Fault Classification of Bearings Using Machine Learning Algorithms

by

Vasco Torgal Dias Duarte

Supervisor: Prof. Dr. Joaquim Gabriel Magalhães Mendes

Dissertation presented to

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

in partial fulfilment for the Degree of

Master of Science in Mechanical Engineering – Automation

Porto, June 2017

This page was intentionally left blank.

Abstract

In a world where machinery has a primary role in the productivity of most industrial companies, predictive maintenance is crucial. Also, considering that bearings are present in the majority of industrial machinery, most of the monitoring, in one way or another, has a central focus on these elements.

At the same time, we find ourselves undergoing an industrial revolution – Industry 4.0 –, where low cost distributed monitoring and data transfer are vital. Considering these goals, which directly translate to a large amount of data generated, it is no surprise that the implementation of Artificial Intelligence, more specifically Machine Learning algorithms, is gaining growing importance when it comes to data analysis in real-time.

With this in mind, the development of this dissertation tries to present a practical case where both, monitoring of bearings and Artificial Intelligence, can be applied, always considering the main theoretical concepts of each topic: the range of frequencies to study when analyzing the bearings, the kind of equipment to measure and acquire data, as well as the algorithm to use in order to train the Machine Learning model – in this case, an Artificial Neural Network.

This project uses bearings of anemometers during data acquisition, both inside a wind tunnel as well as in an acoustically-isolated environment. Relative to the data analysis, most of the work was done using LabVIEW in junction with MATLAB scripts. The Virtual Instruments created enabled the data analysis and visualization, which allowed proper parameter selection in order to implement an Artificial Neural Network.

The final result was a trained Artificial Neural Network model capable to distinguish faulty bearings from healthy ones, under certain limitations due to the small size of the available knowledge. It was left for future works the use of different data acquisition methods for real-time classification on the field, as well as the application of this model to Acoustic Emission data, which allows for better predictive maintenance. It is also expected that this model has the ability to classify different flaws, given better acquisition conditions.

This page was intentionally left blank.

Resumo

Num mundo onde as máquinas têm um papel importantíssimo na produtividade de muitas empresas industriais, a manutenção preditiva tem um papel crucial. Considerando ainda que os rolamentos estão presentes na maioria de máquinas industriais, a monitorização, de uma maneira ou de outra, tem um foco central nestes elementos.

Ao mesmo tempo, encontramo-nos hoje numa revolução industrial – Indústria 4.0 –, na qual monitorização distribuída de baixo custo e transferência de dados são fundamentais. Considerando estes objetivos, que se traduzem diretamente em grandes quantidades de dados gerados, não é surpresa que a introdução de Inteligência Artificial, mais especificamente algoritmos de aprendizagem automática, esteja a ganhar importância em campos como a análise de dados em tempo real.

Com isto em mente, o desenvolvimento desta dissertação procura apresentar um caso prático onde a monitorização de rolamentos e a inteligência artificial possam ser aplicados, tendo sempre como base os conceitos teóricos de ambos os tópicos: as frequências mais relevantes durante a análise dos rolamentos, o tipo de equipamento utilizar para medição e aquisição de dados, assim como o algoritmo escolher para treinar o modelo de aprendizagem automática – neste caso, uma rede neuronal artificial.

Neste projeto foram utilizados rolamentos de anemómetros durante a aquisição de dados, tanto num túnel de vento como num ambiente acusticamente isolado. Relativamente ao tratamento de dados e desenvolvimento da rede neuronal, a maioria do trabalho foi desenvolvido utilizando LabVIEW em conjunto com scripts de MATLAB. Os Instrumentos Virtuais criados permitem a análise e visualização dos dados adquiridos, o que por sua vez leva a uma seleção apropriada dos parâmetros a utilizar para implementar uma rede neuronal artificial.

O resultado final foi um modelo treinado de uma rede neuronal artificial capaz de distinguir rolamentos defeituosos de rolamentos em bom estado, sob certas restrições devido ao tamanho reduzido da base de dados disponível. Para trabalhos futuros foram deixados o desenvolvimento de métodos de aquisição de dados mais adequados para a classificação de anemómetros em tempo real no terreno, assim como a aplicação deste modelo em dados de emissão acústica, que permite uma melhor manutenção predictiva. Com melhores condições de aquisição de dados é ainda expectável que este modelo consiga classificar diferentes falhas de rolamentos.

This page was intentionally left blank.

Acknowledgments

To Prof. Dr. Joaquim Gabriel, suppervisor of this dissertation, I thank the opportunity to develop this work on a topic of such interest to me.

To the LAC department, from INEGI, for all the equipment and facilites available, with a special mention to Luís Mendes and Eng. Miguel Marques, whose support throughout the whole semester was invaluable, even during those times where everything seemed to go wrong.

To Prof. Dr. Jorge Seabra and all the CETRIB department, with a special thanks to Dr. Ramiro Martins, for the encouragement and help given.

To my colleagues in the room L003, with whom I had the opportunity to share ideas, opinions and, the most important, good memories.

To my friends, for all the coffee breaks, all the dinners, all the meetings, as well as everything else.

To my family, for all the support given, not only during these months, but also during all these years.

To Sara, for everything.

This page was intentionally left blank.

Contents

List of Figures

List of Tables

1 Introduction

It is no novelty that technological progress has been increasing exponentially for the last half century [1]. With this growth, the opportunities to apply new computational technologies to every work field becomes a reality. On this light, one of the main purposes of this dissertation comes as a means to gain knowledge on one of the fields with the biggest potential of computer science: Artificial Intelligence; more specifically, the goal is to apply Machine Learning to data analysis.

At the same time, the industrial world has a very clear focus: improve the productivity of processes by maximizing machinery uptime and minimizing the time spent on maintenances. As such, even if the maintenance cycles of most machinery are provided by the manufacturer, a growing focus on condition monitoring allow predictive maintenance. Not only that, but most machinery has one major component in common: bearings. This causes most of the monitoring to focus on bearings. However, to perform an accurate monitorization, one must fully understand data acquisition and data analysis principles.

With these subjects in mind, this dissertation tries to shed a light on how to apply Machine Learning algorithms to develop a program capable of analyzing data and classify it. Given the potential of the possibility above, this program will focus on trying to classify bearings according to whether they are healthy or faulty. To do so, an analysis on the fundamental fault frequencies of the bearings was done, along with a study to different methods of analyzing these frequencies, in order to filter the data according to its relevance to the classification of the bearings. The data acquisition was performed on anemometers with two bearings.

To achieve this goal, different data measurement devices were studied: microphones, accelerometers and acoustic emission sensors. However, only the first two were used for actual data acquisition, due to the cost of the last one. Along with these sensors, proper data acquisition and processing methods were studied, and performed with the help of equipment such as a DAQ device and a signal conditioner.

Lastly, after a proper database was acquired, suitable parameters were selected to train a Machine Learning algorithm. In this case, an Artificial Neural Network. An initial model capable of detecting different types of waveforms was programmed in LabVIEW, and, after proving it to be successful, it was adapted to detect the condition of bearings from anemometers. After training multiple models, the most accurate one was implemented, resulting in somewhat accurate results, with more than 75% accuracy when classifying a bearing. However, no meaningful conclusions can be extrapolated to wider samples, since the database used for the development of this project was too small, with only one sample of a reference anemometer without noise.

Nevertheless, this dissertation opens the possibility to perform bearing monitoring on the field, which improves the ability to implement predictive maintenance on the anemometers, instead of strictly following the indications of the manufacturers, given a larger database to work with.

1.1 Main Goal

The main goal of this dissertation is to develop an automated system capable of classifying bearings according to whether they are faulty or not. To reach this goal, the following objectives were defined:

1. Study the main differences between healthy and faulty bearings, and how to detect them;
2. Study the basic principles of different applicable data measurement devices, such as microphones, accelerometers and acoustic emission sensors;
3. Apprehend solid knowledge on how to work with LabVIEW, more specifically the data acquisition and signal processing toolkits;
4. Develop a data acquisition system capable of extract waveforms with meaningful data regarding the bearings' condition;
5. Study the concepts of Artificial Intelligence and Machine Learning;
6. Design a functional Artificial Neural Network with Back-propagation;
7. Development of an Artificial Neural Network which takes as inputs selected features from the acquired waveforms and outputs the correct state of the tested bearing.

1.2 Structure

Chapter 2 offers a theoretical approach to the state of the art of the areas necessary for the development of this dissertation. First, an analysis to bearings – the main mechanical component of this project – is made, focusing more specifically on the most commonly used monitoring methods in order to obtain, as well as on the fault frequencies on which bearing faults can be detected; this is followed a study on data acquisition methods, starting with different types of sensors along with the data acquisition devices to them associated, as well as proper methods of data processing and conditioning; finally, this chapter is closed with an approach to Artificial Intelligence and Machine Learning, both core aspects on this project.

In Chapter 3 one can understand how to implement basic data analysis features on LabVIEW, along with a detailed approach to Artificial Neural Networks – the algorithm chosen to develop during this dissertation. As such, in this chapter is developed of an Artificial Neural Network model on LabVIEW, with the goal of classifying different types of waveforms. This model will be crucial for the final goal of this dissertation.

In Chapter 4 is made an analysis to the properties of the bearings to study, focusing on their geometry, followed by the selection of the equipment, as well as a description of the different data acquisition conditions used throughout the project. Considering the bearings are from anemometers, two of the situations are inside of a wind tunnel, whereas the last one is done inside an acoustically isolated environment.

Chapter 5 presents the results of the multiple data acquisitions, including the implementation of an Artificial Neural Network on the data acquired with an accelerometer – the only one where the measurements can be repeated in the same conditions with considerable reliability. A first project for a soundproof box for data acquisition is also presented in this chapter. It presents an analysis to the results obtained from multiple trained models, culminating in the selection of the one with the best results for further data analysis. This chapter finishes with a critical analysis to the results obtained with the implementation of the algorithm, which mostly comes down to the small size of knowledge available for training.

Finally, Chapter 6 summarizes and discloses the final conclusions of this dissertation at the same time as it presents ideas for future works.

2 State of the Art

This chapter focuses mainly on three themes. The first one presents a theoretical analysis to bearings, their importance in modern industry and the most common types of monitoring; the second theme is centred on data acquisition, studying different sensors, data acquisition devices and data processing methods; finally, the third one focuses on providing an insight on the basic concepts of Machine Learning.

2.1 Bearings

A bearing is an element designed to reduce friction between two moving parts – consequently reducing heat generation –, by allowing a middle component (usually metal spheres or rollers) to roll in between while being subjected to a certain load, axial and/or radial [2].

Regarding the type of bearing, the most common one is the rolling-element bearing (Figure 2.1). In fact, rolling-element bearings are present in more than 90% of the rotating machinery found in both commercial and industrial applications [3]. Given the importance of these components, it is necessary to have the best control possible of multiple variables that influence their performance. These factors include, but are not restricted to, the proper mounting of the bearing, the lubricant, pressure, temperature, among others.

Figure 2.1 - Rolling element bearing [4]

2.1.1 Monitoring and Predictive Maintenance

On one hand, since the influence of many of the factors enumerated above are hard to simulate on laboratory conditions, there is a need of monitoring and maintaining the bearings throughout their operational life. On the other hand, it is incommensurable to demount the machines where the bearings are allocated on a periodical basis for maintenance – primarily due to losses in productivity. As such, monitoring methods are used to get estimations on the bearing's operational life, which allows for predictive maintenance: the ability to predict the optimal time to perform maintenance of a certain component or machine. Predictive maintenance is responsible for results such as [5]:

- 12% less expense on scheduled maintenances;
- 30% lower maintenance costs;
- 70% fewer breakdowns.

The most common monitoring methods are shown in Figure 2.2. By analyzing this figure, it is easy to understand that methods such as visual, physical and monitoring are not advisable for predictive monitoring due to their low future-maintenance value: a bearing in which a fault can be detected by these techniques is usually already beyond repair, but due to their simplicity they are the most used method throughout the years for basic observations. Not only that, but thanks to additional equipment, it is possible to amplify these observations, increasing their accuracy and predictive potential [6].

