COPPE
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
UFRJ

TÍTULO DA TESE

Luiz Rennó Costa

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientadores: Nome do Primeiro Orientador Sobrenome
Nome do Segundo Orientador Sobrenome
Nome do Terceiro Orientador Sobrenome

Rio de Janeiro
Fevereiro de 2011

TÍTULO DA TESE

Luiz Rennó Costa

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

Prof. Nome do Primeiro Examinador Sobrenome, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, Ph.D.

Prof. Nome do Terceiro Examinador Sobrenome, D.Sc.

Prof. Nome do Quarto Examinador Sobrenome, Ph.D.

Prof. Nome do Quinto Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2011

Rennó Costa, Luiz

Título da Tese/Luiz Rennó Costa. – Rio de Janeiro: UFRJ/COPPE, 2011.

XI, 15 p.: il.; 29, 7cm.

Orientadores: Nome do Primeiro Orientador Sobrenome
Nome do Segundo Orientador Sobrenome
Nome do Terceiro Orientador Sobrenome

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Civil, 2011.

Bibliography: p. 15 – 15.

1. Primeira palavra-chave. 2. Segunda palavra-chave. 3. Terceira palavra-chave. I. Sobrenome, Nome do Primeiro Orientador *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Lakad Matataaag! Normalin,
Normalin.

Agradecimentos

Gostaria de agradecer a todos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

TÍTULO DA TESE

Luiz Rennó Costa

Fevereiro/2011

Orientadores: Nome do Primeiro Orientador Sobrenome
Nome do Segundo Orientador Sobrenome
Nome do Terceiro Orientador Sobrenome

Programa: Engenharia Civil

Apresenta-se, nesta tese, ...

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

THESIS TITLE

Luiz Rennó Costa

February/2011

Advisors: Nome do Primeiro Orientador Sobrenome
Nome do Segundo Orientador Sobrenome
Nome do Terceiro Orientador Sobrenome

Department: Civil Engineering

In this work, we present ...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The increasing demand for energy over the past decades created a surge of renewable sources, as opposed to the more traditional petroleum fuel. However, that still is an essential fuel for our society, so that several companies like Shell, Exxon, and Petrobras have extensive and complex networks for extracting and distributing different kinds of liquid fuel through pipelines. Brazil for instance has roughly 8.000 kilometers (km) [2] of oil pipelines (both refined and crude) scattered throughout the country.

In order to maintain and prevent failure (which could be catastrophic) several tests, both destructive and non-destructive were developed, and amongst the latter, one that stands out is the Acoustic Emission (AE) test. It relies on the radiation of acoustic (physical) waves that occurs when a material undergoes irreversible changes both at a micro and macroscopic scale (Section 1.5).

Those changes can come from the propagation of a crack, corrosion, or in other words, irreversible damage to the material structure. Therefore it should be possible to monitor and tell in real time how dangerous it is.

1.2 Theory

This section gives a brief revision of the theoretical topics this thesis extends upon.

1.2.1 Learning

Learning is something all beings have experienced one way or another, it is intrinsic to our nature and an essential process to our evolution as a society. It can be defined as a acquisition of knowledge through interaction, be it with the outside world like books, people, or with one's own self.

Knowledge however, is a more abstract concept and can be interpreted in several different ways, for instance, both knowing how to sew a scarf and differentiating square from circle can be considered knowledge. The different between those is that in the latter, the knowledge is static, interacting with it means only identifying which is which. Knowing how to sew a scarf however, means attaining domain over the cloth's process of transformation, given a simple piece of cloth (input), one would always be able to transform it into a scarf.

The ability to learn a transformation process is something rather powerful because one can change properties of the output merely changing the input, instead of relearning the whole process again, for example, being taught how to sew using only blue cloth does not impede one to sew a red scarf, needed only to change the fabric one applies the process.

Trying to create a machine that is able to act like we (thinking entities) do has been an open challenge since the 1950s [3] and motivated countless studies amongst several decades on the field of Machine Learning (ML).

It is important to note however that mathematical complexity is a central factor in ML. Separating squares from circles can be done using rather simple mathematical formulae, sewing a scarf not. One can even map all the hand movements and develop formulae that reign it, but the complexity would make computation unrewarding. The goal of ML is to create a simpler model capable of acquiring that complex knowledge through learning.

