vdots & \vdots \end{bmatrix}$
Train Set
$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_\kappa \\ \vdots \end{bmatrix}$
Distances
$\begin{bmatrix} d(e_{1,1}, e_1) \\ d(e_{1,2}, e_1) \\ \vdots \\ d(e_{c,k}, e_1) \\ \vdots \end{bmatrix}$
$\begin{bmatrix} d(e_{1,1}, e_2) \\ d(e_{1,2}, e_2) \\ \vdots \\ d(e_{c,k}, e_2) \\ \vdots \end{bmatrix}$
...
$\begin{bmatrix} d(e_{1,1}, e_\kappa) \\ d(e_{1,2}, e_\kappa) \\ \vdots \\ d(e_{c,k}, e_\kappa) \\ \vdots \end{bmatrix}$
...
min()
min()
min()
min()
min()
Class of e_1
Class of e_2
...
Class of e_κ
...

Figure 3.5: This is a schematic representation of the Nearest Neighbor Rule with a number of closest vectors $K = 1$ to choose from. To sum up, it consists of computing all the distances from each feature vector of the Test Set to all of the vectors of the Train Set and, then, finding the minimum distance for each case. The class of the vector of the Train Set having the minimum distance from a particular test vector will also be the class of this latter one.

Correlation The correlation between two sample units i and h, computed the averages $\mu_i = \frac{1}{p}\sum_{j=1}^{p} a_{ij}$ and $\mu_h = \frac{1}{p}\sum_{j=1}^{p} a_{hj}$, is:

$$d_{ih}^{corr} = \frac{\sum_{j=1}^{p}(a_{ij} - \mu_i)(a_{hj} - \mu_h)}{\sqrt{\sum_{j=1}^{p}(a_{ij} - \mu_i)^2 \sum_{j=1}^{p}(a_{hj} - \mu_h)^2}} \tag{3.10}$$

3.3.2.1 Chi-Square Distance (χ^2)

Maintaining the conventions of Sec.3.3.2, some preliminary definitions need to be made before properly giving the formulation of Chi-Square Distance between two sample units i and h (χ^2_{ih}).

First of all, let the Train Set be a matrix of finite size $q \times p$, where the rows are the sample units (*i.e.* the feature vectors) and the columns are the species (*i.e.* the values computed for each feature from the various acquisitions) [20]. Then, let the total for sample unit i ($a_{i+} \in \mathbb{R}$) and the total for sample unit h ($a_{h+} \in \mathbb{R}$) be:

$$a_{i+} = \sum_{j=1}^{p} a_{ij} \tag{3.11}$$

$$a_{h+} = \sum_{j=1}^{p} a_{hj} \tag{3.12}$$

Let also the total for species j along the training set be $a_{+j} \in \mathbb{R}$:

$$a_{+j} = \sum_{k=1}^{q} a_{kj} \tag{3.13}$$

The χ^2 distance between two sample units i and h can then be defined as:

$$\chi^2_{ih} = \sqrt{\sum_{j=1}^{p} \frac{1}{a_{+j}} \left(\frac{a_{hj}}{a_{h+}} - \frac{a_{ij}}{a_{i+}} \right)^2} \tag{3.14}$$

This distance is similar to the basic Euclidean one in Eq.3.7, but it is weighted by the inverse of the species total and each values of a sample unit is weighted by its own total. Such a fact uniforms the orders of magnitude, letting differences on very different scales be comparable and with similar importance in the final computation of the distance.

According to [6], this kind of distance has proven to be the most reliable one in the classification of either numerical, categorical and mixed datasets. In addition to the satisfying percentage of success of a few experiments conducted, that is a valid reason why the results from the χ^2 distance are considered the most significant.

Part II

Experimental results

Chapter 4

The experimental setup

During the one-year-long time of this work, various setups had been taken into account. In the next sections some steps of the evolution of the setup are presented to show how to apply the previous concepts on home appliances. The presumption of enforceability of the outcome is obtained starting from a wide-range spectral analysis using high-end equipment and then narrowing the range and lowering the budget up to the utilization of the cheap Allegro's ACS718 on-board current sensor.

All through the work, the system considered was not the complete refrigerator as the ones that can be seen in stores, but rather a much simplified version of it. Especially in the first phase, the mere couple of the inverter and the compressor was considered as the system. With increasing knowledge of the phenomena involved, then, also the motherboard on which the current sensor was welded was considered as part of the system.

Therefore, in Sec.4.1 the principal parts of the system will be described, so that the evolution of the setup will be shown in Sec.4.2.

4.1 The system

Starting from the notions of the refrigeration cycle, the inverters and the compressors will be generally described, together with the instruments to manage and control them.

4.1.1 The refrigeration cycle

The refrigeration cycle is the mathematical model for the functioning of the refrigerators. What the machine performing such a cycle does is extracting heat from a cooler heat source to move it to a warmer heat sink. Declined in the case of refrigerators, this cycles accomplishes the extraction of heat from the cool inside where the food is stored, to expel it to the warmer room-temperature environment. Since, for the second law of thermodynamics,

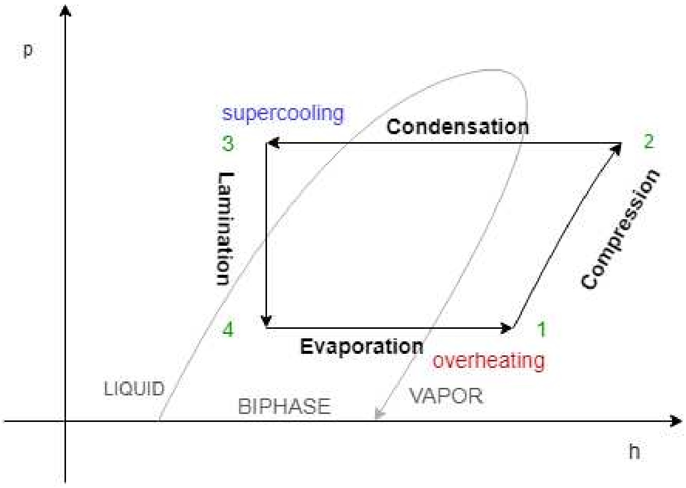


Figure 4.1: The forceful variations in pressure and phase of the working fluid allow the transportation of heat from the inside of the machine towards the outside. In this p-h plane, the cycle assumes a rather simple shape.

heat cannot spontaneously flow from a location at a lower temperature to another at a higher one, some mechanical work must be made to achieve the refrigeration cycle. Such a mechanical work, in the case of refrigerators, is given by a compressor. In a thermodynamic circuit, such compressor thrusts a working fluid through a condenser, a lamination valve and an evaporator (Fig.4.1).

In the case of the specific refrigerators under investigation, the working fluid or “coolant” is a hydrocarbon called isobutane R600a.

The refrigeration cycle can be represented on a pressure-vs-enthalpy plane, like in Fig.4.2. With reference to this figure, the steps are:

1-2 the compressor sucks the fluid in a superheated steam state, taking it from a low suction pressure to a high discharge pressure;

2-3 the fluid passes through the condenser and releases heat to the outside, becoming at first liquid (following an isotherm) and, then, subcooled liquid, while keeping the same pressure;

3-4 the subcooled liquid is expanded by a lamination valve that brings the fluid back to the low suction pressure so that it partially vaporizes;

4-1 the biphase fluid flows through the evaporator and receives heat from the outside, becoming at first steam (following an isotherm) and then superheated steam, while keeping the same pressure (the cycle starts again from here).

Now that the components of the cycle are identified, the focus will be put on the compressor and its role as provider of mechanical work: in Sec.4.1.2 a general

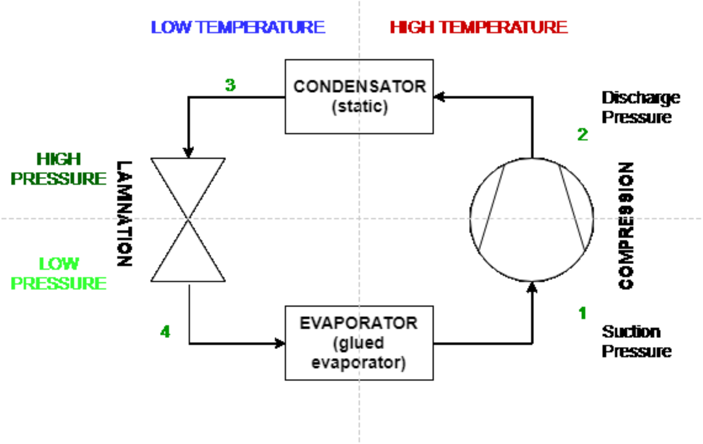


Figure 4.2: The components necessary to the refrigeration cycle are visible here: compressor, condenser, valve and evaporator. Temperature and pressure levels are highlighted in the proper area of the circuit.

overview on the compressors and the way they are supplied and controlled will be given.