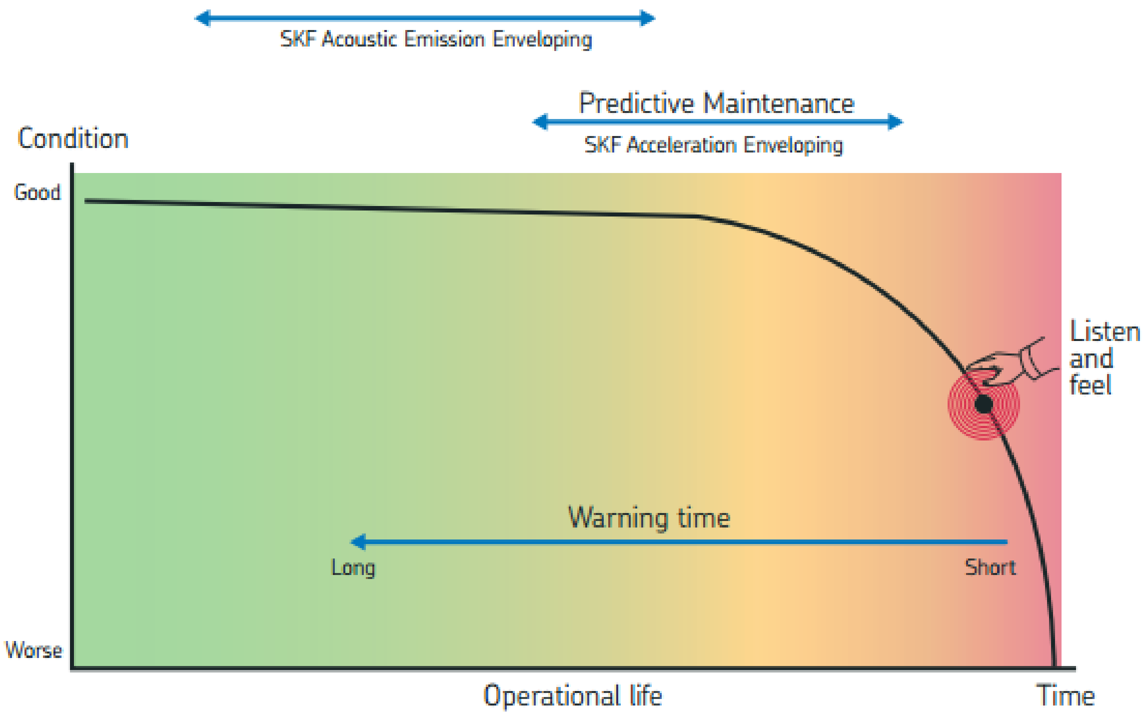


Figure 2.2 - SKF™ schematics on different monitoring methods according to bearing operational life [7]

Visual monitoring allows the detection of the level of corrosion and deterioration of components, as well as the observation of the lubricant level. The inclusion of sight-gauge in machines and the usage of products such as thermal imaging guns (Figure 2.3) or endoscopes increase the monitoring value of this technique.

Figure 2.3 - Thermal imaging gun [6]

Audible monitoring enables the detection of bearing damage through noises. However, without proper equipment this results in one of two cases: the bearing's noise is masked by the environmental noise or the bearing is already so damaged that the predictive value is null, as the machine is most likely already undergoing mechanical failure. Yet, with the aid of stethoscopes (Figure 2.4) or ultrasonic probes, sound detection can be much more accurately directed to the monitored bearing's position and frequency range.

Figure 2.4 - Audible monitoring with an electronic stethoscope [8]

Also, using methods such as Spectral Subtraction (Figure 2.5), a substantial amount of noise can be eliminated. This method requires two data measurement devices (such as two microphones), one directed at the monitored bearing, while the other used as an environmental reference. Then, by subtracting both the spectral signals, a noise-filtered signal is obtained.

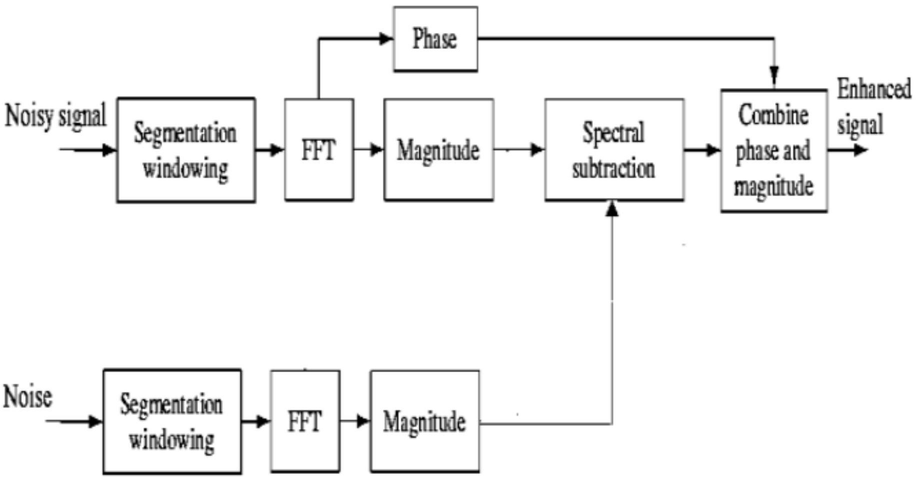


Figure 2.5 - Spectral Subtraction method block diagram [9]

Regarding physical monitoring, it is the most useful method when it comes to sensorial monitoring and can be used to monitor temperature, vibration and lubrication. Just by touching the monitored machine, given that there is a performance data history, one can detect qualitative variations. For more accurate monitoring however, multiple products can be used to get quantitative measurements, from thermometers to fully programmed machine condition equipment (Figure 2.6).

Figure 2.6 - SKF™ Machine condition advisor [10]

The most used technique in the past years is the usage of accelerometers to measure the direct vibration frequencies from the bearing. Using accelerometers proves advantageous in multiple points, as it allows for a better predictive maintenance (as shown in Figure 2.2, this method allows for fault detection as soon as approximately 50% of the bearing's operational life), with the ability of detecting multiple kinds of faults. This comes at a cost of dedicated equipment to acquire the data, but it is usually a more than worthy trade-off. The other cost associated to this type of equipment is the considerable expertise necessary to properly filter the relevant data (similarly to audible monitoring, where the correct frequency range should be filtered to obtain more accurate results). To overcome the lack of know-how, there are companies focused on developing software capable of doing this data analysis (Figure 2.7) or even dedicated signal conditioners (Figure 2.8).

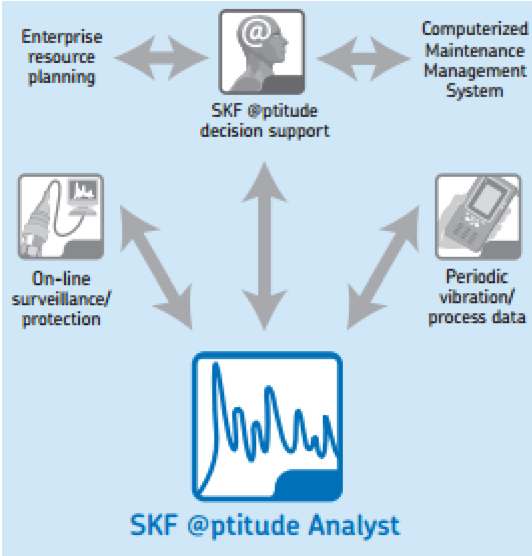


Figure 2.7 - SKF™ *@ptitude Analyst* main functions [11]

Figure 2.8 - PCB Model 682B05 *Bearing Fault Detector* [12]

Nevertheless, companies are now focusing more and more on Acoustic Emission (AE) analysis [13], as it allows for full monitoring throughout the bearing’s operational life, enabling predictive maintenance virtually as soon as a bearing is mounted. AE is based on the analysis of elastic and stress waves at much higher range of frequencies – when compared to vibration analysis. These frequencies range from 30kHz to 1MHz, but most of the energy present due to faults of bearings is around 100-500kHz. It is important to note that this higher frequency range has a significant advantage over vibration analysis, since the normal functioning of the machinery has almost no relevance on this range (mechanical vibrations are not relevant above 20kHz) [14]. Figure 2.9 represents the traditional signal acquisition and preprocessing procedure with an AE sensor.

Acoustic emission also presents multiple other advantages. While vibration analysis on complex machinery must be performed on multiple positions to understand which component has defects (if even that can be detected), on AE analysis the process is much more effortless, as almost only fault-generated signals are detected. This translates in a much lower level of expertise needed to perform this kind of monitoring [15].

Figure 2.9 - Traditional AE Signal Acquisition and Preprocessing [16]

Most of the AE analysis as well as vibration analysis utilizes a technique called "Envelope Detection". This will be explained in the next sub-section.

Finally, considering that industry is embracing the concept of Industry 4.0 to automate the data transfer between all the machinery and control centers [17], the ability to have detailed information about the status of one of the core components of most machinery throughout its whole life is almost a no-brainer, giving rising importance to methods such as AE analysis and to software such as the SKF™ *@ptitude Analyst* (Figure 2.7).

2.1.2 Fundamental Fault Frequencies of Bearings

To proper study the data acquired, independently of the method, a basic notion of the expected frequency peaks when analyzing the frequency spectrum of the data is necessary.

As such, the formulas represented in Equation (1) represent the fundamental fault frequencies of a ball bearing, given its geometry [3].

$$\begin{gathered} BPFI = \frac{N}{2} \times F \times \left(1 + \frac{B}{P} \times cos(\theta)\right) \\ BPFO = \frac{N}{2} \times F \times \left(1 - \frac{B}{P} \times cos(\theta)\right) \\ FTF = \frac{F}{2} \times \left(1 - \frac{B}{P} \times cos(\theta)\right) \\ BPF = \frac{P}{2B} \times F \times \left[(1 - \left(\frac{B}{P} \times cos(\theta)\right)^2\right] \end{gathered} \tag{1}$$

Where,

BPFI - ball pass frequency inner race (Hz),

BPFO - ball pass frequency outer race (Hz),

FTF - fundamental train frequency (Hz),

BPF - ball pass frequency (Hz),

N - number of balls,

F - shaft frequency (Hz),

B - ball diameter (mm),

P - pitch diameter, the diameter measured from the center of two opposing balls (mm), Θ - contact angle, "the angle of contact between a plane perpendicular to the bearing axis and a line joining the two contact points between the ball and the inner and outer raceways" (Figure 2.10) [18].

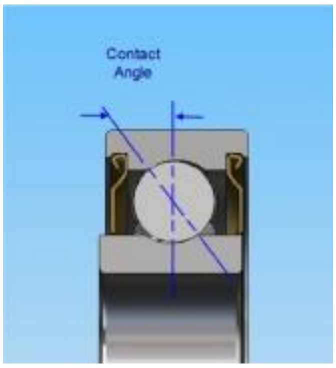


Figure 2.10 - Contact angle [18]

Due to the different noise present in most machineries though, it is usually not possible to filter these fault frequencies from the rest of the signal. Nevertheless, the vibrations resulting from the faults excite the bearings structural resonance, resulting in amplitude modulated signals at higher frequencies. A method to analyze this high frequency signals is called Envelope Detection (Figure 2.11). This process starts by filtering high frequency noises and low frequency signals (Figure 2.11a), resulting in a signal composed only by a burst of high frequencies (Figure 2.11b). This signal is then "enveloped" by using Hilbert Transform (Figure 2.11c). This signal is finally converted to a frequency domain, where the fundamental frequencies can be detected (Figure 2.11d) [19].

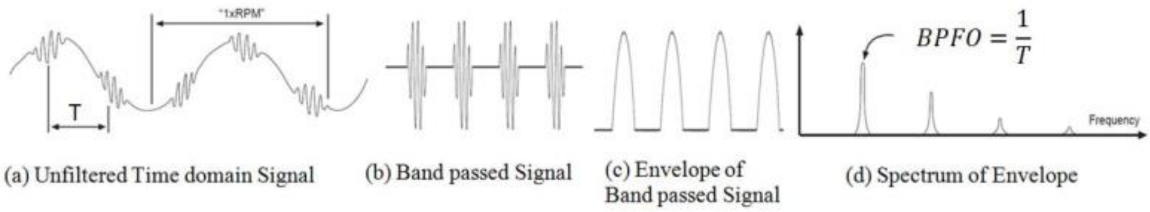


Figure 2.11 - Envelope detection process [19]

Multiple variations of this method include the implementation of wavelets, support vector machines, along with other techniques [20][21].

2.2 Data Acquisition

As seen before, to perform maintenance of bearings, most of the techniques require data measurement devices. The devices chosen for the development of this dissertation consisted on microphones (for analysis in the audible spectrum), accelerometers (for vibrations) and AE sensors (high-frequency elastic/stress waves) [22]. All these devices fundamentally result in some form of electrical signal, usually voltage, current or resistance.

Since the analysis of bearings occurs when they are in rotating, the output takes the shape of a waveform. When performing this kind of data measurements, it is important to remember the *Nyquist* sampling theorem, which states that, to avoid aliasing (misrepresentation of a signal due to under sampling), the sampling frequency must be at least two times higher than the maximum frequency to be measured – the Nyquist frequency.

However, this does not fully prevent under sampling. In fact, to avoid this phenomenon, a good rule of thumb is to sample with a frequency ten times higher than the data's maximum frequency.