However, one problem is still unresolved, how to measure the learning process. One can find it learnt how to sew a scarf when it does not deforms when pulled, or when it satisfies someone else. Expanding on that latter concept, one can for instance, attribute a beauty scale for the scarf (suppose for simplicity that the scale goes from 1 to 10), and only consider that it learnt how to sew a scarf when it can reliably sew a scarf that receives a 10 from the beauty scale.

The creation of a measure capable of telling if learning is in fact happening is of utmost importance for the learning process. We can then extend the definition of learning to the acquisition of knowledge through interaction that, as a by-product, increases the performance measure associated with that knowledge. The contrary is also possible, instead of increasing a performance measure, one can decrease an error measure.

Learning from the ML point of view can be separated in 3 main classes, Supervised Learning (Section 1.2.2), Unsupervised Learning (Section 1.2.3) and Reinforced Learning. All those have the same principles, they all have a model, performance measure and a task to perform. The main difference is how they receive and interpret the input data.

1.2.2 Supervised Learning

As an undergrad is taking a calculus course, he is given several examples and exercises. Examples are composed of simpler questions and the necessary procedures, after reading them and taking the more challenging exercises, he gives them to the teacher, which in turn proceeds to tell him what he did right or wrong. He is then given a score based on how many questions he had right (how many hit the "target"), this procedure is repeated until the undergrad is satisfied with his score.

That is a prime example of Supervised Learning, as it can be defined as a method to train the model by directly teaching it. This is achieved through the addition of "labels" or "targets". The procedure from a mathematical point of view (Figure 1.1) is fairly straightforward, your input data $\mathbf{x}$ is fed to a model that yields the output $\tilde{\mathbf{y}}$. The output is compared to the target $\mathbf{y}$ and a performance/loss score $L(\cdot)$ is calculated.

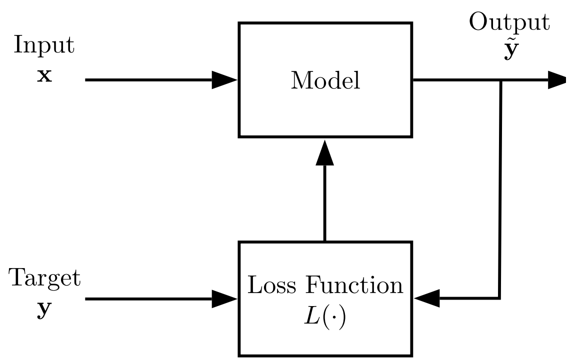


Figure 1.1: Simplified Supervised Learning Schematic

1.2.3 Unsupervised Learning

Contrary to the aforementioned Supervised Learning method, Unsupervised Learning does not have a well specified "target" and therefore the model now has a different purpose, to create some sort of structure based solely on the inputs and the relationships between them. Unsupervised Learning typically translates to clustering.

Clustering is the action of grouping together inputs that have some degree of similarity, for instance, trying to separate different animals by the number of legs they possess, or which environment they reside is a type of unsupervised clustering since it depends only on the characteristics of the input data. A good example of this approach is recommendation systems.

1.3 Artificial Neural Networks

What mainly highlights our species is the seemingly unending capacity to learn and adapt. The organ responsible for such, the brain has been subjected to

Over the years, several studies (?????) have been published on the brain and its composition although several things are still undiscovered or undetermined.

The brain is the most complex organ in the human body and hosts 10 to 20 billion neurons, its main component. These neurons (Figure 1.2) are physically connected through its dendrites and axons, these connections are susceptible to electrical impulses called synapses, most neural cells send signals through the axons and receive them from the dendrites.

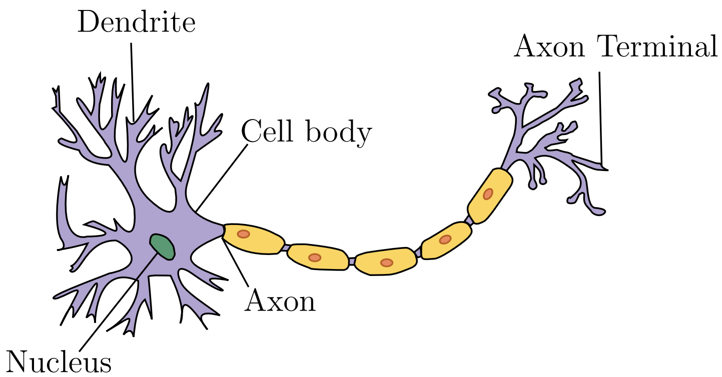


Figure 1.2: Structure of a Typical Neuron. Taken from [1]

While these operations are well understood, the way millions of interconnected neurons work and cooperate is still undiscovered(???). The Artificial Neural Network (ANN) is a supervised learning model (as the name states) inspired by that complex structure of the interconnected neurons, mainly composed of *Perceptrons*.