4.1.2 Compressors and inverters

The energy to let the compressor do its mechanical work is provided by an electric power supply, usually the grid (230 V). Actually, the compressor is an electric motor moving a piston, which pushes the fluid in the circuit. The electric energy is not plugged directly into the compressor: instead, it is regulated by an inverter, *i.e.* a power converter that will provide the compressor with a three-phase voltage able to move the internal rotor according to the control logic, thus providing the proper mechanical work. As it can be seen in Fig.4.3, the power source is plugged into the inverter and a low-voltage square-wave signal is used as the input to the modulation of the three-phase voltage of the stator.

The main advantage of modern inverter refrigerators on the old on-off ones is that a smooth regulation can now be performed, with a dramatic increase in efficiency. For the sake of simplicity, the characteristic bounding the frequency of the 12 V Pulsed Width Modulation (PWM) control signal and the speed of the rotor has been constructed to be linear (Fig.4.4). The PWM signal can be sent either by the motherboard (in the ordinary operation), or by a wave generator in the laboratory. The modulation is provided by a symmetric bus-clamped Space Vector Modulation (SVM).

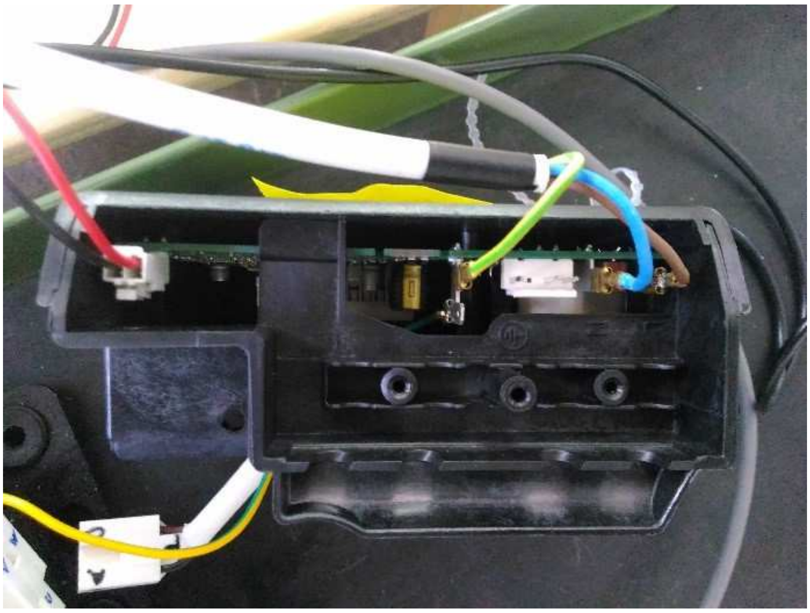


Figure 4.3: On the top-left corner, the black and red wires of the frequency-control signal can be seen. On the top-right corner, the single-phase power supply is plugged, while, on the bottom-left corner, the connector for the three-phase supply of the compressor is ready to be plugged.

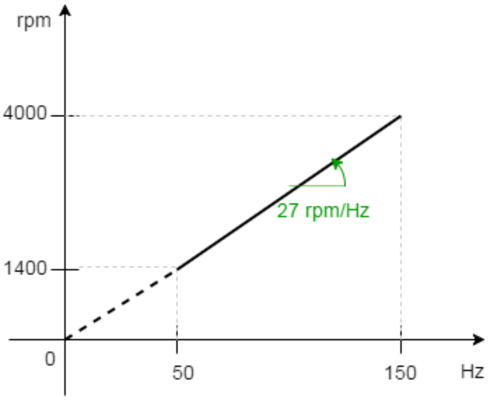


Figure 4.4: The characteristic between speed and frequency of the 12 V PWM is designed to be linear for the most part of couples inverter-compressor. In this example, the minimum speed of 1400 rpm corresponds to a PWM at 50 Hz, while the maximum speed of 4000 rpm to a PWM at 150 Hz. The constant slope is therefore of 27 rpm/Hz. This is the case of all the couples inverter-compressors examined.

4.1.3 The probes and the oscilloscope

The high-end equipment consisted of:

- Teledyne LeCroy Motor Drive Analyzer (MDA810A), 2.5 GS/s, 12 bit, 1 GHz (also referred to Digital Storage Oscilloscope or DSO);
- Teledyne LeCroy High-Voltage Differential Probe HDV3206, 2000 V, 120 MHz;
- Teledyne LeCroy Current Probe CP030A, 30 A, 50 MHz;
- EMtest NetWave Test Supply (for certified tests).

The elevated accuracies, sampling rates and bandwidths of these tools allow much more in-depth analyses then those that can be performed with the Allegro Current Sensor ACS718-10B (ACS718). The purpose of using them beforehand was to observe the system on the widest spectrum possible before acting on the motherboard. Gradually narrowing the spectrum taken into account, we made sure to still see what we needed.

4.1.4 The Microcontroller Unit and the Motherboard

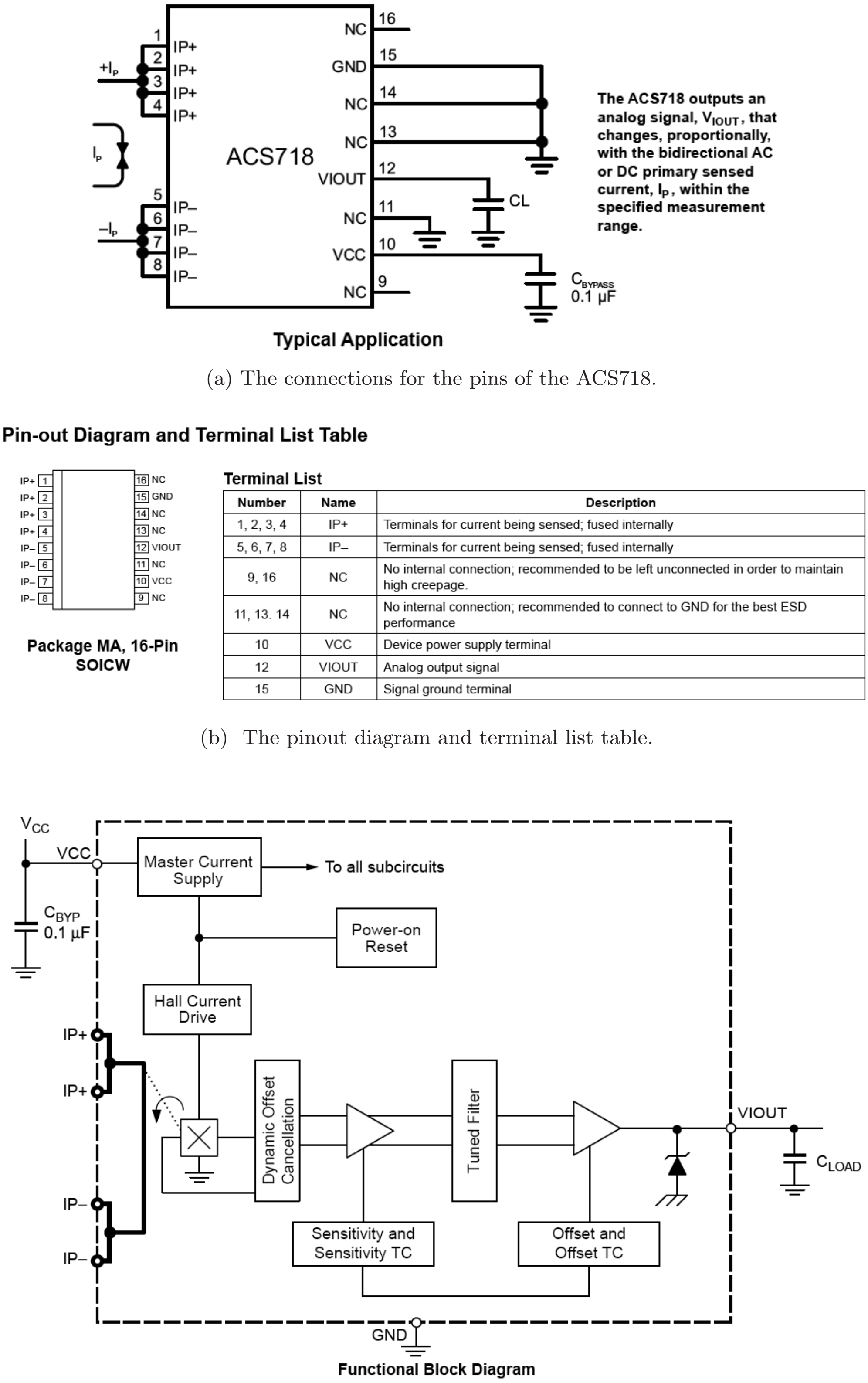
The MCU considered is the Kinetis MKE04Z128VLK4 ARM Cortex-M0+[23], which is mounted on the new generation of refrigerators' motherboards. It is a 32-bit MCU core with a 48 MHz CPU frequency, with memories of up to 64128 KB of flash memory and 16 KB of SRAM. For the purposes of this work, other interesting characteristics of this controller are those of its ADC: a 12-bit converter with up to 16 channels.