When choosing a sensor, one must also take into consideration multiple characteristics, such as [23]:

- Type of output signal (analog or digital)
- Sensitivity (output change per unit of input),
- Range (maximum and minimum values the can be measured),
- Precision (degree of reproducibility of a measurement),
- Resolution (minimum input variation detected by the sensor),
- Offset (difference between the expected output value and the actual output value),
- Nonlinearity (difference between the ideal curve and the actual measured curve),
- Hysteresis (lag between the variation of the input and that same change on the output),
- Response time (time elapsed between the change of an input and that same change on the output)
- Dynamic linearity (sensor's capability to respond to and follow abrupt changes on the input)

The next subsections will discuss in further detail the types of data measurement devices mentioned above, followed by data acquisition devices and data processing methods

2.2.1 Microphones

A microphone is a device used to measure sound pressure waves, converting those same pressure waves into electrical signals. Since the human ear frequency range is between 20 and 20kHz, a microphone usually has a sampling frequency of 44.1kHz, which allows proper aliasing of data in all human audible range.

Regarding the type of microphone according to their construction and acquisition mode, we can make separate them in the following classes [24]:

- Dynamic,
- Ribbon,
- Condenser (which includes the MEMS – usually found in phones – and Electret variations),
- Piezoelectric,
- Others (liquid, carbon, fiber optic, laser, …).

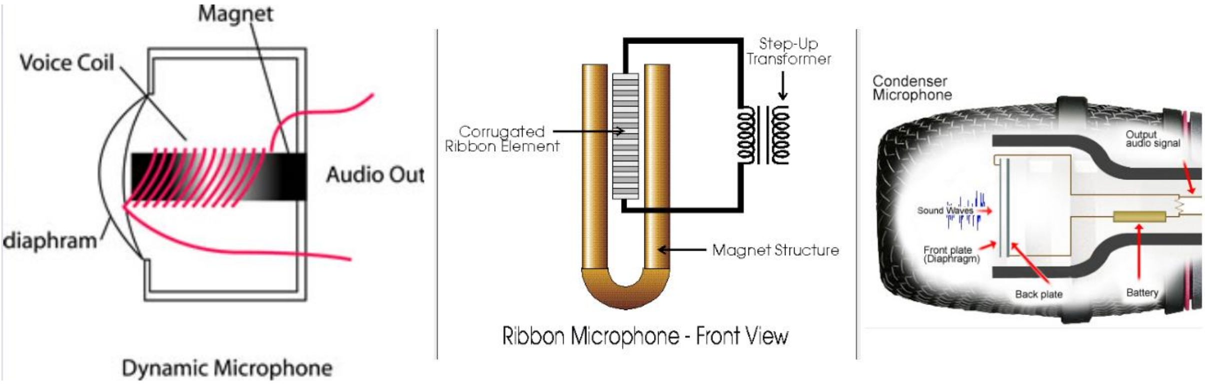


Figure 2.12 - Different types of microphones: dynamic (left), ribbon (center) and condenser (right) [24]

A microphone can also be distinguished according to their directionality [25]:

- Unidirectional (most common type),
- Bidirectional (used when two people are talking face to face in radio shows),
- Omnidirectional (captures sound in the same way all around).

Usually, when working with computers, the data acquisition can be done directly with a sound card. The same goes for mobile phones, with the particularity of the audio format in which the data is saved being different: while on a computer the predefined audio type is .wav (uncompressed data), which preserves better audio quality, a smartphone saves the data in a .mp3 file (compressed data), resulting in data loss.

To standardize the type of files used during data analysis, the data will be converted from .mp3 to .wav when using a smartphone.

2.2.2 Accelerometers

An accelerometer is a device used to measure proper acceleration (the acceleration of its moving parts in relation to rest of its body – at rest). These devices have multiple applications, some as simple as measuring gravity, detecting if a certain object is in upright position or, as in this study, to measure vibrations on machinery.

According to the application in mind, one can choose a single-axis accelerometer versus multi-axis accelerometers. Also, according to the functioning method, most of the existing accelerometers are either capacitive, piezoresistive or piezoelectric [26].

Unlike microphones, which already have a dedicated soundcard in every computer nowadays to treat the data, accelerometers require a dedicated DAQ device (see subsection 2.2.4) so that data can be read from the accelerometer and written to a computer. To achieve this purpose, the DAQ device must be configured to work at the same sampling rate as the dynamic response of the accelerometer. Also, it is important to properly preprocess the signal. This is usually done with the help of signal conditioners and/or preamplifiers.

2.2.3 Acoustic Emission Sensors

Acoustic Emissions correspond to stress waves which result from "sudden internal stress redistribution of the materials caused by the changes in the internal structure" [27]. These waves can have multiple causes, being most of them related to damaged components. As such, the study of such waves enables a qualitative analysis to the damages of a certain structure since their very beginning. To do so, AE sensors are used, acquiring waves within frequency ranges from 30kHz to 1MHz [28], which prevents an overlap with mechanical vibrational waves, meaning that AE is virtually undisturbed by mechanical noises other than faults (such as imbalance and misalignments) [14].

These sensors can also be split into multiple classes [29]:

- Piezoelectric,
- Resonant,
- Wide Band,
- Differential,
- Capacitive.

In a similar fashion to accelerometers, AE sensors also require a DAQ device. The expected difference between the DAQ devices used for accelerometers and for AE sensors resides mainly in the sampling rates, which are much higher in AE analysis, and on the amplification rates, which are again much higher in AE analysis.

2.2.4 Data Acquisition Devices

While a sensor measures a physical phenomenon, converting it to an electric signal, the purpose of data acquisition (DAQ) device is often to act as an interface between a sensor and a computer, having the following main functionalities:

- Signal acquisition per se;
- Analog-to-Digital (ADC) conversion;
- Amplification and filtering;

Data acquisition is often incomplete if the signal is not properly conditioned. Table 1 summarizes the most used types of signal conditioning according to the sensor used.

Further down the road, when the signal is already being analyzed on a computer with proper software (such as MATLAB and LabVIEW), other functionalities are used, such as:

- **Filters:** filter certain frequencies, while trying to keep the passband frequency response intact;
- **Fast Fourier's Transform (FFT):** convert a signal from its time domain to a frequency domain, or vice versa with the inverse FFT;
- **Data splitting:** split the original data in smaller time intervals, given that these intervals are larger than the inverse of the smallest frequency to be analyzed, providing more data samples;
- **Parameter selection:** analyze the waveform data to extract properties such as mean, Root Mean Square (RMS), standard deviation, kurtosis, skewness, etc.

With this kind of analysis, providing the data measurement and acquisition are optimal, it is possible to proceed to the implementation of Machine Learning Algorithms to study the data with more accuracy and detail.

Table 1 - Signal conditioning for different sensors [30]

	Amplification	Attenuation	Isolation	Filtering	Excitation	Linearization	CJC	Bridge Completion
Thermocouple	x			x		x	x	
Thermistor	x			x	x	x		
RTD	x			x	x	x		
Strain Gage	x			x	x	x		x
Load, Pressure, Torque (mV/V, 4-20mA)	x			x	x	x		
	x			x	x	x		
Accelerometer	x			x	x	x		
Microphone	x			x	x	x		
Proximity Probe	x			x	x	x		
LVDT/RVDT	x			x	x	x		
High Voltage		x	x					

2.3 Artificial Intelligence and Machine Learning

The concept of Artificial Intelligence (AI) appears from the attempt to understand and reproduce intelligence entities. As such, it is possible to define AI as the science that focus on developing machines capable of showing intelligence [31]. There are multiple applications nowadays which require the usage of AI, such as self-driving cars, speech recognition, computer vision, among others.

Considering the increasing amount of *big data* – everyday 2.5 quintillion bytes of data are created, resulting in 90% of the data in the world today having been created in the last two years [32] –, methods are needed to process these gigantic amounts of data. Since it is unconceivable to analyse this data by hand, automated data analysis becomes a must.

This is where Machine Learning (ML) comes in: ML is one of the core aspects for the functioning of AI, known as the ability of a machine to learn and adapt from any given data without being explicitly programmed: “A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.” [33]. This results in automatic pattern detection on the tested data, which then enables future predictions from those same detected patterns [34].

To properly understand ML, the basic concepts of Task, Performance and Experience should be apprehended. However, it is important to first understand the steps of learning in ML. Machine Learning is performed using Learning Algorithms. There are hundreds of algorithms available, so knowing how to choose the best one is often difficult, not only because of the diversity, but also due to sometimes it is not possible to find an optimal algorithm, and trade-offs must be made. Nevertheless, the selection process can be split into three components: Representation, Evaluation, and Optimization [35]. Table 2 shows multiple examples of these components.

Table 2 - The three components of learning algorithms [35]

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

- **Representation**: in ML, one must choose which data parameters to select for a proper data analysis: the parameters must be chosen having in mind minimizing the number of parameters as well as being able to properly detect data patterns; then, according to the type and number of parameters, different algorithms should be selected for data representation;
- **Evaluation**: evaluation functions should be used to check the accuracy of an algorithm, enabling distinction between good and bad algorithms. This concept is identical to ML Performance, which will be discussed in more detail in subsection 2.3.2;
- **Optimization**: while evaluation is used to distinguish different algorithms, optimization is the method used during algorithm learning for that same algorithm to produce the best results possible, enabling a proper evaluation.

2.3.1 Machine Learning Tasks

Considering the definition of "task", learning is not a task. Instead, learning is the "means of attaining the ability to perform the task" [37] (e.g. when making a robot walk, walking is the task, which can be implemented by making the robot learning how to walk or by programming all the steps needed to walk).

In ML, a task is the way of a ML system processing features from the object we want to learn from (such as values of pixels from an image, or parameters of a waveform). The most common ML tasks are the following [37]:

- **Classification:** given an input $\boldsymbol{x}$, the program must develop a function that outputs a vector $\boldsymbol{y}$ of values "0" and "1", corresponding to the classes to which the tested data set belongs to;
- **Classification with missing inputs:** similar to classification tasks, but the program must develop multiple functions to classify different subsets of $\boldsymbol{x}$, according to the missing inputs. More exactly, if $\boldsymbol{x}$ has n components, the program must develop 2^n functions;
- **Regression:** predict a numerical value output given an input $\boldsymbol{x}$;
- **Transcription:** given a non-textual type of data, transcribe into a discrete, textual form (e.g. speech recognition, or image-text recognition);

- **Machine translation:** translating text from one language to another, for example, gaining importance when analyzing linguistic slang, or popular sayings;
- **Structured output:** a task that englobes multiple other (such as transcription and translation), outputting data with relationships to the input;
- **Anomaly detection:** with some similarities with classification, when given multiple inputs flags some of them as abnormal;
- **Synthesis and sampling:** generates new samples of data similar to the training data (often used in video games to generate textures);
- **Imputation of missing values:** given an input with missing values, the program tries to predict possible values;
- **Denoising:** given a corrupted input, the program tries to predict the clean version of the same input;
- **Density/Probability mass function estimation:** given a set of inputs, the program is expected to output a function p_d, which can be seen as a probabilistic density/mass function (whether the inputs are continuous/discrete), being then able to predict which events are more or less likely to occur. Most of the other tasks require the program to implicitly define the probability function of the inputs in order to perform their main task correctly.

2.3.2 Machine Learning Performance

As said before, Machine Learning Performance and the concept of Learning Evaluation are identical: it is a quantitative measure of how well a certain algorithm is learning. This measure usually varies according to the algorithm chosen, unless we are comparing algorithms, in which case, for equality reasons, the measure must be the same.

Since the purpose of a performance measure is to evaluate the algorithm after the learning process has been finalized, the data set subjected to the measurement is not the training set, but instead a test set, used specifically to test the accuracy of the algorithm [37].

According to the task at hands, different performance measures should be used. For instance, when using regression tasks, a commonly used cost-function is the *sum squared error* function, while on classification tasks *cross entropy* cost-functions are the standard [38].

2.3.3 Machine Learning Experience

Experience-wise, algorithms can be split in Supervised Learning, Unsupervised Learning and Reinforced Learning. These will now be discussed in detail.

Supervised Learning

Supervised Learning is the most used type of ML, having its main focus on creating a model able to make prediction on unknown data, given an adequate set of labeled training data. Supervised Learning can be represented by the following expression,

$$\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i^N$$

where **D** represents a set of N input-outputs, which is called the training set. Each input, $\boldsymbol{x}_i$, is a vector composed by multiple features, such as the size and number of storeys of a house. Consequently, the outputs, $\boldsymbol{y}_i$, can either be a single scalar (which, in the previous example, would be the price of the house) or a vector. In the latter case, the output is said to be categorical, as usually only the components to which the sample belongs to take the value ‘1’, while all the other are 0-value components. As seen previously, cases with this kind of outputs are known as classification tasks, while the scalar-output ones are known as regression tasks [34]. These are the two main tasks present in Supervised Learning.