1.4 Perceptron

The *Perceptron* can be interpreted as mathematical model for the neuron (Figure 1.3), it receives an input vector ($\mathbf{x}$) through the dendrites, modifies it and releases an output ($\mathbf{y}$) down the axons.

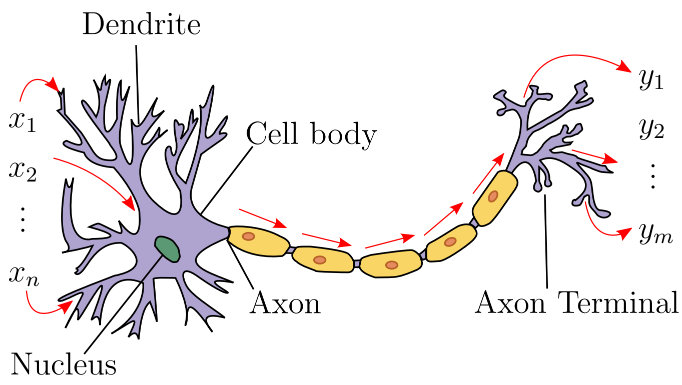


Figure 1.3: Structure of a Typical Neuron with Inputs and Outputs. Adapted from [1].

Where x_n and y_m are elements from the n and m-dimensional input and output vectors ($\mathbf{x}$ and $\mathbf{y}$) respectively. Initially, this model was used as a binary classifier using the synapses as weights, $m = 1$ and the addition of a bias (b) element, which gives:

$$y = f(\mathbf{wx} + b) = sign(\mathbf{wx} + b) = \begin{cases} 1, & \text{if } \mathbf{wx} + \mathbf{b} > 0 \\ -1, & \text{if otherwise} \end{cases} \tag{1.1}$$

Where $\mathbf{w}$ is a $[1\ x\ n]$ vector of weights. This model can be interpreted as a line in the input space that can both be translated (relative to the origin) and rotated. That line acts as a boundary separating two regions, or in other words, classifying two different groups of data.

The *Perceptron* Learning Algorithm (PLA) is rather simple, for

However, the boundary is still a straight line, meaning that if the data is not linearly separable (vast majority of real cases) it is impossible to achieve zero classification error. It would be interesting then to create different sorts of non-linear boundaries, to achieve that, one can "smooth" the underlying function $f(\mathbf{wx})$ using for example $tanh(\cdot)$, $arctan(\cdot)$, etc.

Also, this thesis is going to adopt a slightly different notation, $\mathbf{wx} + b \rightarrow \mathbf{wx}$ where $w_0 = 0$ and $x_0 = 1$ thus giving:

$$\mathbf{y} = f(\mathbf{wx} + b) = f(\mathbf{wx}) \rightarrow \Phi(\mathbf{wx}) \tag{1.2}$$

Where $\mathbf{w}$ are the synapses (weights) and $\Phi(\cdot)$ is an arbitrary mapping (activation function). It is important to note that there is a certain "flow" of information from $\mathbf{x}$ to $\mathbf{y}$, there is no feedback in the *Perceptron* model (Figure 1.4).

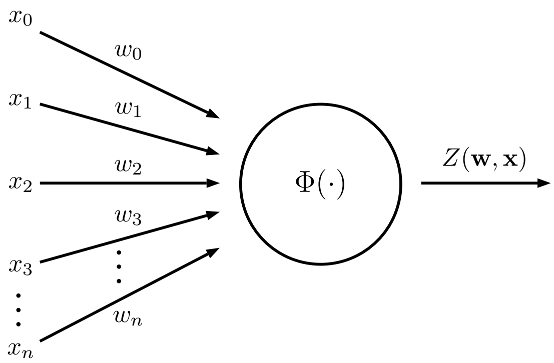


Figure 1.4: Complete Perceptron Mathematical Model.