The motherboard on which the the MCU is inserted is the ERF2600 by Electrolux, designed for the control of the whole refrigerator system.

4.1.5 The Allegro's ACS718 Current Sensor

The High Isolation Linear Current Sensor IC with 850 $\mu\Omega$ Current Conductor[1] is a low-offset linear Hall-Effect sensor providing an output voltage proportional to the current that flows through a proper copper path located near the surface of the die. The connections of this chip can be seen in Fig.4.4.

The working principle is that the current flowing on the copper path (getting into pins 1—4 and out from 5—8) generates a magnetic field which is sensed by a Hall Integrated Circuit (IC), which in turn provides a proportional output voltage. This version of the ACS718 (10B) has a current sensing range of ±10 A and a sensitivity of 200 mV/A, with a zero-current offset equal to half of the continuous supply voltage V_{CC}, *i.e.* $0.5 \times V_{CC} = 2.5\,V$. The zero-current offset is due to the fact that the measure of current is bidirectional. So, given these specifications, the reliable range of output voltage is between 0.5 and 4.5 V,



The ACS718 outputs an analog signal, V_{IOUT}, that changes, proportionally, with the bidirectional AC or DC primary sensed current, I_P, within the specified measurement range.

(a) The connections for the pins of the ACS718.

Terminal List

Number	Name	Description
1, 2, 3, 4	IP+	Terminals for current being sensed; fused internally
5, 6, 7, 8	IP–	Terminals for current being sensed; fused internally
9, 16	NC	No internal connection; recommended to be left unconnected in order to maintain high creepage.
11, 13. 14	NC	No internal connection; recommended to connect to GND for the best ESD performance
10	VCC	Device power supply terminal
12	VIOUT	Analog output signal
15	GND	Signal ground terminal

(b) The pinout diagram and terminal list table.

(c) The functional block diagram.

Figure 4.4: Description of the pins of ACS718[1].

spanning the range ±10 A. The noise density is low, with a peak of $70\,mA_{rms}$ at the maximum bandwidth of 40 kHz.

4.2 The construction of the setup

For the initial part of the laboratory activities, the setup comprehended some high-end equipment that allowed to look deep in the nature of the electrical signals of the system. The quality of the instruments in Sec.4.1.3 presented low levels of noise and elevated bandwidths. While gradually ensuring that those high-end characteristics were not strictly necessary, the performances of the equipment were progressively lowered down to the "real application" setup, suitable for mass production. The familiarity with the instruments and with the characteristics of the system was acquired hand to hand; likewise, the ability of extracting useful acquisition was gradually gained.

4.2.1 The first setup

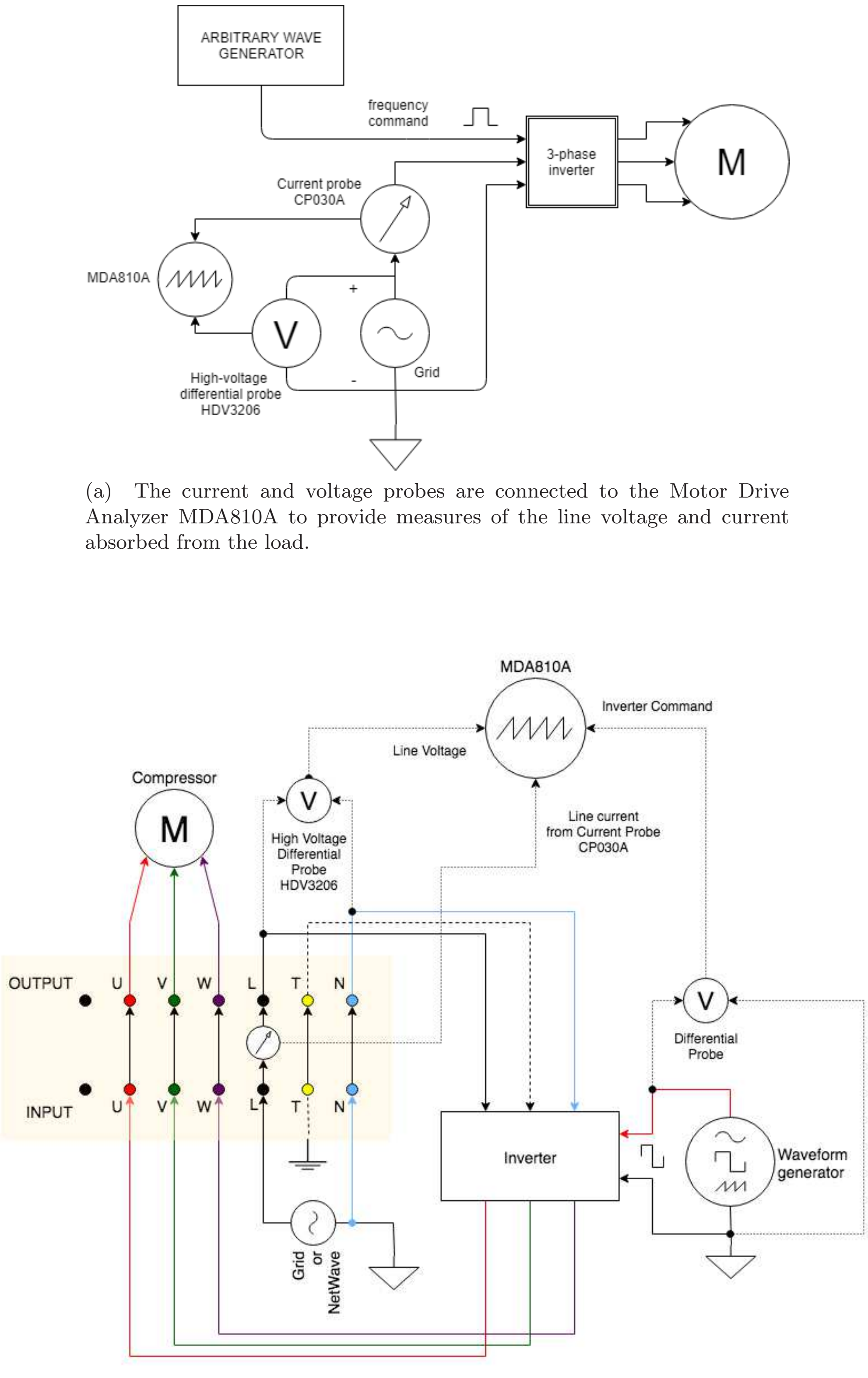
The first type of setup was meant to investigate the nature of electrical signals (*i.e.* voltage and current) on various refrigerator sub-system, composed by couples of inverters and compressors. Since every numerical elaboration on the data acquired had to be made offline and outside the experimental setup (on Matlab or in C language, see Sec.5.1.1 and Sec.5.1.2), the approach was a sort of external observation.

Fig.4.4 illustrates the setup: an arbitrary wave generator gives a 12 V PWM command modulating its frequency in a proper range. The inverter responds to that command regulating the voltage of the stator, thus provoking the movement of rotor of the compressor at a corresponding speed. The current probe is clamped on the line cable supplying the inverter, so that the current absorbed by the load is measured and stored on the Motor Drive Analyzer. Also the line voltage is monitored.

Using this setup, many records of voltage and current of several couples of inverter and compressor were taken. Iin this phase, the purpose was to test the wavelet decomposition software and spot as many peculiarities as possible, in order to refine a classification method to differentiate the couples from one another.