Considering a more graphical example of a classification problem, Figure 2.13 shows a set of 30 samples, half of which is classified as "+", while the other half is classified as "o". Using a proper classification algorithm, it is possible to find a boundary which enables the classification of new data into the proper category, given its x_1 and x_2 values.

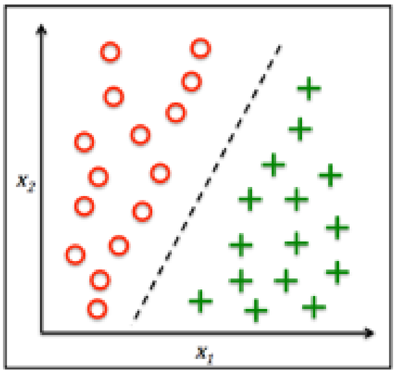


Figure 2.13 – Example of a classification problem [39]

Classification tasks can be used for multiple kinds of real-world applications, such as spam filtering, image classification, handwriting recognition, etc. [34].

Supervised learning algorithms are generally based on estimating the probability distribution of the output, and not the output itself, given a certain input $\boldsymbol{x}$ and an adequate parametric vector $\boldsymbol{\theta}$, $p(y|\boldsymbol{x};\boldsymbol{\theta})$. [37]

Considering regression problems, where the output usually follows a Gaussian distribution,

$$p(y|\boldsymbol{x};\boldsymbol{\theta}) = N(y;\boldsymbol{\theta}^T\boldsymbol{x},\boldsymbol{I})$$

in which the mean is $\mu = \boldsymbol{\theta}^T \times \boldsymbol{x}$ and the variance $\sigma^2 = \boldsymbol{I}$, it is possible to generalize linear regression on classification tasks: by considering a case with only two classes, the first class will take the value "0" and the second class will take the value "1". By using an activation function that reduces the output to a value between 0 and 1 (such as a *sigmoid function* ϕ – Figure 2.14), which then allows for a proper class prediction:

$$p(y = 1|\boldsymbol{x};\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}^T\boldsymbol{x})$$

This is known as logistic regression (even though it is used in classification tasks).

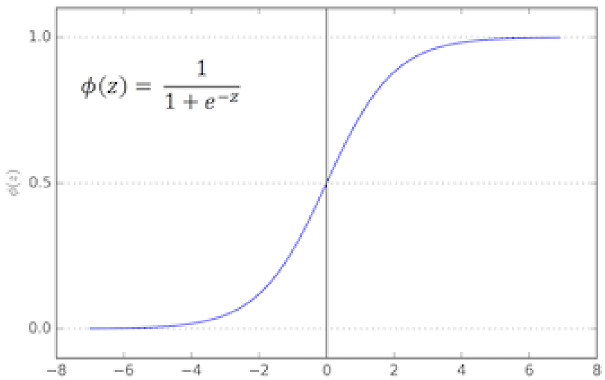


Figure 2.14 - Sigmoid function [40]

Supervised Learning is also the main technique used to train algorithms such as Artificial Neural Networks, Support Vector Machines and decision trees, among others. Figure 2.15

shows a block diagram that summarizes the implementation process of a Supervised Learning Algorithm.

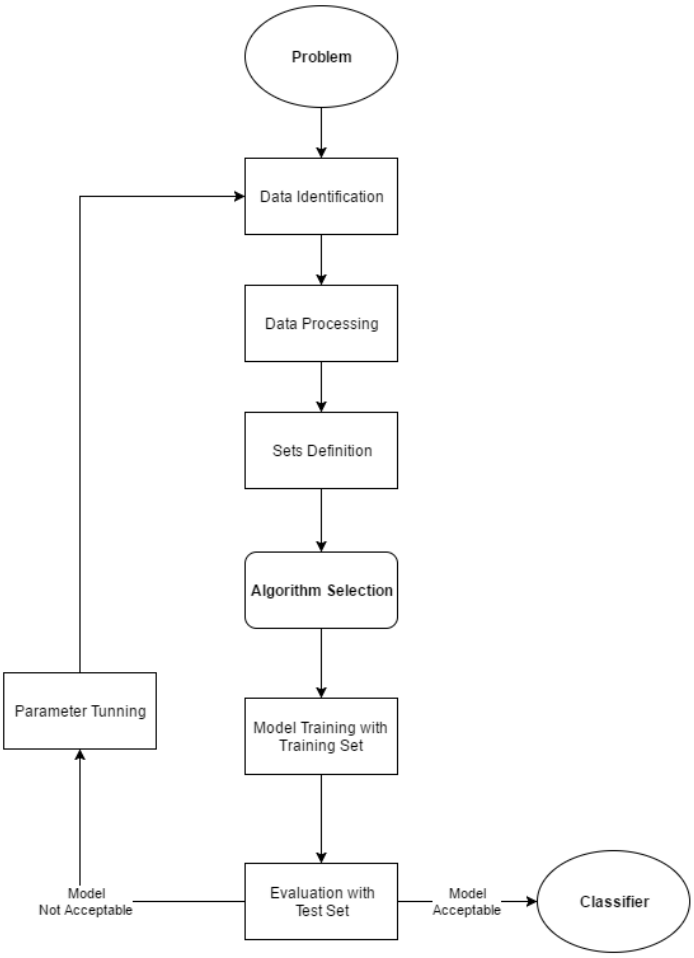


Figure 2.15 - Supervised Learning implementation process (block diagram)

Unsupervised Learning

In Unsupervised Learning the training data consists only on inputs, and the goal is to find patterns that allow grouping data. This is the form of Machine Learning closest to how humans process new knowledge [34]. As Geoffrey Hinton, a professor of ML in the University of Toronto said:

"When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds. So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information: from the input itself." – Geoffrey Hinton, 1996.

$$\boldsymbol{D} = \{\boldsymbol{x}_i\}_i^N$$

In this case, **D** represents a dataset of N inputs. The goal is to find a set of parameters which allows for extraction of meaningful information from the data, without the external guidance of a known output. Simply put, the task expected is a density estimation given from a model $\boldsymbol{\theta}$, which can be represented by the form $p(\boldsymbol{x}|\boldsymbol{\theta})$.

To do so, it is necessary to simplify the data. Three of the most common ways to do so are lower dimensional representations (Figure 2.16), sparse representations and independent representations.

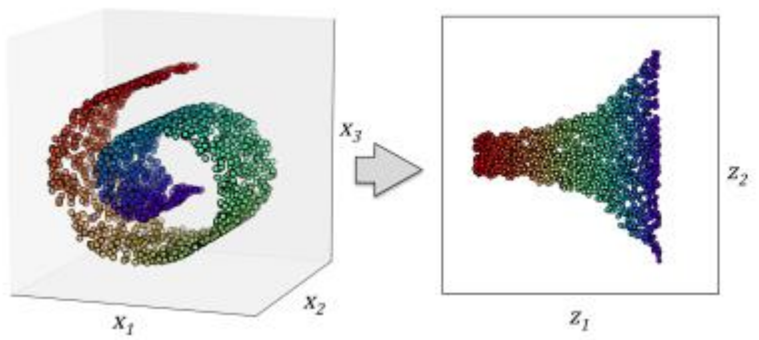


Figure 2.16 - Non-linear dimensionality reduction [39]

There are multiple algorithms that allow these simplifications, such as Principal Component Analysis (PCA) and *k*-means Clustering.

The purpose of PCA is to lower the dimensionality of the inputs by learning a representation of data. Its main goals are to represent the data in a lower dimension (Figure 2.17) and to perform a linear transformation so that the data varies along the axes (Figure 2.18).

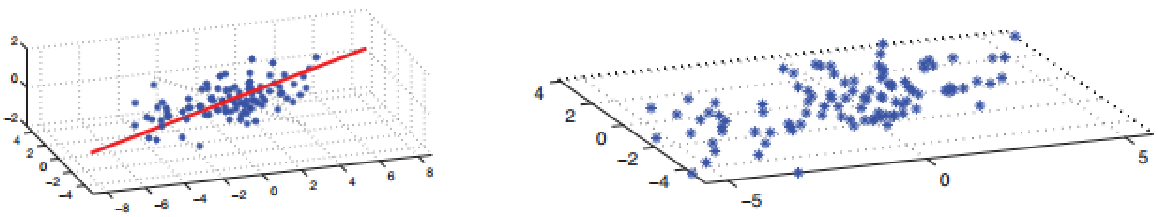


Figure 2.17 - Lower dimensional representation (right) of a data set (left) [34]

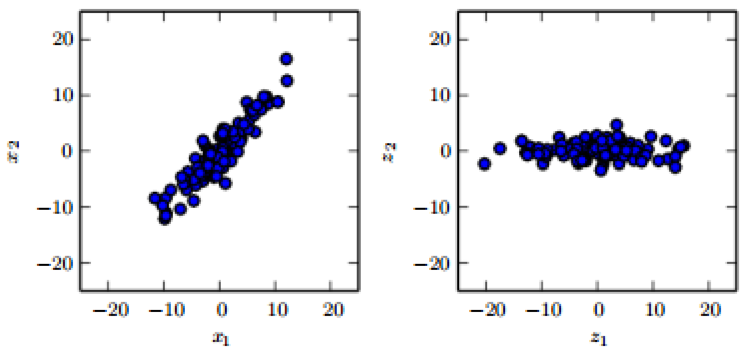


Figure 2.18 - PCA Application (right) to a data set (left), so that the data varies according to z_1 and z_2 [37]

Similarly, the purpose of *k*-means Clustering is to select *k* centroids μ so that each input can be assigned to a single cluster. Standardly each input $\boldsymbol{x}$ is reduced to a lower order vector $\boldsymbol{h}$ which is then assigned to the closest centroid.

Each iteration of a program running *k*-means Clustering has two steps: first each input reduced-order vector is assigned to the closest centroid μ^i; in the second step, the centroid values are changed to the mean of all vectors assigned to each cluster. Figure 2.19 shows three clusters for a certain data set.

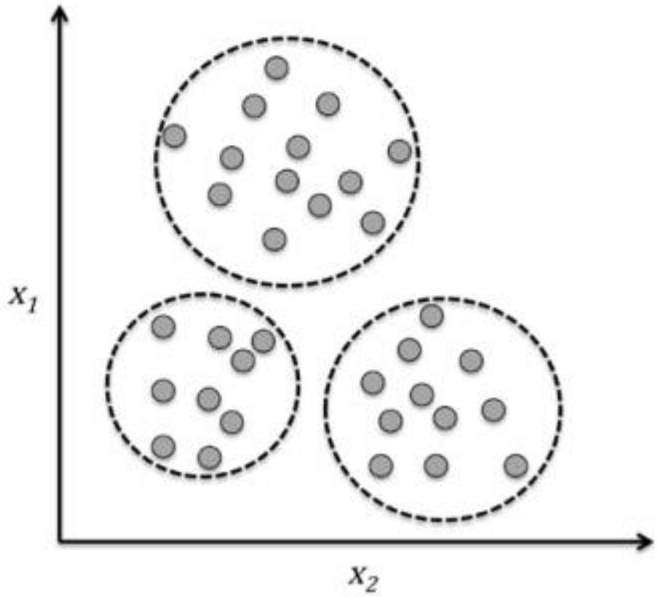


Figure 2.19 - Data clustering according to x_1 and x_2 [39]

Reinforcement Learning

Reinforced Learning, on the other hand, focuses on trying to learn what to do by maximizing rewards, whose values depend on the action taken [41]. This method is done by developing a system that interacts directly with its environment (Figure 2.20), improving its performances by trial and error. There are some similarities between Supervised Learning and Reinforcement Learning, but while in the first one there is a specific target for each input (a known output), in the latter the program only knows how well it is doing in comparison to other action choices, due to the reward system [41], [42]. This kind of learning has multiple applications, such as game engines (chess, back-gammon, etc.) and advertising (the program learns what kind of advertisements to show to each user by trial and error).

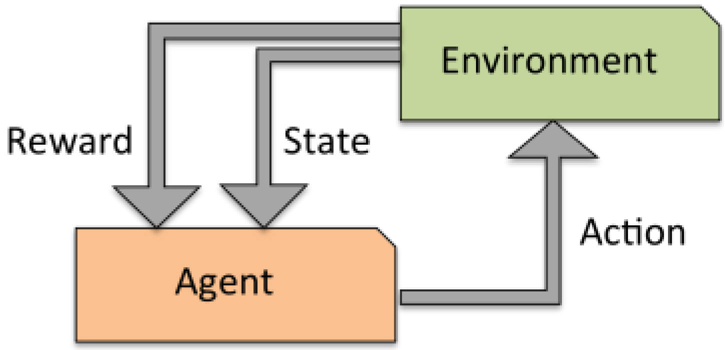


Figure 2.20 - Reinforcement Learning schematics [39]

The purpose of a RL program is to achieve one goal (e.g. on game engines the goal is to win the game) by selecting the maximum reward action-path from the starting point to the goal point. One thing to note is that when choosing an action, the program is not only setting a reward for that action, but also changing the possible subsequent actions, which in turn will have different rewards than if another action path was chosen. As such, two parameters must be taken into account: exploitation (reward-analysis for different actions on the same step) and exploration (full-reward analysis from the starting point to the goal) [41].