1.4.1 Multi Layer Perceptron

1.4.2 Backpropagation

1.5 Acoustic Emission

1.6 Bibliography Review

Chapter 2

Materials and Methods

This chapter focus on describing the used materials and methods utilized in this thesis. The acoustic emission tests (Section 2.1) had 3 separate data acquisition systems, one using an industrial device (DISP-16C) made by Physical Acoustics (PAC), one with another industrial device (AMSY-5) made by Vallen Systems and a custom one denominated Streaming.

Both Streaming and Vallen data were analysed throughout the project duration, however, this thesis concentrates solely on the Streaming data. Both are similar in the way that they acquire temporal data from the AE (not its parameters), however, the AE waveforms captured by the Vallen system had several disadvantages when compared to the Streaming counterpart.

The Vallen data is a collection of fixed length AEs concatenated to form a $L \times N$ matrix where L is the AE length and N the number of captured AEs. This matrix was provided as a MATLAB formatted data file (.MAT) containing the waveform using 64-bit double-precision floating-point format. Unfortunately, this system is not guaranteed to capture all waveforms, this severely hinders some essential preprocessing stages (Sections 2.3.2 and 2.3.3), making it a rather unreliable (the captured AE may not be from the crack propagation) and with no means of improving its reliability, therefore all of the Vallen data was discarded.

Thus, this chapter begins detailing the destructive test done, extends to the raw data format used, describes all the preprocessing done and the reasoning behind it, then details the waveform capture procedure all the way to creating the final dataset used to train a neural network model (Section 1.3) with parameters from both the AE temporal data and its frequency spectrum.

2.1 Acoustic Emission Test

The AE tests were performed by engineers of the Physical Metallurgy Laboratory (LAMEF) at the Federal University of Rio Grande do Sul (UFRGS). It uses a closeended steel (API XL 60 series) pipe with 20 inch diameter, 40 metre length and 1.45 centimetres of thickness, a semi-elliptical pre-crack extending until half of its thickness (approximately $0.7cm$) was inserted at half its length.

The hydrostatic test consists of a gradual increase of pressure (after filling the tube) followed by plateaus until it bursts (Figure 2.1), this gradual increase is denominated *cycle*. In total, 4 (four) tests were done, with the first one discarded since it did not burst. They were denominated CP1, CP2, CP3 and CP4 respectively.

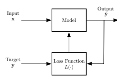


Figure 2.1: Pressure and Crack Dimension for Test CPX

A sensor array was then disposed on top of the naked steel and the rubber cape (Figure 2.2) throughout the tube's length. Immediately surrounding the crack a couple of ultrasound sensors were placed to measure its propagation throughout the test, those work based on Time-of-flight diffraction ultrasonics (TOFD) [4], [5] and are named with respect to the technique (TOFD sensors).

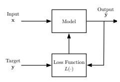


Figure 2.2: Sensor Array Disposition for Test CP3

Three different streaming sensors (Figure 2.2) were used, the R1.5 [6], R15[7] and WD [8] sensors. Their inner workings and configuration (including conditioning circuits) will not be discussed in this work. It is important to note however, that they work on different frequency ranges (Table **??**) and naturally have diverse responses.

Table 2.1: Streaming Sensors Frequency Range.

Sensor	Frequency Range (kHz)
R1.5	5 - 20
R15	50 - 400
WD	100 - 900

These sensors were then sampled using a (not specified) 16 bit Analogue-to-Digital Converter (ADC) at $2.5GHz$ during the whole test, their data was then put into a special formatted file created by National Instruments (NI), the Technical Data Management Streaming (TDMS) file [9].

2.2 Streaming Raw Data

Normally, these files can be opened using some specific NI software like *DIAdem* but the LAMEF engineers made slight modifications (Figure 2.3) when saving the ADC data, using a 32*bit* word (normally holding one sample) made of two concatenated 16*bit* integers (that came from the ADC), thus effectively doubling the amount of data stored without requiring additional space.