4.2.2 Setup with DAQ module

On a second stage, the elaboration of the data was still offline and outside the setup. An important difference, though, was the decrease in bandwidth introduced making the current pass through the ACS718 and measuring it with a Data AcQuisition (DAQ) module (Fig.4.4). The National Instruments' DAQ NI-9205, supplied by the cDAQ NI-9174 chassis, allowed to sample the



(a) The current and voltage probes are connected to the Motor Drive Analyzer MDA810A to provide measures of the line voltage and current absorbed from the load.

(b) An *ad hoc* box was designed to let the access to line and three-phase more simple.

Figure 4.4: Diagram of the first experimental setup.

differential signal VIOUT–GND at an assigned rate and write the data on a file, by means of a LabView program.

The elaboration of new data was still the same as in Sec.5.1.1 and Sec.5.1.2, but the approach was slightly different, since a new part of the real setup was introduced, even though as a passive component. The main difference from the setup in Sec.4.2.1 is that the bandwidth of the current signal was limited by the ACS718. To accomplish this, the current sensor, welded on the motherboard, needed to be powered: before the PCB, an AC/DC converter was placed to provide the board with a 12 V DC supply. On the board, there also was a DC/DC converter in turn providing a 5 V DC power supply necessary to the ACS718. In this way, when the current absorbed by the load flew through the sensor, it responded with a proportional voltage that could be measured with the DAQ between the test-point S163 and the ground reference. To avoid disturbance from the grid in the measure of the current, the 230 V power supply was taken from the EMtest NetWave Test Supply.

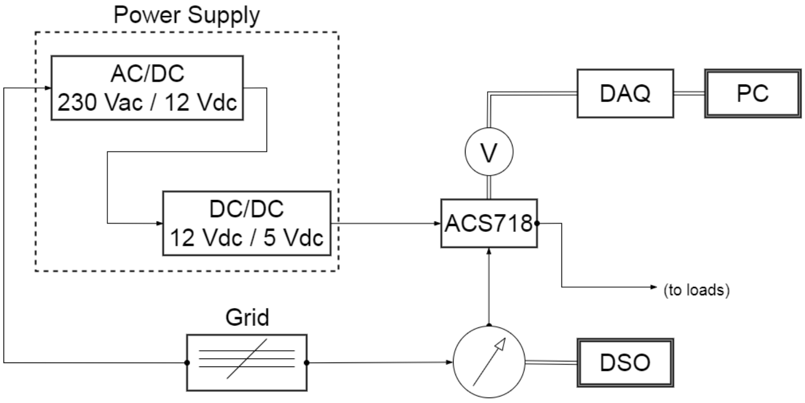
The purpose of the analysis with this setup was again to differentiate couples of inverter and compressor from one another, but also to find peculiarities in the current signals of faulted inverter: those faults were voluntarily activated with manual buttons (like in Fig.4.4b).

4.2.3 Setup with MCU

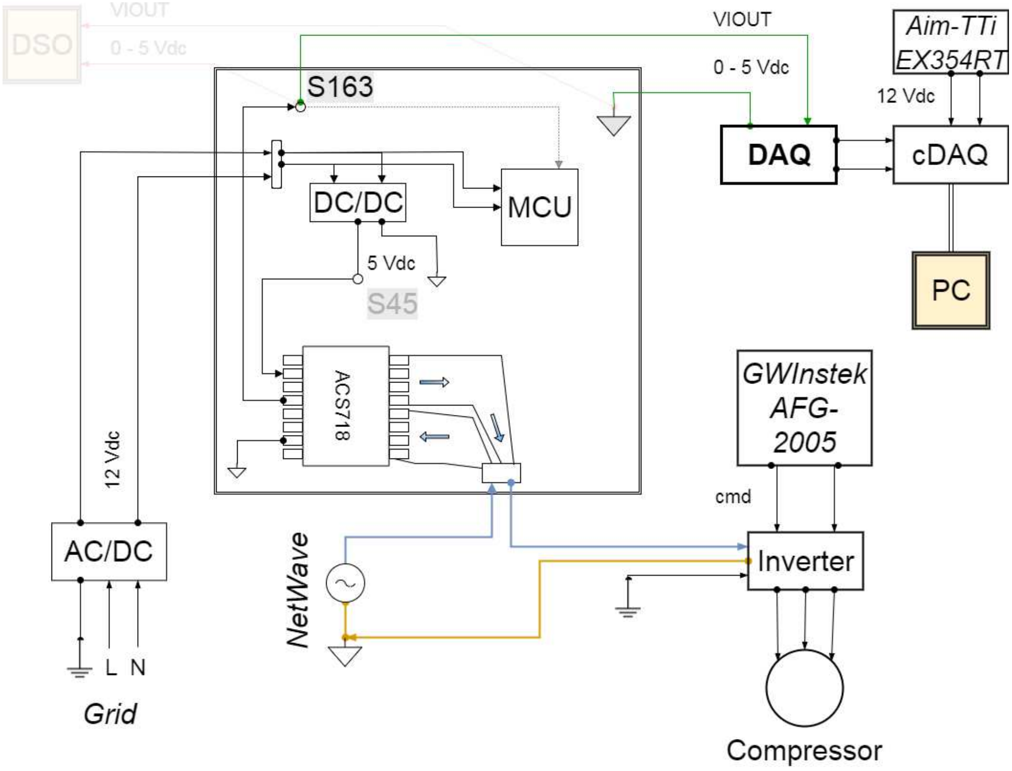
The last stage was performing an online in-the-loop acquisition and classification of the current measures: the current passed through the ACS718 and its measures were acquired using the ADC of the MCU (Fig.4.4).

This last setup is the simplest from the point of view of the hardware, since every signal is managed internally: once the motherboard is supplied, it supplies in turn the ACS718, which responds to the passage of the load current with a proportional voltage. This voltage is then sampled by the internal ADC of the MCU, which then stores and analyzes the data.

In this last phase, the main purpose was to acquire samples of faulted inverters (faults that could be activate with a button, as it can be seen in Fig.4.4b) and to identify them online immediately.

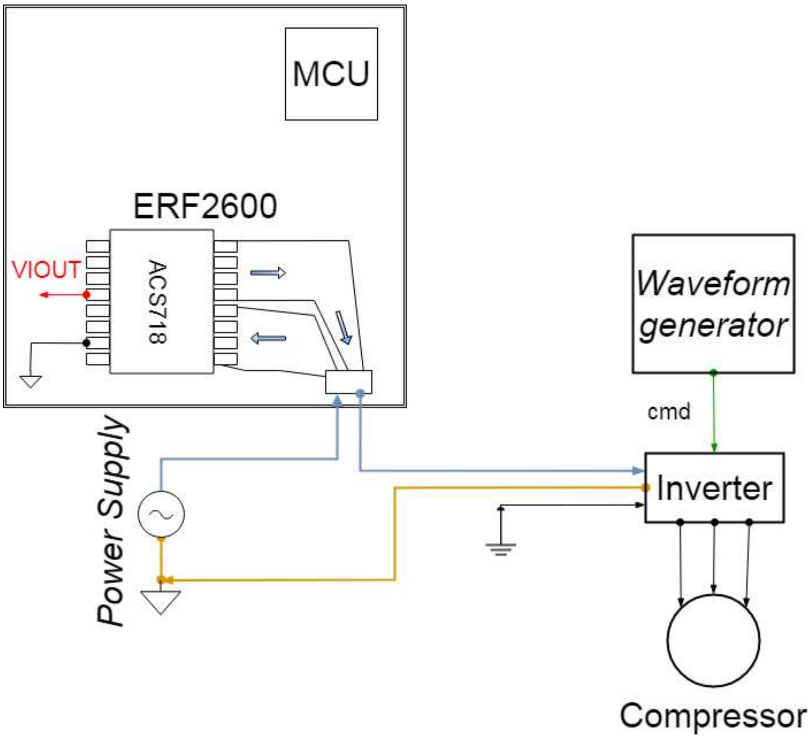


(a) Block diagram reporting the logic of the setup.