2.4 Summary

With this chapter, one can understand the importance of bearings as a major component of virtually every rotating machinery. This results in a necessity of monitoring these components throughout their whole operational life, with the help of different types of hardware (thermal cameras, accelerometers, among others) and software (such as SKF™ *@ptitude Analyst*). Regarding the methods that focus on frequency analysis, the knowledge of the fundamental fault frequencies of the bearings is a must, even if it often necessary to implement techniques such as Envelope Detection.

Then, the different sensors available to measure data, along with the proper data acquisition and processing methods, were discussed. The sensors studied were microphones, accelerometers and acoustic emission sensors. A brief analysis to the data acquisition equipment to be paired with these sensors was made, as well as processing methods such as filters and FFTs.

Finally, this chapter gave a notion of Machine Learning, one of the core aspects of Artificial Intelligence. The understanding of the different tasks available, as well as the types of learning according to the data base enables a smarter choice when selecting a Machine Learning algorithm, which then can be evaluated according to its performance.

3 Artificial Neural Network Development on LabVIEW

This chapter tries to familiarize the reader with LabVIEW, a systems engineering software based on visual programming [43] at the same time as a study to Artificial Neural Networks – the Machine Learning algorithm chosen to develop during this project – is made.

This reconnaissance was done before the actual bearing's data acquisition. During this period, multiple Virtual Instruments (VIs) were created, either to test LabVIEW's most basic functionalities or to develop a fully functional Artificial Neural Network.

3.1 Data Analysis with LabVIEW

The first step towards data acquisition and processing was to create a VI to represent multiple signal functions, as well as their FFTs (Figure 3.1 and Figure 3.2).

To this purpose, a signal resulting of the sum of four sine waveforms and periodic random noise was created. After, through FFT analysis with a Hanning window, the minimum and maximum fundamental frequencies of the signal were detected through a "Peak Detector" sub-VI. With these frequencies, a bandpass Butterworth filter was created to isolate the noises with frequencies below the minimum frequency and above the maximum frequency (as seen in Figure 3.2).

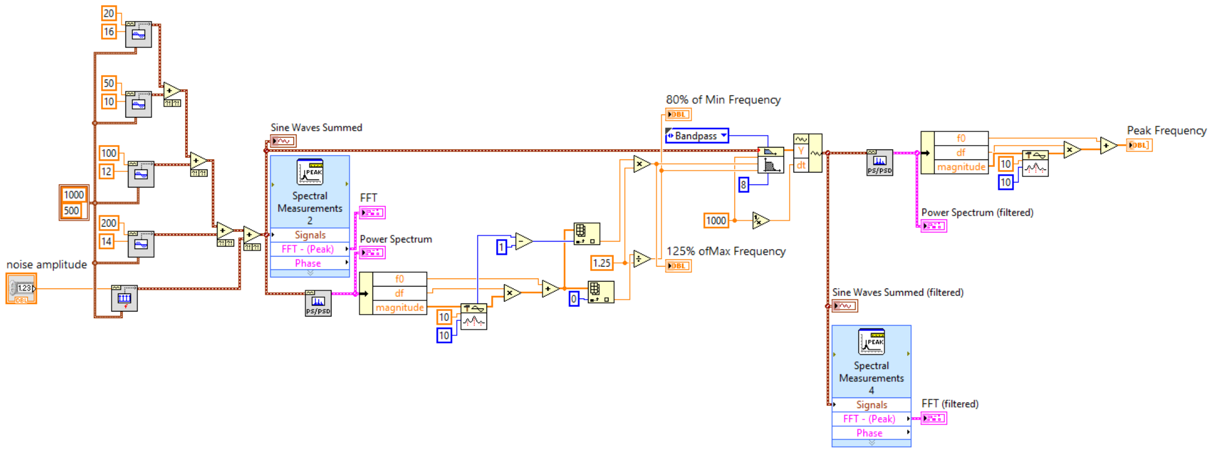


Figure 3.1 – "Butterworth filter and FFT testing" block diagram

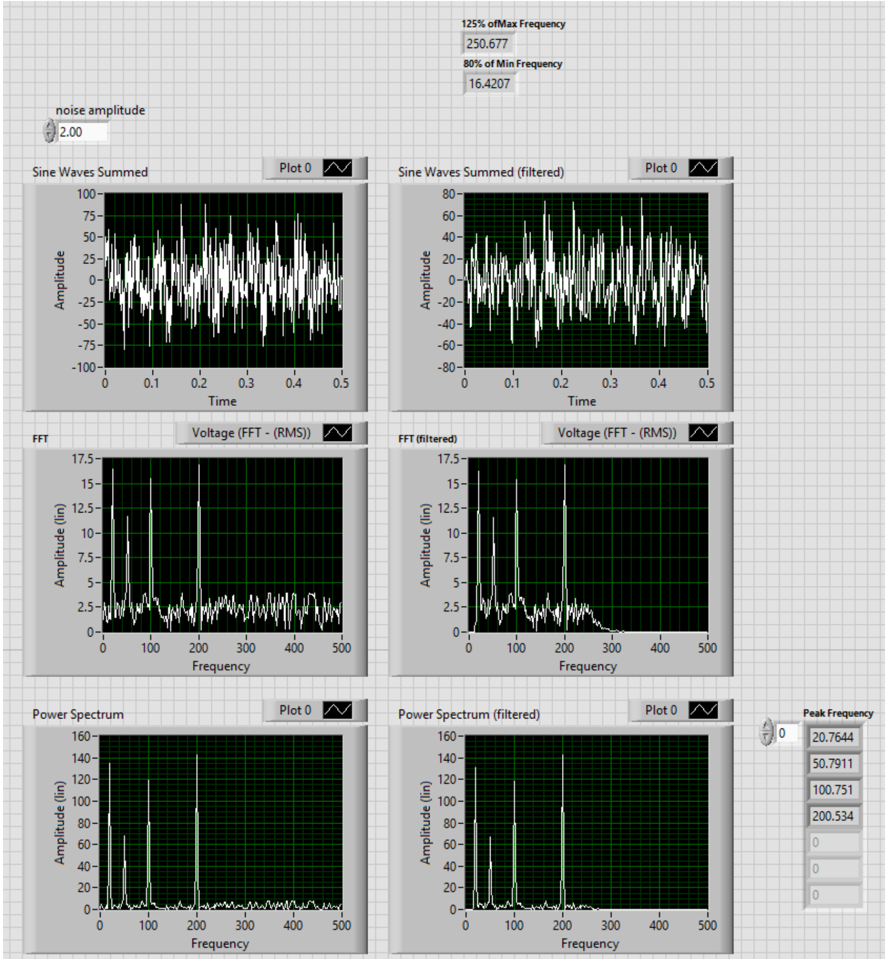


Figure 3.2 – “Butterworth filter and FFT testing” interface

3.2 Development of the Artificial Neural Network

3.2.1 Definitions

Before approaching the functioning of an Artificial Neural Network, it is best to understand some key concepts when using Machine Learning Algorithms, in this case specifically an Artificial Neural Network:

- **Data splitting**: when training a ML algorithm, it is wise to separate the initial data set in three components (the percentages below are indicative values):
  - Training set (60%): data used for the algorithm to learn a proper function;
  - Cross-validation set (20%): set used to compare different models and choose the one with the best performance;
  - Test set (20%): data used to test the accuracy of the learned model;
  - If only one model is being used, the data should be split in 60-70% training set and 30-40% test set.
- **Overfitting**: this phenomenon happens when the model learnt by the algorithm fits the training set too well, meaning it will have near 100% accuracy on the training set data, but will fail when used on other data (Figure 2.2).

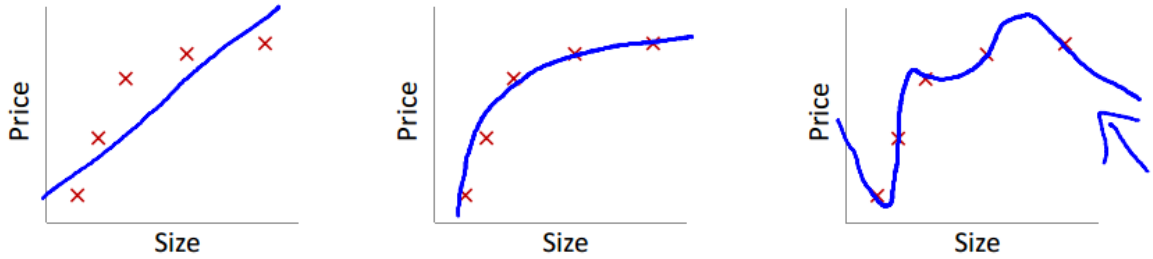


Figure 3.3 - Overfitting (right) [44]

- **Regularization**: way to prevent overfitting, by reducing the magnitude of the weights (expect the bias term, which is not regularized) on the final function.
- **Weight Initialization**: weights are randomly initialized according to the following formula, to ensure that the parameters remain small throughout the learning process, making it more efficient (s_l and s_{l+1} represent the size of the layers adjacent to $\theta^{(l)}$) [44]:

$$W = rand(s_{l+1}, 1 + s_l) * 2 * \frac{\sqrt{6}}{\sqrt{s_l * s_{l+1}}} - \frac{\sqrt{6}}{\sqrt{s_l * s_{l+1}}} \tag{2}$$

- **Normalization**: reducing the range of all the inputs to a certain interval (usually [0;1] or [-1;1]) so that certain parameters don't bias the whole model. It is not mandatory, but in some cases the model won't be able to learn from the data if normalization is not implemented. Data normalization is done by converting the data to a Gaussian curve:

$$y_{norm} = \frac{y - \mu}{\sigma} \tag{3}$$

- **Data selection**: When choosing a training set, this should be as general as possible, such as to cover as many different types of samples as possible, or else the model, even if it does not overfit, will not be able to adapt when a new type of data is acquired.
- **One-hot Encoding**: To classify different types of input, their output value must be different. One commonly used option is to use on-hot encoding, which means the output for each class is a vector of size *n* (equal to the number of classes) with a single component with value "1", while all the others have 0-value. The 1-value component is different for each class, assuring a proper output differentiation. This method is often preferred over a single output with different values for each class, as this latter method intrinsically values some classes higher than others, which may cause confusion to the model further down the road.

3.2.2 Artificial Neural Network

This subsection will focus on a detailed analysis on Artificial Neural Networks (ANNs), the chosen algorithm to develop the practical component of this dissertation.

As seen previously, an ANN is a Machine Learning Algorithm often used in Supervised Learning. The concept emerges from the attempt to simulate Biological Neural Networks, such as the human one. However, there is still a long way to go before ANNs reach the complexity level of biological ones. Not only that, but an ANN does not provide better solutions than explicit programming [45].

Nonetheless, an ANN has two major advantages:

- It is based on very simple computations (requiring little processing power), independently of the problem at hands;
- It is not restricted to a set of problems, being extremely adaptable (like a human or any other animal's brain).

These properties alone give ANNs a large ground of work in Machine Learning compared to many other algorithms.

The basics of an ANN, once again, try to replicate the biological systems, more specifically neurons (Figure 3.4) and full nervous networks. In these networks, inputs are acquired by nerves (e.g. when touching something the affected nerves will be activated, generating signals – inputs) and then transmitted through multiple intertwined neurons, until a final output is transmitted to the brain, that processes the information.