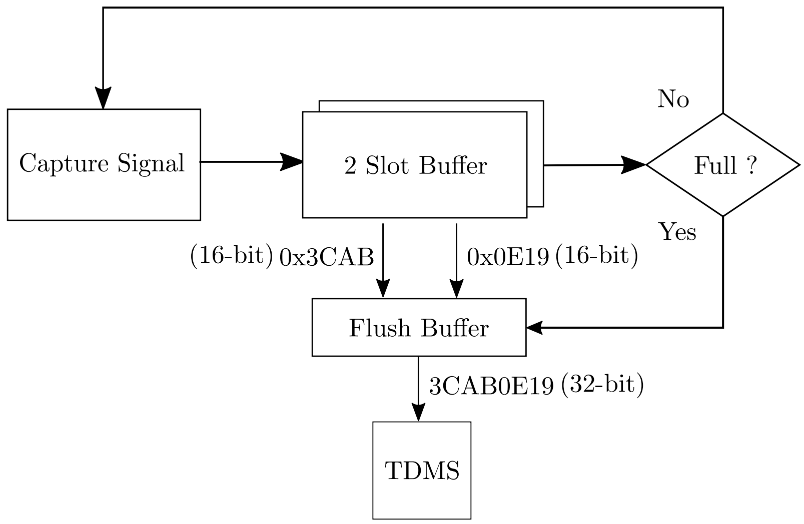


Figure 2.3: Low-Level Modifications Done to Signal Acquisition.

In order to read the TDMS files, LAMEF provided a special compiled LABView routine that transforms each TDMS file to a binary one and a MATLAB script that loads the file to memory. Out of all the tests, only the first one (CP1) did not burst, even after 5 filling cycles, since identifying the burst is essential (Section **??**) all CP1 data was discarded and it will not be discussed any further.

A single TDMS file translates to roughly 6.7 seconds (Figure 2.4), containing 2^{24} samples taken from its 16 different sensors (channels) leading to a $2^{24} \times 16$ integer (16-bit) matrix. In sum (Table **??**), all data came from (eventually) ruptured ducts and theoretically contain useful data.

Table 2.2: Hydrostatic Test Summary

Test	Date	Pre-Crack Depth (millimetres)	Rupture Cycle	Pressure at Rupture (bar)	Data Volume (Gigabytes)	Files
CP1	18/03/2015	5.5	X	-	3900	X
CP2	28/07/2015	7.9	1	230	900	1513
CP3	06/11/2015	6.6	1	264	900	1500
CP4	23/05/2016	7.0	2	283	1500	4261

(a) TDMS File 900, Channel 12. (b) TDMS File 900, Channel 7.

Figure 2.4: Data from TDMS File 900, Both Channel 12 (a) and 7 (b) From Test CP3 with Zero (0) Mean.

2.3 Preprocessing

The preprocessing was likely the lengthiest part of this thesis, it is mainly divided in 3 big blocks, an initial resolution analysis (Section 2.3.1) to determine which files contain useful information, a TOFD signal removal (Section 2.3.2) and a final identification and cleaning of the pressure bomb AE signal (Section 2.3.3).

These three stages are invaluable for treating the enormous amount of data ($\approx$ 1TB per test) and coming up with a reliable and efficient way to extract AE from the propagating crack (Section 2.4).

2.3.1 Resolution Analysis

Upon first inspection, not all files seemed to contain relevant data (Figure 2.4a), most of them contained only a noise bar (Figure 2.4b) thus creating the first problem, determining which files (and channels) contained relevant information so that time would not be wasted trying to find AE waves immersed on noise.

One simple way to determine digital signal quality is through its resolution, signals that are encoded using 10 bits (therefore 1024 digital levels) should have more valuable information than signals encoded with, for instance, 2 bits (therefore 4 digital levels). Using that as a base, it was heuristically stipulated that 8 bits would be the bare minimum (since it gives approximately 1% of the ADC's dynamic range).

The count of digital levels and effective bits used to encode each file and channel was calculated (Figure 2.5) and only those that had a minimum of 8 bits were considered good enough to be further inspected.

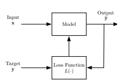
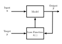


Figure 2.5: Sensor Array Disposition for Test CP3

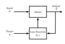
2.3.2 TOFD Removal

TOFD was not a concern in previous works (Section **??**) since it was not evident (even though it can be seen analysing the parameter data). This however, is not true when working with Streaming data, the TOFD signal is clear and has a well defined period and also comes in blocks (Figure 2.6).