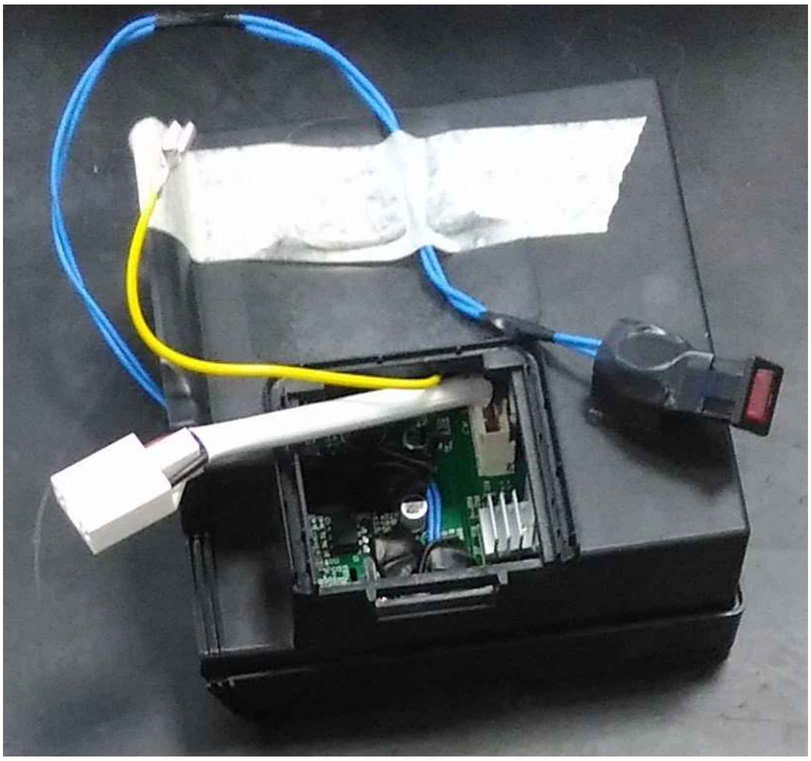


(b) More advanced setup with DAQ. S163 and S45 are the testpoints on the motherboard.

Figure 4.4: Setup with NI DAQ 9205.

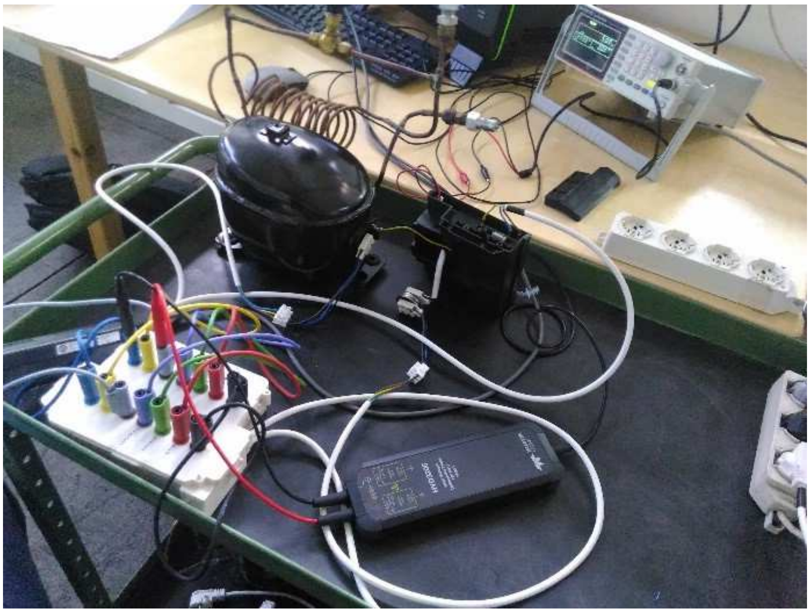


(a) Diagram of the final setup.

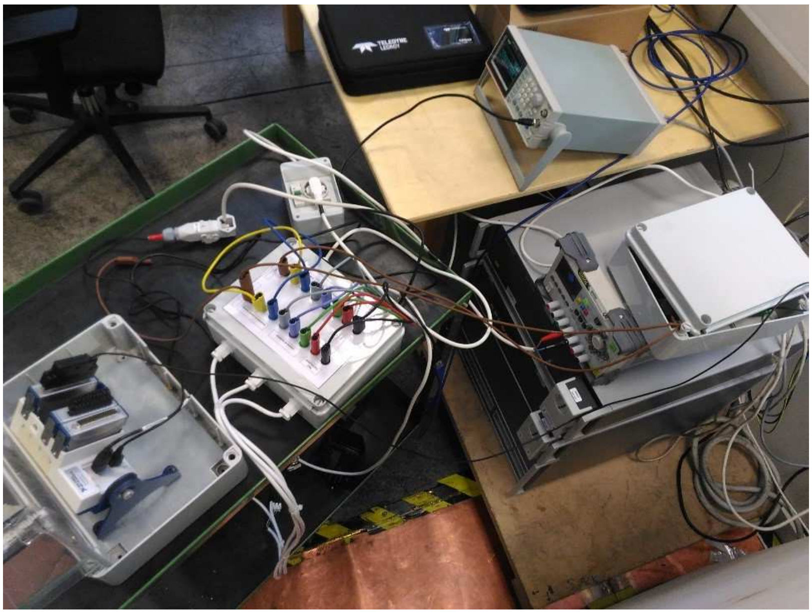


(b) Example of prototype inverter with button to activate the fault.

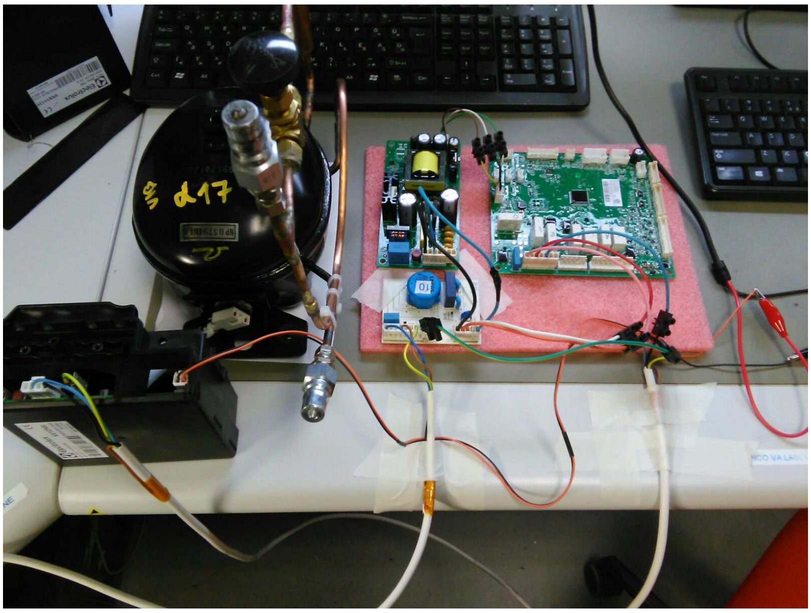
Figure 4.4: Final setup without any external instrument or acquisition board.



(a) Initial setup. The differential voltage and current probes connected to the Motor Drive Analyzer can be spotted (Sec.4.2.1).



(b) Setup with DAQ (Sec.4.2.2).



(c) Final setup (see Sec.4.2.3).

Figure 4.3: Photos of the evolution of the setup.

Chapter 5

The code

A large portion of time has been dedicated to coding, mainly in two languages: Matlab and C.

5.1 The logic of the code

The exploration of the subject has been made exploiting Matlab libraries and documentation, especially in the first phase of the work. Since such functions were not fully open-source, personal C functions were wrote with a further effort so that the results were the same as Matlab. The purpose of the coding work was providing the company with a fully original code that would have enabled the Wavelet Packet Analysis and the Classification on the final application. A further reason why original C code had to be written was that the program running on the MCU was already written in that language and the new one had to be integrated. In the following, no piece of code will be displayed in order to maintain reserved what was developed for Electrolux S.p.A., but rather the approach and the obstacles encountered during the development will be described.

5.1.1 Coding on MATLAB

Especially for the initial part of the research, Matlab's library and documentation were useful to understand the nature of the wavelet decomposition and to explore and experiment various alternatives in a protected simulation environment. The software was always running offline on a local machine where the files of the acquisitions were imported.

No real-time deadline was required and the focus was on the research of new competences and strategies to come up with a possible resolution of the problem. At first, not much attention was paid either to the computational limits of the final MCU or the limitations of the C language (such as the disconvenience of operating with matrixes). The approach was to exploit all the resources that the

Matlab environment offered in order to solve the problem and then adapt the code to a C version.

Finally, a simulation environment where the user can create a Train Set and compare it with a Test Set of coefficients extracted from the wavelet decomposition has been developed. The psudocode summarizing the algorithm created on Matlab can be seen in Psd.5.1.1.