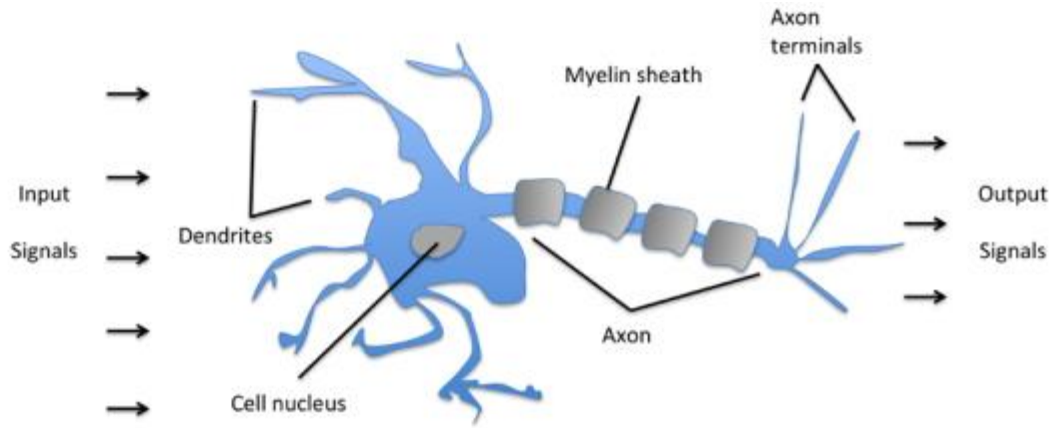


Figure 3.4 – Neuron [39]

In this process, a neuron can be seen as nothing more than a logic gate with *n* inputs and *m* outputs, which are activated according to the stimulus received. Figure 3.5 schematizes a single neuron in mathematical terms. In this case, the output h($\boldsymbol{x}$) is a function of the inputs x_1, x_2 and x_3, of the parameters $\boldsymbol{W}$ (the weights) and b (the *bias term*, usually equal to +1).

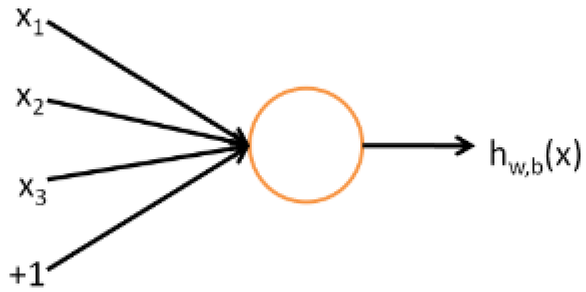


Figure 3.5 - Mathematical equivalent of a neuron [46]

When considering an ANN, the process is the same. We have multiple layers connected by artificial neurons. Figure 3.6 schematizes a 3 layer ANN: the first layer represents the input, which, after being added a *bias term*, goes through a neuron to the second layer (with *n* nodes), being the input split in *n* parameters (according to number of nodes). The output of this second layer goes to the third layer through another neuron, resulting on the actual output of the whole network. A good rule of thumb is that the number of nodes should be smaller than the previous layer and higher than the next layer.

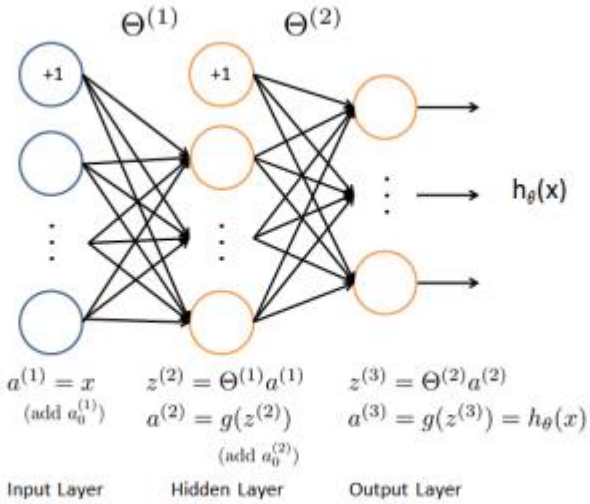


Figure 3.6 – Artificial Neural Network model [44]

Each neuron has a set of parameters $\boldsymbol{\Theta}$ (weights) that transforms the input $\boldsymbol{a}^{i-1}$ (with a bias term included) in $\boldsymbol{z}$, an intermediary value, which goes through an activation function g (as seen in 2.3.3 when studying logistic regression), resulting in the output of that same neuron $\boldsymbol{a}^{i}$:

$$\boldsymbol{a}^{i} = w(\boldsymbol{a}^{i-1})$$
$$\boldsymbol{z}^{i} = \boldsymbol{\theta}^{i-1}\boldsymbol{a}^{i-1} \tag{4}$$
$$\boldsymbol{a}^{i} = g(\boldsymbol{a}^{i-1})$$

Since the only values that are visible from the outside of the ANN, every layer between the first (input layer) and the last (output layer) is called "Hidden Layer". ANNs usually have one or two hidden layers, while Neural Networks with more hidden layers are usually called Deep Neural Networks [44].

The process described above is called feed forward propagation. However, if the output of the neural network is not compared with the actual output, there is no learning in the ANN. A way to overcome this is by using a Back-Propagation Algorithm. The Back-Propagation (BP) is based on stochastic gradient descent and the chain-rule of calculus [34].

To have a comparison between the ANN output and the actual output, a *cross-entropy* cost function is the best option, as we are considering a classification task. This function can be defined as [44]:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(h_\theta(x^{(i)})\right)_k + \left(1 - y_k^{(i)}\right)\log\left(1 - h_\theta(x^{(i)})\right)_k\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(\theta_{ji}^{(l)}\right)^2 \tag{5}$$

Where:

$y_k^{(i)}$ – actual output of the kth <u>parameter</u> from the ith sample;

$h_\theta(x^{(i)})_k$ – ANN output of the kth parameter from the ith sample;

m – number of samples;

λ – regularization parameter;

$\theta_{ji}^{(l)}$ – weight's component of the ith row and jth column between layer s_l and s_{l+1}.

The first term of the cost function is the cross-entropy function, while the second one is a regularization term, whose influence in the cost function is controlled by λ.

The purpose of the BP is to minimize the cost function on every iteration of the ANN during the training process. To do so it is necessary to calculate the error cost on every layer (except the first one, because the input is independent of the model). For the last layer, this is simply the difference between the actual output and the ANN output:

$$\delta_k^N = h_\theta(x)_k - y_k \tag{6}$$

For every other layer though, the expression is slightly different:

$$\delta^{(l)} = \left(\theta^i\right)^T \delta^{(l+1)} .* \, g'(z^{(l)}) \tag{7}$$

where $g'(x)$ is the derivative of the activation function. In the case of the sigmoid function, this comes as:

$$g'(x) = g(x)\big(1 - g(x)\big) \tag{8}$$

With the error costs, it is possible to compute delta, Δ:

$$\Delta^{(l)} = \Delta^{(l)} + \delta^{(l+1)}\left(a^i\right)^T \tag{9}$$

Finally, the regularized gradients come as:

$$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = \frac{1}{m} \Delta_{ij}^{(l)} \qquad \text{for j=0} \tag{10}$$

$$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = \frac{1}{m} \Delta_{ij}^{(l)} + \frac{\lambda}{m} \theta_{ij}^{(l)} \qquad \text{for j≥1} \tag{11}$$

The first term is not regularized because it corresponds to the bias term.

With these values, the weights can be adjusted, with this adjustment being regulated by the learning rate α:

$$\theta_{ij}^{(l)} = \theta_{ij}^{(l)} - \alpha \frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) \tag{12}$$

3.2.3 Artificial Neural Network Implementation

The ANN models used for this dissertation result from an adaptation to LabVIEW of the model developed on the Stanford University *Machine Learning* course, by Andrew Ng [44].

This ANN has only one hidden layer, with a variable number of hidden nodes (chosen in the interface). To implement the ANN correctly, the following steps must be taken:

1. Choose the initial parameters: data parameters to study (RMS, standard deviation, etc.), number of hidden nodes, learning rate α, regularization influence λ and number of iterations for learning. If no value is chosen for the iterations, the ANN will learn from the whole training set;
2. Initialize weights according to the size of each layer (Equation 2);
3. Add bias term (+1) to the input;
4. Multiply the output of step 3 by the first layer's weights, θ^1;
5. Activate the output of step 4 with the activation function, $g(x)$;
6. Repeat steps 3-5 with the output from step 5 and with the second layer's weights, θ^2, which results in the ANN output, $h_\theta(x)$;
7. Calculate the cost function (Equation 5);
8. Calculate the error cost for each layer (Equations 6 and 7);

9. Calculate the deltas (Equation 9);
10. Calculate the gradients (Equations 10 and 11);
11. Adjust the weights (Equation 12);
12. Repeat steps 3-11 through all the iterations;
13. Save the final weights' values.

After these 13 steps are completed, a functional model of the ANN is obtained. The LabVIEW virtual instrument (VI) from Figure 3.7 and Figure 3.8 represent the ANN model applied to detection of distinct types of waves (sine, triangular and square waves). To generate the waves, a random number generator was used to vary their frequency and amplitude. All the sub-VIs from this program can be found on Appendix B.

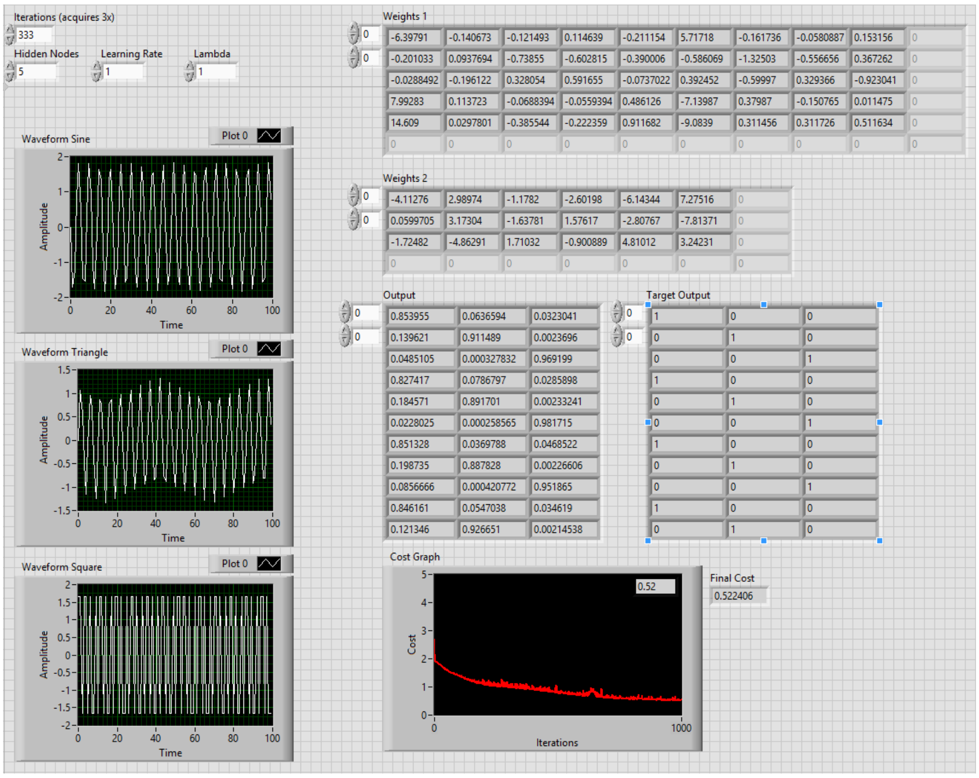


Figure 3.7 – "ANN Learning" interface

As this was a first test, the parameters selected were not optimized, resulting in an input of eight different parameters. With proper selection, this same model could function with only two parameters with the exact same accuracy. Since this is a 3-class problem, by using one-hot encoding the output will have three parameters.

It is also important to note that normalization was not included in this model.