Each block consist of 5 AEs separated by $20ms$ (Figure 2.6b) and each block is 1 second away from each other (Figure 2.6a). These well defined periods can then be used to completely remove all TOFD signals from the entire test.



(a) TDMS File 900, Channel 12.



(b) TDMS File 900, Channel 12 Focused on a Single TOFD Block

Figure 2.6: Data from TDMS File 900, Highlighting the TOFD Characteristics.

To remove those, it is necessary to first identify it, without going into much detail (those are in Section 2.4) a threshold level is defined for each file/channel combination and each point is compared to that threshold, if its amplitude surpasses it, it receives a 1 (TRUE) and 0 (FALSE) if not (Figure 2.7).

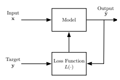


Figure 2.7: Data from TDMS File 900, Channel 12 Containing the Threshold and Indexes that Respectively Surpass It.

Once the beginning of each TOFD wave is found, a simple check is done to verify if they are separated by $20ms + -10\%$. If so, their indexes are saved and the waves are thus identified. In order to allow a finer control over the TOFD block removal, an additional number of samples is taken from both before and after each wave (Figure 2.8). A total of 50000 samples (25000 for each side) was chosen to guarantee the entire block removal, the downside is that any AE that happens to fall inside a TOFD block is also lost, however, considering that each block takes $100ms$ each second, that equates to 10% of data being discarded, which is quite acceptable.

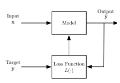


Figure 2.8: Removal of Time-Of-Flight-Diffraction Sensor Signal.

2.3.3 Pressure Bomb Removal

Noise from the pressure bomb was also not considered in previous works (Section 1.6) and it is an indispensable part of the test, used to fill and pressure the duct. This was discovered by accident when, by analysing multiple files at once, a huge unknown (and somewhat periodic) signal appeared (Figure 2.9).

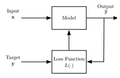


Figure 2.9: Data From TDMS Files 750-770, Channel 7.

Once crossing the files in which that signal appeared with the logs provided by the LAMEF engineers, it was possible to infer that these signals only appeared at the beginning and end of pressure rising, thus concluding that it is indeed a AE signal generated by the bomb and not the crack propagating.

Its removal was done manually, all 7000 more files were checked 10 at a time (a total of more than 700 different plots) and all files that had even a portion of this signal were discarded.

2.4 Wave Capture

Both Section 2.3 and this one contain all that was done in order to transform over 3 Terabytes of noise filled data into a usable, clean and expandable database containing useful acoustic emission data.

It begins by defining a *noise level* for each file/channel combination and using this information to create a floating threshold level that works similar to that described in Section 2.3.2, once that threshold is exceed (both positive or negatively) the beginning of a "hit" is defined, and specific timing parameters (Section 2.4.2) are used to determine its end, thus extracting individual AE waves.

For each of those waves an array of time-domain parameters (Section 2.4.3) is calculated alongside its power spectrum (Section 2.4.4). Both are then processed (Section 2.6) again to be used as input for the neural network.

2.4.1 Estimating Noise Level

Normally, AE tests use industrial equipment with a fixed Threshold (a level that when exceeded triggers the wave capture) which may or may not be changed throughout the test by its technician. However, instead of trying to define a fixed level for each channel and file, a fluctuating limit was created based on the level of noise for each file/channel. This is interesting since the signal to noise ratio (SNR) can (and does) change a lot because of the varying amplitude of the AE signals (Section **??**). Using a volatile limit based on the noise level itself is an attempt to stabilize the SNR for each captured wave.

It is based on a simple metric of standard deviation (Equation 2.1) applied to each file and channel. In other words, suppose the data from one channel, a $2^{24} \times 1$ array, knowing that AE is a relatively rare event and that there exists a bar of white noise (Figure 2.4), it is safe to say that this array is mostly composed of Gaussian white noise.

$$\sigma_X = \sqrt{E[(X - \mu_X)^2]} = \sqrt{\frac{1}{N}\sum_{1}^{N}(x_i - \mu_X)^2} \tag{2.1}$$

Where X is the array of size $N = 2^{24}$, σ_X is its standard deviation, μ_X its mean and x_i the ith array element. Considering that this is composed mostly of Gaussian white noise, it is possible to state:

$$Noise_X \approx 3\sigma_X \tag{2.2}$$

This provides a pretty good noise level estimate (Figure 2.10) that is then (heuristically) amplified by a factor of 3 to create a fluctuating threshold able to isolate acoustic emissions.