Pseudocode 5.1.1

```
                                                              ▷ Script 1
for all acquisitions wanted do
                                ▷ Data should be on two columns (time, signal)
    import data from .DAT file
    save as variables in .MAT files
end for
                                                              ▷ Script 2
for all .MAT files do
    load the variables
    adapt the length of the variables
    perform the Wavelet Packet Tree
    compute the proper Energy Map
    save the Energy Map as a vector               ▷ Draw graphs if you want
end for
                                                              ▷ Script 3
for similar Energy Maps do
    create sets denoting classes
end for
                                                              ▷ Script 4
choose a type of distance for the classification
choose a Train Set and a Test Set
classify the Energy Maps of the Test Set according to the Train Set
```

5.1.2 Coding in C language on a PC

Parallelly to new achievements with the code in Matlab, the needed functions were created in C language, adapting the structures and ways of thinking of C language to the math of wavelet analysis. Translating the theory and the matrix mathematics of Matlab into a lower-level language has lead to some convoluted bunches of structures and pointers. By the time, anyway, C skills were developed so that the code was simplified to run on a much less powerful CPU like the MCU.

Practically, the code was developed on a Linux Virtual Machine (on the distribution Ubuntu 16.04), even employing Eclipse GUI to manage the project. The solely purpose was to imitate the parallel Matlab software, but it turned out to be a precious way to better understand the problem and the computation underneath. As a consequence, the code follows the trace of Psd.5.1.1.

5.1.3 Coding in C language on a MCU

In the final part of the work, the code was further simplified and some particular adjustments were made to save memory and computational time (*e.g.* allocating no more than the necessary space for the variables, with particular attention to the type). The program was developed on Windows 7 (on Atom GUI), since the compilation and documentation algorithm of Electrolux R&D required the compilation via batch files.

In this phase, the purpose was to run the program online on a prototype motherboard using the ACS718 to obtain bunches of measures, decompose them with the Wavelet Transform and classify them. Investigations on the behavior of the system and automatic recognition of abnormalities were carried out. The final code was able to collect Train Sets (if needed) and compare each new acquisition with a provided database of known cases. The pseudocode describing the operation on the MCU is given in Psd.5.1.2. A variant of the code has a predetermined Train Set to which compare the Energy Maps from the new acquisitions.

Pseudocode 5.1.2 ▷ *In main.c*

```
// Initialization of the system
while TRUE do
    // Periodic functions of the system
    if a proper number of samples has been collected then
        Compute the WPD and the Energy Map of the current bunch of data
                                        ▷ relocate the memory as soon as you can
        if the Train Set is not full then                 ▷ Collect the Train Set
            Save the current Energy Map as part of the Train Set
        else if the last energy map filled the Train Set then
            Setup the classification
        else
            Launch the classification of the current Energy Map
        end if
        Reset the counter of the acquired samples.
    end if
end while
```

5.2 The possibilities of the current code, critical issues and future perspectives

What has been achieve is a graphic environment in Matlab to perform simulations and tests offline, and an integrated part of code that performs online pattern recognition on the MCU. Anyway, there are some critical aspects that will have to be solved in the perspective of mass production:

- the supervised learning algorithm needs a lot of tests on voluntarily faulted components in order to collect consistent train sets, with *ad hoc* setups, implying some costs;
- the ADC conversions, the decomposition and the classification occupy some computational resources on the single core of the MCU and are racing with the other processes to be scheduled. This provokes delays that alter the sampling frequency on the real application and this preempts the frequency analysis.
- the result of the classification is related to some choices: the type of decomposition (the filter banks), the set of features and the type of distance considered in the classification algorithm, for instance. An optimized choice sould be made to allow the recognition of most of the cases.

What can be done in the future could be a smart selection of the faults that could be useful and economically convenient to investigate, in order to sacrifice the fewest amount of components during testing and learning. The identification of specific faults to place in the Train Set could also help choose the optimized type of decomposition, which features to consider and what kind of distance opt to, restricting the variability.

Moreover, there should be a dedicated CPU core to collect the data with a precise sampling rate, avoiding the ongoing race condition that causes delays. However, this would imply a complication in the design of the PCB of the motherboard and a rise of the costs. Nevertheless, the most convenient solution would probably be to create a concurrent program on a multi-core platform.

Chapter 6

Case study and Conclusions

6.1 Case study

A case study to show an example of application of what was described so far is presented in this small chapter. The approach to this case has been replied on the investigation of several other voluntarily activated faults.

6.1.1 The problem

In the laboratory, a compressor was found to “knock” when turning off (and sometimes turning on), *i.e.* emitting an audible brief sound due to a mechanical impact inside the case. Such impacts, repeated in time, could lead to premature usury of the components. Such compressor was then investigated not from the point of view of its mechanical parts, but from the electrical standpoint. From the MRA of the current signal, some interesting features were detected.

6.1.2 The approach

Since the discover was made while the setup evolution was still in the first phase, the setup was the one described in Sec.4.2.1. The line cable supplying the inverter was clamped with the current probe CP030A, connected to the MDA810A. At first, various acquisitions on switch on/off cycles were conducted starting with a large bandwidth (sampling rate at 1 GHz) and then, progressively, the sampling rate was restricted since no particular component was spotted at high frequency (using the Fast Fourier Transform (FFT) of the MDA810A).

6.1.3 The results

Lowering the sampling rate enables to focus on narrower bandwidth according to the Nyquist’s Rule: that is why, once an abnormal peak in the FFT was spotted at about 23 kHz (Fig.6.1), the sampling frequency f_s was set to 50 kS/s, *i.e.* the maximum frequency that could be seen was f_{nyq}=25 kHz. This was the most appropriate bandwidth to inspect on that peculiar peak.

After taking various acquisitions on that setup, the MRA with the Wavelet Packet Decomposition was performed so that the moment when the suspected spectral component was manifesting could be accurately spotted. As a matter of fact, the FFT that helped to find the unusual component was just giving a global vision of the signal, thus not providing any information about the nature of that component. The FFT just told that a part of the energy of the signal was some-when located at that frequency. With the wavelet analysis, instead, the location and intensity of such component could be highlighted according to the time-frequency resolution trade-off presented in Sec.2.1.

As it can be seen in Fig.6.3, that component turn out to be periodic and always present during the on-time of the system. By the way, if it had been impulsive, sporadic or even random, that fact could have been proved only by the WPD.

Computing the Energy Map of the wavelet decomposition of such signal returns a relatively high energy on the node comprising the energy of the unusual components, with respect to all the other non-faulty compressors (see Fig.6.2).

That is why, with a restricted set of energies as feature vectors, this compressor could easily be spotted among the others and marked as faulted. One could remonstrate that this classification could have been performed successfully also with a FFT and, for this particular case, that would have been true, since the distinctive component was periodic. However, the nature of faults cannot be foreseen on complex systems, so the WPD must be recognized as a tool that provides better diagnostic, furnishing more information than Fourier's analysis at the cost of a lower frequency resolution.

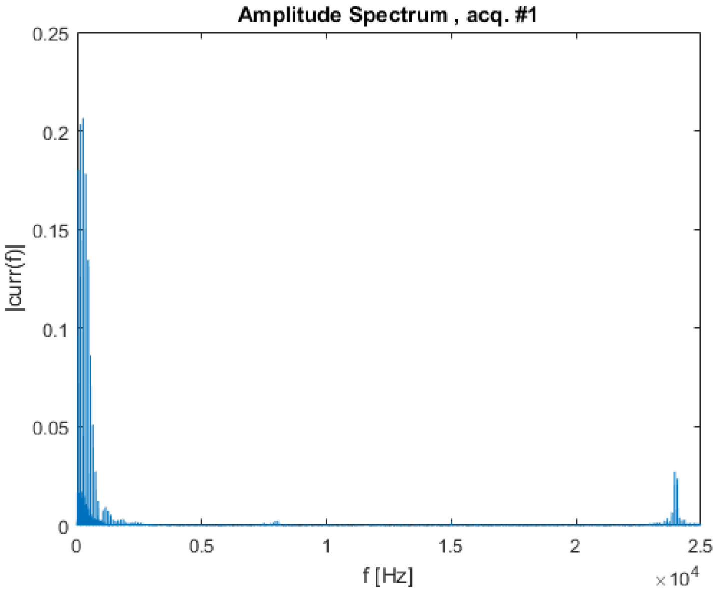


Figure 6.1: The FFT spots an abnormal concentration of energy at about 23 kHz.