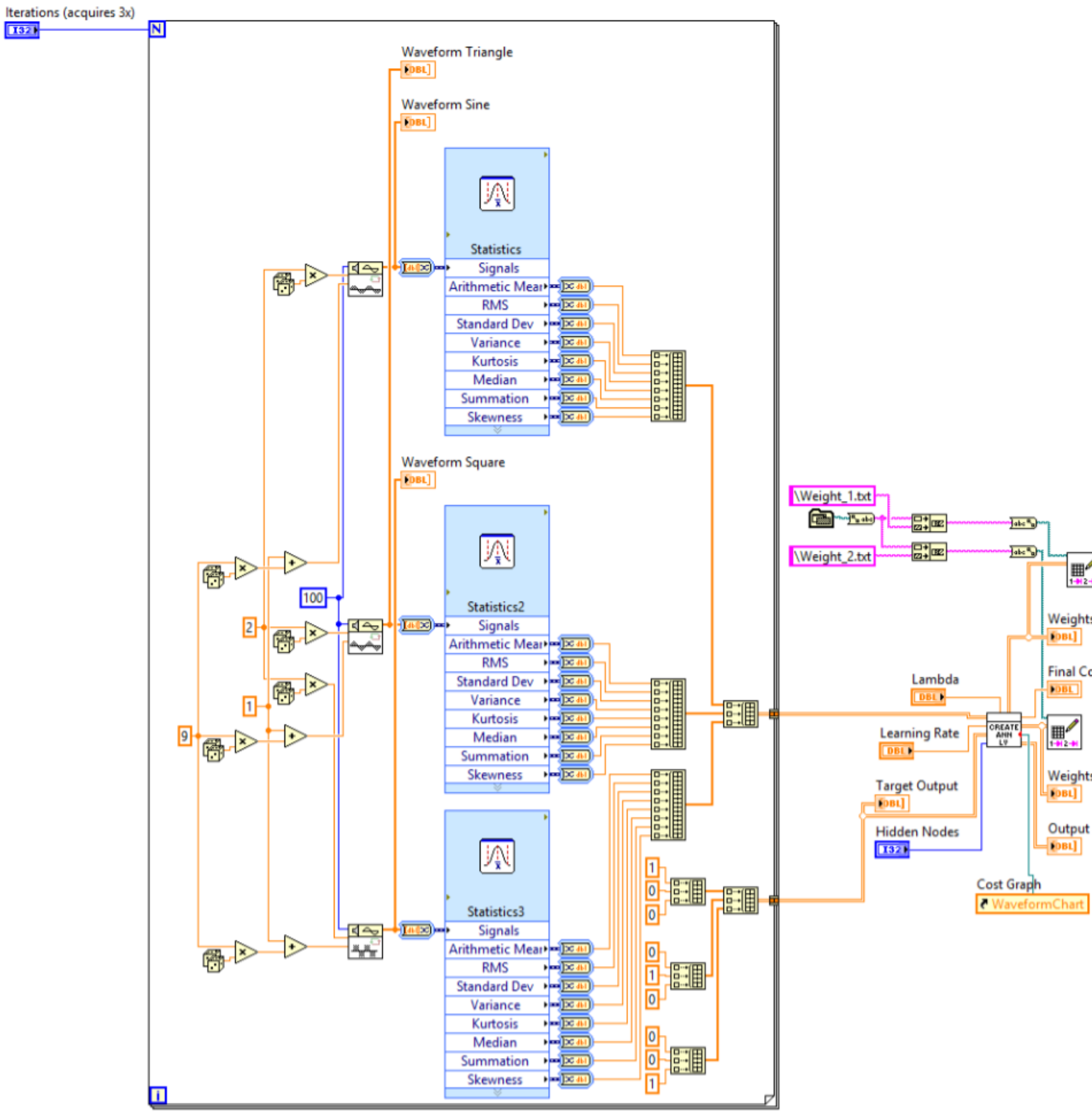


Figure 3.8 – "ANN Learning" block diagram

3.2.4 Testing the Artificial Neural Network

To test the ANN, another VI was created (Figure 3.9 and Figure 3.10). This VI implements the ANN model, as created on the previous section, on the test set. It then classifies this data according to the output's component with the maximum value and compares it with the actual classification.

For example, if a given data sample from a triangular wave (whose one-hot position is the second component of the output vector: 010) outputs values as shown on "Output Tri" on Figure 3.9, the "Test ANN" VI will classify this sample as data from a triangular wave, since the component of the output vector with the maximum value is the second one.

Using this method, and after the initial ANN used a training set of 300 samples (100 for each class), the model displays an error of approximately 9%. However, if the number of training samples is increased to 999 samples (333 of each), the error decreases to approximately 2%, proving the importance of a large database when training a new model.

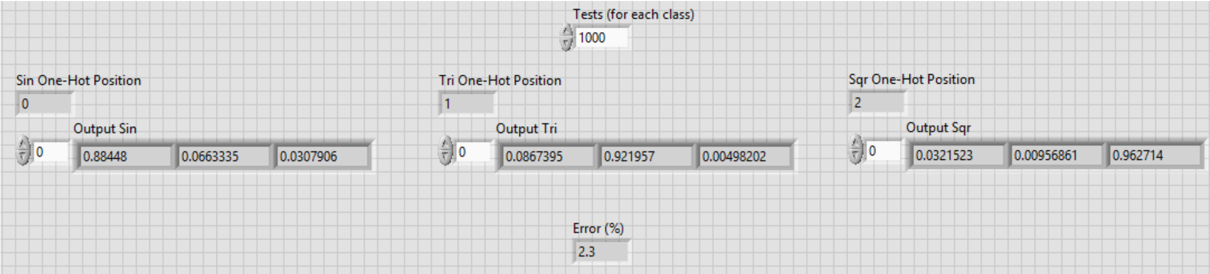


Figure 3.9 – “Test ANN” interface

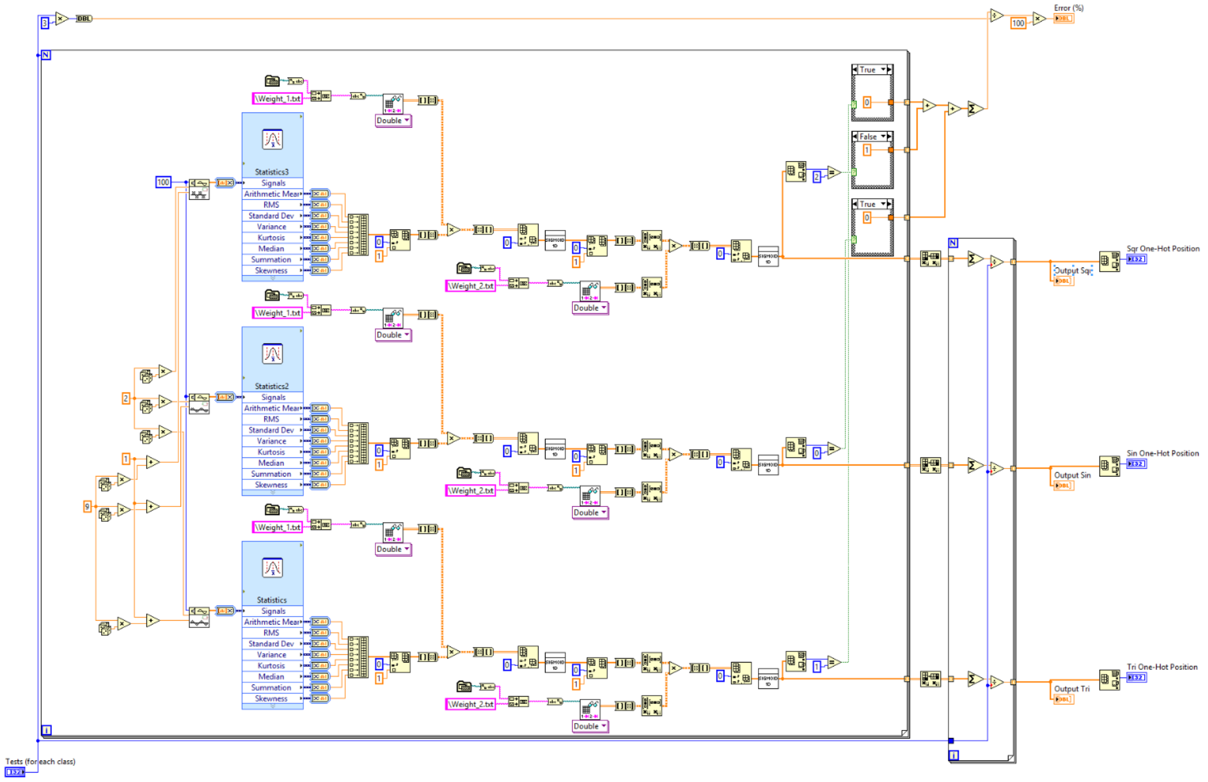


Figure 3.10 - "Test ANN" block diagram

3.3 Summary

With this chapter, a clearer understanding on the basics of LabVIEW can be attained, followed by the implementation of functions such as Fast Fourier Transforms, Peak Detectors and Butterworth Filters.

Also, since the algorithm used during this dissertation is an Artificial Neural Network, a broader review of this topic was done, along with some of the most important definitions related to it (regularization, overfitting, weight initialization, among others), followed by the implementation of this algorithm on LabVIEW. By testing this algorithm, it is possible to understand the influence of the size and variety of the training set on the accuracy of the model.

This page was intentionally left blank.

4 Data Acquisition and Processing

Before delving into the practical analysis of this dissertation it is important to review some aspects with relevance to the work flow: the geometrical properties of the bearings and the method used to distinguish healthy bearings from faulty ones. With this analysis, a first approach to LabVIEW data acquisition and processing was made, followed by a study to the work conditions and programs developed during this dissertation.

4.1 Bearings' Properties and Status

For the development of this dissertation, the bearings studied are the ones used in the *Thies Clima*™ anemometers [47]: Wind Transmitter "First Class" Advanced (details can be found on Appendix A). These bearings are from GRW™ and can be identified by the following part number: SS624-2Z P5.

Figure 4.1 shows an exploded view of this anemometer. The bearings are represented by the numbers 8 and 11.

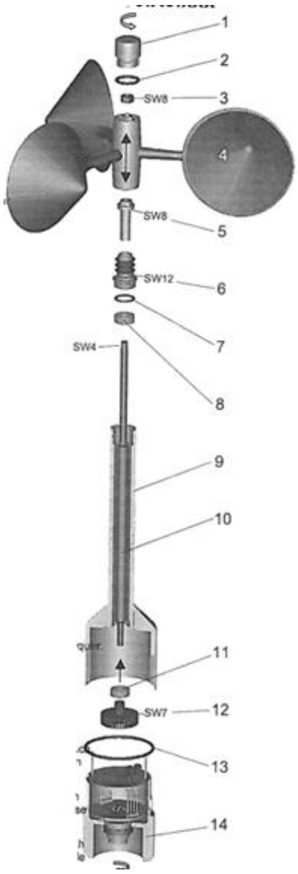


Figure 4.1 - *Thies Clima*™ Wind Transmitter "First Class" Advanced (exploded view)

These bearings are sealed ball-bearings, which prevents particles from contaminating the bearing, reducing its operational life. This is a common measure in hostile environments, such as in the case of anemometers (the wind carries all kind of particles hazardous to the proper bearing's operation).

4.1.1 Geometrical Properties

After opening one of the bearings (Figure 4.2), its geometrical properties were analyzed.

Figure 4.2 - Anemometer's bearing; mounted (left) and cut (right)

The properties, following the nomenclature used in Equation (1), were the following:

N (number of balls) = 8;

B (ball diameter) = 2 mm;

P (pitch diameter) = 10 mm;

Θ (contact angle) = 0-15º (varying with the load).

With these values, we can estimate that the bearing's fundamental fault frequencies reach a maximum around 4.8 times the frequency of the shaft's rotation, F (for BPFI).

4.1.2 Bearings' Status

Regarding the condition of bearings, the tested anemometers were differentiated by the sound heard from the cups when the body was spinning upside down (Figure 4.3).

The reference case was defined as bearings from anemometers which produced no distinguishable sounds when spinning, leading to the assumption that the bearings from anemometers in this condition were healthy. On the other hand, when unnatural noises were heard, the bearings from anemometers in this condition were considered faulty.

Figure 4.3 - Sound analysis of an anemometer

4.2 Equipment Selection

For this project, a smartphone's microphone (24-bit resolution with 192kHz sampling rate [48]) and a headphone's microphone (16 bit resolution with 44.1kHz sampling rate) were the devices used for audio data acquisition.

Regarding accelerometers, the accelerometer PCB 352A24 (Appendix C) was used to acquire vibration data, along with a NI™ USB-6009 DAQ (details on Appendix D). A signal conditioner was also used (PCB Model 480E09 – Appendix E), amplifying the signal 100 times.

Finally, multiple suppliers were contacted to obtain quotations for different AE sensors which fitted the constraints imposed, namely a maximum thickness of 10mm, a frequency range of 100-500Hz and the minimum price possible. Table 3 represents the final sensors selected, with KRN m500 sensor having the best set of properties, given the constraints.

Table 3 - Acoustic Emission sensor selection

Sensor	Brand	Dim. (mm)	Weight (g)	Freq. (kHz)	Price
8152C	Kistler	15x23.5	29	100-900	€€€
m300	KRN	9.5x11.8	3.4	60-475	€
m500		4x4.8	8	60-750	€
VS370-A1	Vallen	7x13.5 (screwed)	3	170-590	€€
VS600-Z1		4.75x5.8	20	550-730	€€
VS700-D		6.3x10	20	150-800	€€
AE104A		8x18	5	100-400	€€
AE144A		8x18	5	100-500	€€

However, due to the already available microphones and accelerometer, none of these AE sensors was acquired, preventing an AE analysis to be made.

4.3 Wind Tunnel of INEGI

4.3.1 Test Conditions

Data acquisitions were made using INEGI's (Institute of Science and Innovation in Mechanical and Industrial Engineering) facilities, more specifically the wind tunnel (Figure 4.4). The acquisitions were made at the same time the anemometer was subjected to a calibration process. The calibration allowed for the acquisition of two sets of data at the same wind speed, ranging from 4 to 16 m/s with intervals of 1m/s.