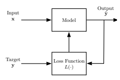


Figure 2.10: Data From TDMS File 900, Channel 7 Highlighting the Noise Level

2.4.2 Timing Parameters

Defining the threshold is but the first step in completely extracting AE information from the data. The DISP16C (used on professional AE tests) user's manual [10] specifies 3 timing parameters used to capture AE, these are the Hit Definition Time (HDT), Hit Lockout Time (HLT) and Peak Definition Time (PDT). However, PDT is used to determine which peak is the highest (important for calculating AE parameters) while capturing the wave at real time. Since this work uses static data, determining the highest peak makes no sense because there is only one highest peak for each hit (the maximum amplitude), therefore its definition will not be covered in this thesis, if needed, the reader can find it at [10].

HDT is the most important parameter and is used to define and fix the end of the "hit". Suppose a signal that exceeds the threshold at a certain time t_0, once HDT seconds has elapsed while the signal had no threshold crossings (stayed under the threshold) a hit and its end are defined. If one sets this parameter too high, adjacent events may appear as a single hit and if set too low, a single AE may be separated into multiple hits.

HLT can be interpreted as a "dead time" after the definition of a hit, it is a slot of time where no hit can begin even if the signal exceeds the threshold. It is used to eliminate lower amplitude echoes that can occur from the AE's reflection throughout the system. It used to have more importance in the 1980's when computational power was rather limited [10].

The tuning of these hyper-parameters is not studied in this thesis since the script made to capture the waves takes around 8 hours for each CP, so it would be extremely time-consuming and deviate from the main objective of this work, although the author would highly recommend this analysis. Those parameters are then defined as suggested by the LAMEF engineers:

- HDT: $1000\mu s$
- HLT: $800\mu s$
- PDT: N/A

This concludes all necessary steps to extract useful

2.4.3 Acoustic Emission Parameters

2.4.4 Frequency Data

2.5 Database Structure

2.6 Model Definition

2.6.1 Network Size

2.6.2 *Transition Time Estimation*

2.6.3 Relevance Analysis

2.6.4 Relevance Analysis

Bibliography

[1] WIKIMEDIA COMMONS. "Neuron". https://upload.wikimedia.org/wikipedia/commons/b/b5/ dez. 2006.

[2] "The World Factbook — Central Intelligence Agency". https://www.cia.gov/library/publications/the-world-factbook/fields/2117.html, 2016.

[3] TURING, A. M. "I.—COMPUTING MACHINERY AND INTELLIGENCE", *Mind*, v. LIX, n. 236, pp. 433–460, 1950. ISSN: 0026-4423, 1460-2113. doi: 10.1093/mind/LIX.236.433.

[4] CHARLESWORTH, J. P., TEMPLE, J. A. G. *Engineering Applications of Ultrasonic Time-of-Flight Diffraction*. Research Studies Press, 2001. ISBN: 978-0-86380-239-3.

[5] SILK, M. G., LIDINGTON, B. H. "The Potential of Scattered or Diffracted Ultrasound in the Determination of Crack Depth", *Non-Destructive Testing*, v. 8, n. 3, pp. 146–151, jun. 1975. ISSN: 0029-1021. doi: 10.1016/0029-1021(75)90024-9.

[6] PHYSICAL ACOUSTICS. "R1.5 - 5-20 kHz Very Low Frequency AE Sensor". https://www.physicalacoustics.com/by-product/sensors/R1.5-5-20-kHz-Very-Low-Frequency-AE-Sensor, .

[7] PHYSICAL ACOUSTICS. "R15I-AST - 150 kHz Integral Preamp AE Sensor". https://www.physicalacoustics.com/by-product/sensors/R15I-AST-150-kHz-Integral-Preamp-AE-Sensor, .

[8] PHYSICAL ACOUSTICS. "WD - 100-900 kHz Wideband Differential AE Sensor". https://www.physicalacoustics.com/by-product/sensors/WD-100-900-kHz-Wideband-Differential-AE-Sensor, .

[9] NATIONAL INSTRUMENTS. "The NI TDMS File Format". .

[10] MISTRAS GROUP. "DiSP with AEwin USER'S MANUAL". jul. 2011.