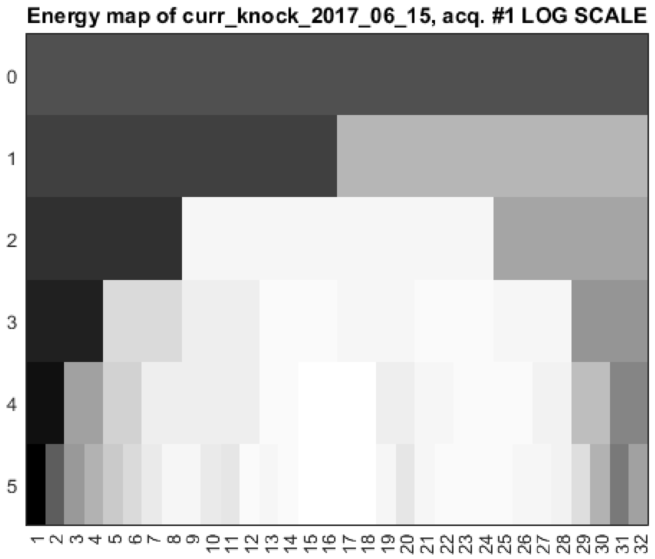


Figure 6.2: This logarithmic-scaled energy map highlights a suspect concentration of energy on the nodes at the highest frequencies.

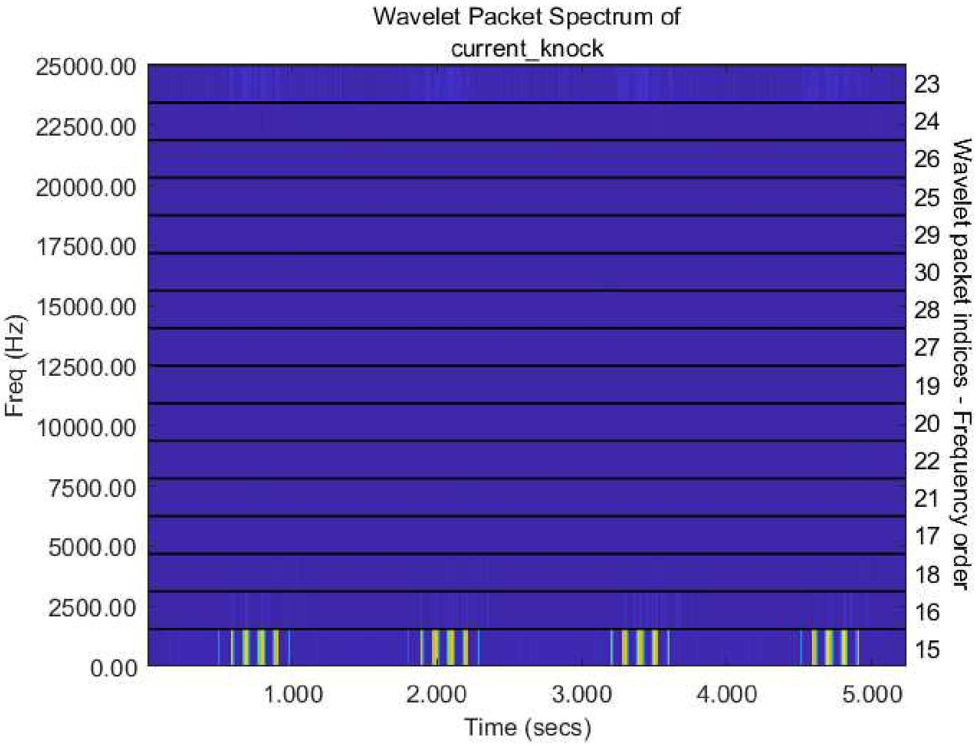


Figure 6.3: This Wavelet Packet Spectrum highlights a suspect persistent presence of energy on the nodes at the highest frequencies of the visible spectrum (maybe it is a bit hard to notice without zoom).

6.2 Conclusions

As it has been illustrated during this work, starting from the most used method of spectral analysis of signals, *i.e.* the Fourier's decomposition, a mathematical alternative was proposed with the wavelet decomposition. Exploiting the Multi-Resolution provided by this kind of analysis, an energy map of the signal both in time and frequency could be created. Using that map or parts of it as a feature set representing the signal, an algorithm of classification (in this case, following the Nearest-Neighbor rule) could recognize future acquisitions of the same signal among different sets representing other signals.

The experimental setup evolved from the minimal system made of the inverter and the compressor, whose absorbed current was measured with a high-end current probe. Then, proceeding with and intermediate in-the-loop acquisition device, the final setup was finalized with the real-application motherboard, with a cheap on-board current sensor. Eventually, these measures were acquired with an internal ADC of the MCU and the following wavelet decomposition, calculation of the energy map and classification were performed online by the MCU.

In Sec.6.1 an example of application of the wavelet analysis was presented. Some time-frequency features extracted from the MRA were shown to be useful in the following classification of the signal in exam. Nevertheless, as spotted in Sec.5.2, there were some limitations on the applicability of the method and more experimentation was to do before taking the procedure to mass production.

Part III

References

Bibliography

[1] Allegro MicroSystems LLC: *ACS718.* June 2015. – Rev. 01 - Corrected pinout diagram

[2] Boyer, Keith G.: *The Fast Wavelet Transform*, DeVry Institute of Technology, Master's Thesis, 1995

[3] Daubechies, Ingrid: *Ten Lectures on Wavelets.* Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 1992. – ISBN 0-89871-274-2

[4] DSPRelated.com: *Upsampling and Downsampling.* – `https://www.dsprelated.com/freebooks/sasp/Upsampling_Downsampling`, Accessed: 06-Sept-2017

[5] Fowler, Mark: *Wavelet Transform Theory.* – Lecture slides, State University of New York at Binghamton

[6] Hu, Li-Yu ; Huang, Min-Wei ; Ke, Shih-Wen ; Tsai, Ching-Fong: The distance function effect on k-nearest neighbor classification for medical dataset. In: *SpringerPlus* (2016), August. – Online published `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4978658/`

[7] : *Essentials in Wavelet Theory.* Online available `http://www.colorado.edu/engineering/CAS/courses.d/ASEN5519.d/kaist.lecture.11.pdf`

[8] : *The Fast Fourier Transform.* Online available `http://www.colorado.edu/engineering/CAS/courses.d/ASEN5519.d/kaist.lecture.03.pdf`

[9] Kaplan, Ian: *The Wavelet Packet Transform.* March 2002. – `http://www.bearcave.com/misl/misl_tech/wavelets/packet/`, Accessed: Apr-2017

[10] Learned, Rachel E.: *Wavelet Packet Based Transient Signal Classification*, Massachusetts Institute of Technology, Master's Thesis, September 1992

[11] Learned, Rachel E. ; Karl, William C. ; Willsky, Alan S.: Wavelet Packet Based Transient Signal Classification. In: *Time-Frequency and Time-Scale Analysis, Proceeding of the IEEE-SP International Symposium*, IEEE, October 1992. – Conference Location: Victoria, BC, Canada

[12] Liu, Chun-Lin: *A Tutorial of the Wavelet Transform.* February 2010

[13] Liu, Yonggang ; Liang, X. S. ; Weisberg, Robert H.: Rectification of the Bias in the Wavelet Power Spectrum. In: *Journal of Atmospheric and Oceanic Technology* 24 (2007), December, p. 2093–2012

[14] Maimon, Oded (Ed.) ; Rokach, Lior (Ed.): *Data Mining and Knowledge Discovery Handbook.* Chap. 15, Springer US, 2005. – Clustering Methods

[15] Mallat, Stephane: *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* 3rd. Academic Press, 2008. – ISBN 0123743702, 9780123743701

[16] MathWorks: *Fast Wavelet Transform (FWT) Algorithm.* – `https://it.mathworks.com/help/wavelet/ug/fast-wavelet-transform-fwt-algorithm.html#f8-3283`, Accessed: 27-Apr-2017

[17] MathWorks: *Wavelet Packets: Decomposing the Details.* – `https://it.mathworks.com/help/wavelet/examples/wavelet-packets-decomposing-the-details.html`, Accessed: 27-Apr-2017

[18] MathWorks: *wentropy.* – `https://it.mathworks.com/help/wavelet/ref/wentropy.html`, Accessed: 15-May-2017

[19] MathWorks: *wpspectrum.* – `https://it.mathworks.com/help/wavelet/ref/wpspectrum.html`, Accessed: 15-May-2017

[20] McCune, Bruce ; Grace, James B.: *Analysis of Ecological Communities.* Chap. 6. Gleneden Beach, OR, USA : MjM Software Design, 2002. – Distance Measures

[21] Mednikov, Eugene: *Wavelet Packets.* 2003. – Lecture slides, Haifa University

[22] Murugappan, Murugappan ; Nagarajan, Ramachadranm ; Sazali, Jacoob: Classification of human emotion from EEG using discrete wavelet transform. In: *J. Biomedical Science and Engineering* 3 (2010), p. 390–396. – School of Mechatronic Engineering, University Malaysia Perlis (UniMAP), Perlis, Malaysia

[23] NXP Semiconductors: *KE04 Sub-Family Reference Manual.* July 2016. – Rev. 03 nÂ°. MKE04P80M48SF0RM

[24] Pan, Guan-Chen: *A Tutorial of Wavelet for Pattern Recognition.* – Taipei, National Taiwan University

[25] Polikar, Robi: *The Wavelet Tutorial.* Online available `http://web.iitd.ac.in/~sumeet/WaveletTutorial.pdf`. 1996. – Second Edition

[26] Strang, Gilbert: *Linear Algebra and its application.* Chap. 6.3, p. 331–338. Belmont, CA, USA : Thomson Learning, 1988. – Singular Value Decomposition

[27] Strang, Gilbert: Wavelets and Dilation Equations: A Brief Introduction. In: *SIAM Review* (1989), No. 31, p. 613–627. – Online available `http://www.colorado.edu/engineering/CAS/courses.d/ASEN5519.d/siamrev.pdf,Accessed:22-Aug-2017`

[28] Strang, Gilbert: Wavelets. In: *American Scientist* (1994), April, No. 82, p. 250–255. – Online available `http://www.colorado.edu/engineering/CAS/courses.d/ASEN5519.d/amsci.pdf`, Accessed: 22-Aug-2017

[29] Tang, Yuanyan: Status of pattern recognition with wavelet analysis. In: *Front. Comput. Sci. China* (2008), p. 268–294

[30] Torrence, Christopher ; Compo, Gilbert P.: A Practical Guide to Wavelet Analysis. In: *Bulletin of the American Meteorological Society* (1998), p. 61–78

[31] Wang, Ruye: *Properties of Z Transform.* September 2014. – `http://fourier.eng.hmc.edu/e102/lectures/Z_Transform/node6`, Accessed: 06-Sept-2017

[32] Wickerhauser, Mladen V.: *Lectures on Wavelet Packet Algorithms.* April 1992

[33] Wikipedia: *Quadrature mirror filter.* – `https://en.wikipedia.org/wiki/Quadrature_mirror_filter`, Accessed: 30-Aug-2017

[34] Zhang, G. ; Lu, Z. ; Ji, G. ; Sun, P. ; Yang, J. ; Zhang, Y.: Automated Classification of Brain MR Images by Wavelet-Energy and k-Nearest Neighbors Algorithm. In: *2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, Dec 2015, p. 87–91. – ISSN 2168-3042

List of Figures

Glossary

f_{nyq} Nyquist frequency is generally considered as a half of the sampling rate (f_s). By the Nyquist-Shannon sampling theorem, this is the maximum bandlimit that allows a perfect reconstruction of the original signal. 74

f_s sampling rate. 14, 20, 73, 87

ACS718 Allegro Current Sensor ACS718-10B. 59, 61, 63, 71

ADC Analog-to-Digital Converter. 23, 59, 63, 72, 77

ARM Advanced Reduced Instruction Set Computer (RISC) Machine. 2

CS Customer Service. 1, 2

CSTs Customer Service Technicians. 1, 2

CWT Continuous Wavelet Transform. 10, 12, 13, 14, 15, 16, 21

DAQ Data AcQuisition. 61, 63

DWT Discrete Wavelet Transform. 16, 20, 21, 24, 25, 28, 30

FFT Fast Fourier Transform. 73, 74

FIR Finite Impulse Response. 25

FT Fourier Transform. 5, 6, 7, 8, 9, 11, 14, 18, 20, 34, 45

IC Integrated Circuit. 59

MCU Microcontroller Unit. 2, 49, 59, 63, 69, 70, 71, 72, 77

MDA810A Motor Drive Analyzer. 59, 62, 73

MRA The Multi-Resolution Analysis gives the tools to study a signal along multiple dimensions simultaneously, *e.g.* both time and frequency. 7, 8, 41, 73, 74, 77

PCB Printed Circuit Board. 1, 2, 63, 72

PWM Pulsed Width Modulation. 57, 58, 61

R&D Research & Development. 1, 71

RISC Reduced Instruction Set Computer. 87

STFT Short-Time Fourier Transform. 6, 10, 11, 12, 14, 26, 28, 45

SVM Space Vector Modulation. 57

WPD Wavelet Packet Decomposition. 28, 30, 32, 36, 37, 42, 45, 74

WPS Wavelet Packet Spectrum. 41, 42, 44

WP Tree Wavelet Packet Tree. 26, 27, 28, 29, 30, 31, 33, 34, 40, 41, 42, 45, 46, 48

WPT Wavelet Packet Transform. 25, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 41, 44, 45, 48, 49

WST Wavelet Series Transform. 15

WT Wavelet Transform. 7, 8, 9, 10, 11, 12, 13, 15

Acknowledgments - *Ringraziamenti*

Dopo un anno di ricerca e sfide passato in azienda (e un altro anno trascorso per finire gli esami...), è finalmente tempo di veder coronato il mio tormentato percorso verso questa laurea magistrale. In tale circostanza, alcuni ringraziamenti sono d'obbligo, perché da solo veramente non ce l'avrei mai fatta.

In primis, ringrazio la mia famiglia, perché mi ha sempre sostenuto in tutti i modi e, con i miei amici, ci ha creduto anche quando io non ci credevo più. Tutti loro sono stati la piattaforma su cui ho potuto arrestare la mia caduta quando le cose non andavano bene e su cui ho potuto darmi lo slancio una volta riacquistata la fiducia in me stesso necessaria per completare le mie fatiche universitarie.

Poi, chiaramente, debbo dei ringraziamenti alle persone che mi hanno aiutato durante l'anno trascorso presso la sede di Susegana di Electrolux Italia S.p.A., a partire dal team elettronico di Luca Rizzo, in cui sono stato inserito. Tutti mi hanno fatto sentire accolto, trasmettendomi conoscenze e buon umore quotidianamente. In modo paricolare, ci tengo a ringraziare Massimiliano Soligo per tutto il tempo e gli sforzi che ha dedicato affinché la mia parte di lavoro in laboratorio potesse svolgersi al meglio. Ho molto a cuore anche ringraziare Stefano Zanin per le ore dedicate a discutere e rivedere il codice. Allo stesso modo, Stefano Dal Ben per la disponibilità dimostrata nella preparazione della parte meccanica.

Certamente niente di tutto questo sarebbe potuto accadere senza la guida e il supporto del mio relatore, il prof. Alessandro Sona, che ringrazio per l'opportunità che mi ha fornito e per l'accompagnamento in questo percorso.

Dulcis in fundo, merita un ringraziamento enorme il mio tutor aziendale (e, posso dire, amico) Tristano Dal Canton. Si è impegnato ogni giorno a coordinare me e gli altri colleghi in modo che il mio lavoro di tesi potesse andare avanti. Ha saputo guidarmi nelle attività e indirizzarmi nei momenti di dubbio con professionalità. Ma soprattutto è stato sincero, spontaneo e comprensivo.

Inizialmente, il periodo di internship per la tesi era previsto durasse sei mesi. Ma vari fattori come l'ottimo ambiente di lavoro, la numerose attività che restavano ancora da svolgere (in quello che era diventato un progetto anche in parte "mio") e altre circostanze della vita, mi hanno portato a prolungare tale *periodo di altri sei mesi. Anche se ciò ha comportato un inevitabile ritardo in quelle che erano le mie previsioni per gli studi, non mi pento della mia scelta, conscio che questa è stata una validissima esperienza professionale e di vita.*

Grazie a tutti quelli che non vedevano l'ora, come me, che arrivasse il giorno in cui mi sarei laureato.

Ora sono finalmente pronto a cominciare un capitolo tutto nuovo della mia vita.