Figure 4.4 – INEGI's wind tunnel [49]

As seen on sections 4.1 and 2.1.2, the maximum fault frequency should have a maximum value around 4.8F, which means that the sampling rate should have a minimum value around 48F, where F is the shaft's rotational speed. By consulting Appendix A, it is possible to understand that, at a radius of 80mm (where, for calculation purposes, the accelerometer's cup speed can be assumed equal to the wind's speed), the anemometer has a linear speed of 4-16m/s during the calibration process. By converting this to rotational speed:

$$n\left(\frac{rad}{s}\right) = \frac{v\left(\frac{m}{s}\right)}{r(m)} \Rightarrow n = \frac{4\sim16}{0.08} = 50\sim200 rad/s \approx 7.95\sim31.83 Hz \tag{13}$$

This means that the maximum fault frequency is around 4.8*31.83=152.8Hz. Considering that the acquisition system's sampling rate should be ten times higher than highest sampled frequency, all the acquisition devices should have a minimum sampling rate of 1.53kHz.

At the same time, considering the smallest rotational frequency is approximately 8Hz, when splitting samples these should never be smaller than 1/8=0.125s. To ensure this, data is always split in 1s samples (or bins), for data sets creation purposes.

The following scheme (Figure 4.5) pictures the acquisition process with an accelerometer, where the acquired signal is filtered and amplified by the signal conditioner, transmitted by the DAQ device and finally processed with a LabVIEW VI. The process with a microphone doesn't require the signal conditioner nor the DAQ device.

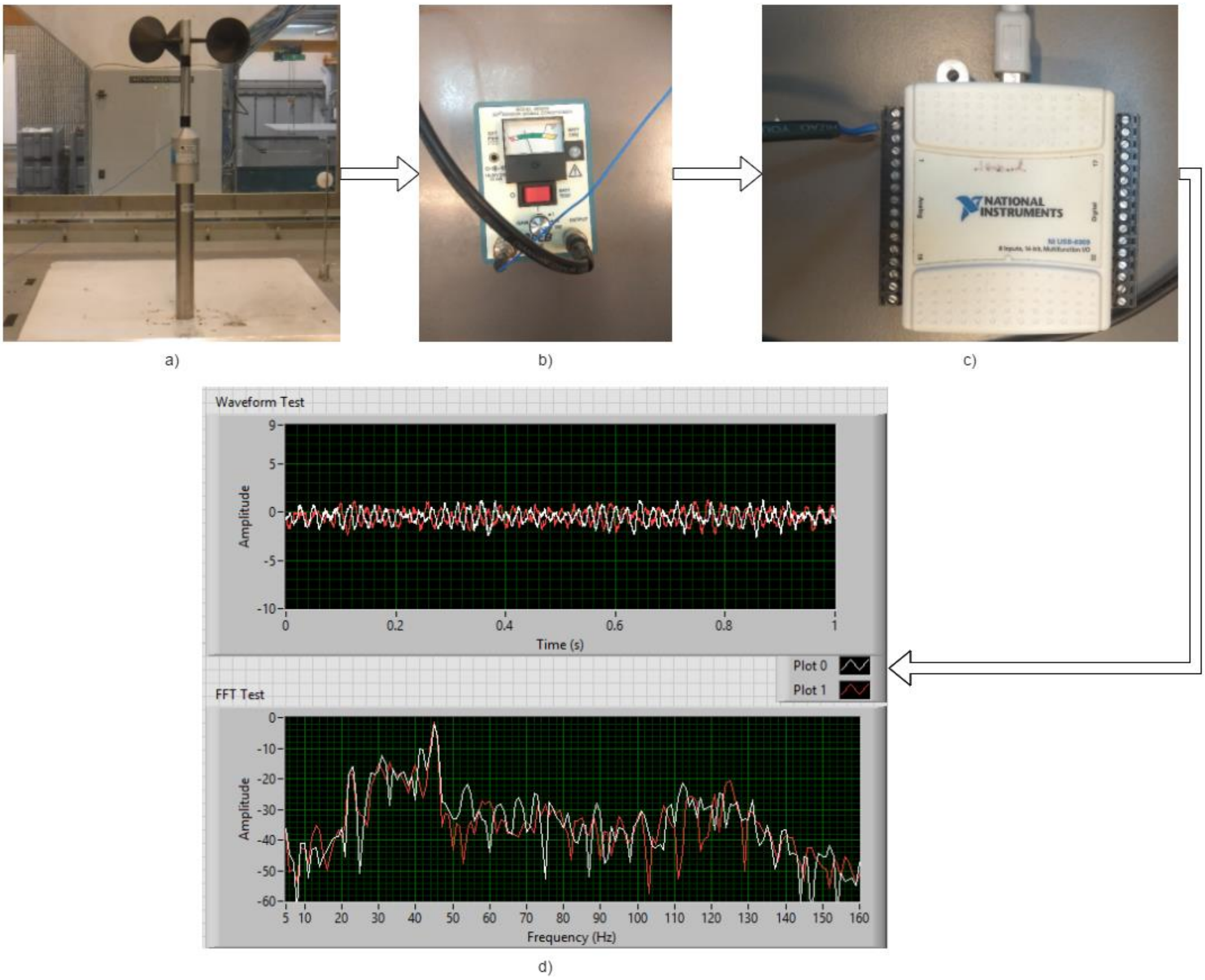


Figure 4.5 - Block diagram representing the data acquisition process with an accelerometer: a) accelerometer attached to the anemometer with duct tape; b) signal conditioner; c) DAQ device; d) data analysis on LabVIEW

4.3.2 Microphone

By attaching a simple headphone’s microphone to the anemometer’s support shaft on the outside of the wind tunnel (to reduce the noise from the wind), audio data were acquired. To do so, a simple VI was created (Figure 4.6) and integrated on the calibration VI, using a sampling rate of 44.1kHz.

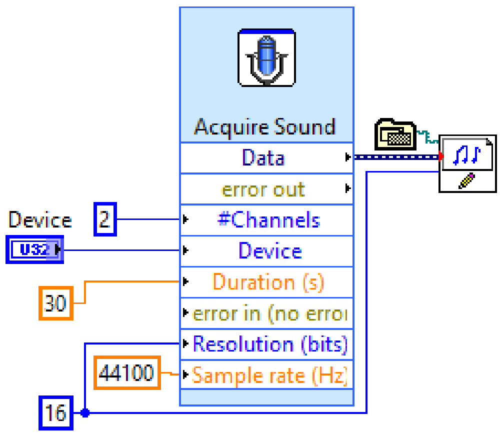


Figure 4.6 – “Sound Acquisition” VI

This VI acquires data from the microphone in stereo mode with a resolution of 16bits, for a duration of thirty seconds at a sampling rate of 44.1kHz. The data is then saved at the specified directory.

4.3.3 Accelerometer

The conditions for data acquisition with an accelerometer were similar to the ones with a microphone. The main difference was the position of the accelerometer, which was attached directly in the anemometer's shaft inside the wind tunnel with duct tape, as well as the signal conditioner and the DAQ device (Figure 4.5). Also, the accelerometer terminals were connected in differential mode, in order to decrease the noise in the measurements.

The VI created was also similar to the microphone's one, with a "DAQ assistant" sub VI instead of a direct "Acquire Sound" sub VI (Figure 4.7). Considering the accelerometer used (Appendix C) has the same performance from 1 to 8000Hz, a sampling rate of 8000Hz was used, also for thirty seconds.

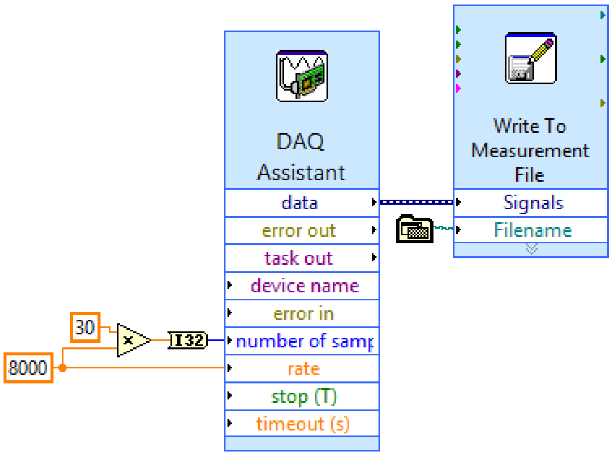


Figure 4.7 – "Accelerometer Data Acquisition" VI

The data is again saved at the specified directory.

4.4 Microphone on Isolated Environment

The last data samples were acquired using a smartphone's microphone placed close to an upside-down anemometer whose shaft was rotated by hand (Figure 4.8). This procedure was done in an isolated environment (inside an empty conference room) to cancel outside noises. It should be noted that the microphone placed on the desk was only included in this process to work as a support for the smartphone.

After the audio samples were acquired (as high-resolution audio .mp3 files), they were converted to .wav files with a MATLAB function (Figure 4.9), so that they would be compatible with LabVIEW.

Figure 4.8 - Sound acquisition in an isolated environment

```matlab
function [] = mp3towav()
%Converts .mp3 files to .wav files
Files=dir([pwd '\*.mp3']);
Files={Files.name}';
i=1;
while i<size(Files,1)
    i=i+1;
    File=strjoin(Files(i,1));
    [filename, FS]=audioread(File);
    File=strrep(File,'.mp3','');
    audiowrite([File '.wav'], filename, FS);
end
```

Figure 4.9 - MATLAB function that converts all .mp3 files in a folder to .wav files

4.5 Summary

This chapter provided an insight on the properties of the studied bearings, allowing an estimation of their fundamental frequencies. It also explained the method used to distinguish faulty bearings from healthy ones, by spinning the anemometers upside down.

Preceded by the selection of the different equipment used for data acquisition, the different acquisition methods used during this project were explained, either inside a wind tunnel (with a microphone and with an accelerometer) or in an acoustically isolated environment. The mounting conditions were described, as well as the basic VIs used for data acquisition.

This page was intentionally left blank.

5 Experimental Results

In this chapter, the data acquired using the methods described in the Chapter 4 is analysed to understand its applicability to train an Artificial Neural Network model. This is done by understanding if each data set has both parameters that allow bearings' distinction, as well as an acquisition methodology that can be repeated with reliability, providing accurate results.

The database that meets these requirements is then used to train an Artificial Neural Network model, whose results are analysed to understand its potential on bearing's fault detection on any anemometer of the same class.

5.1 Acquired Data Analysis

This section shows how the data was analysed for each one of the databases (inside the wind tunnel with both a microphone and an accelerometer, as well as with a microphone in an acoustically isolated environment) in order to select the most viable one to create a functional ANN model.

5.1.1 Wind Tunnel: Microphone

After the data was acquired, a VI was created to analyze the acquired data (Figure 5.1 and Figure 5.2). This VI displays the signal and computes its FFT.

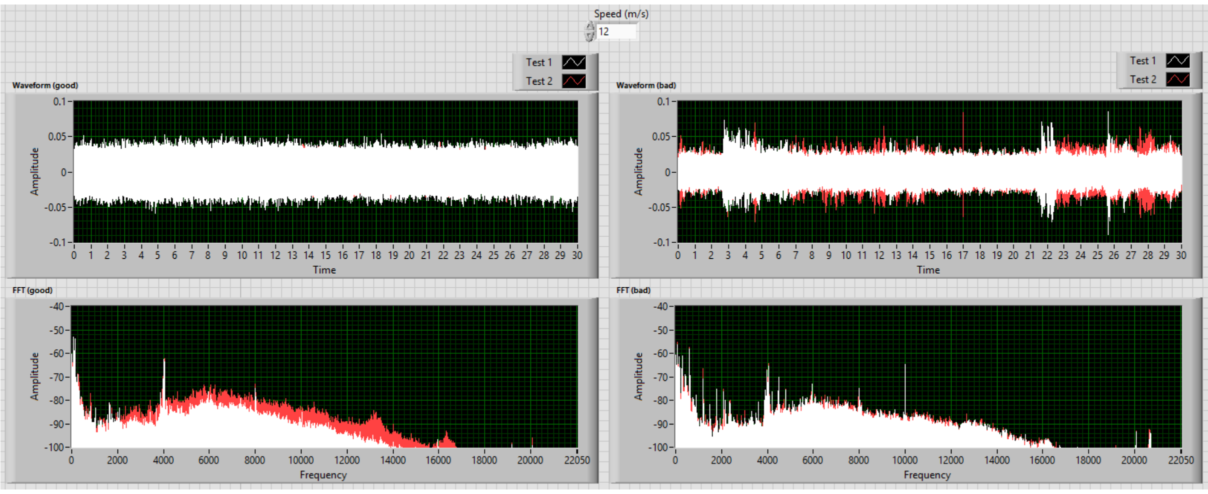


Figure 5.1 – "Audio Files Analysis" interface. The first row of graphs represents the waveform of two 30 seconds samples of good (left) and faulty (right) bearings, at a speed of 12 m/s; the second row represents the FFTs of the same signals

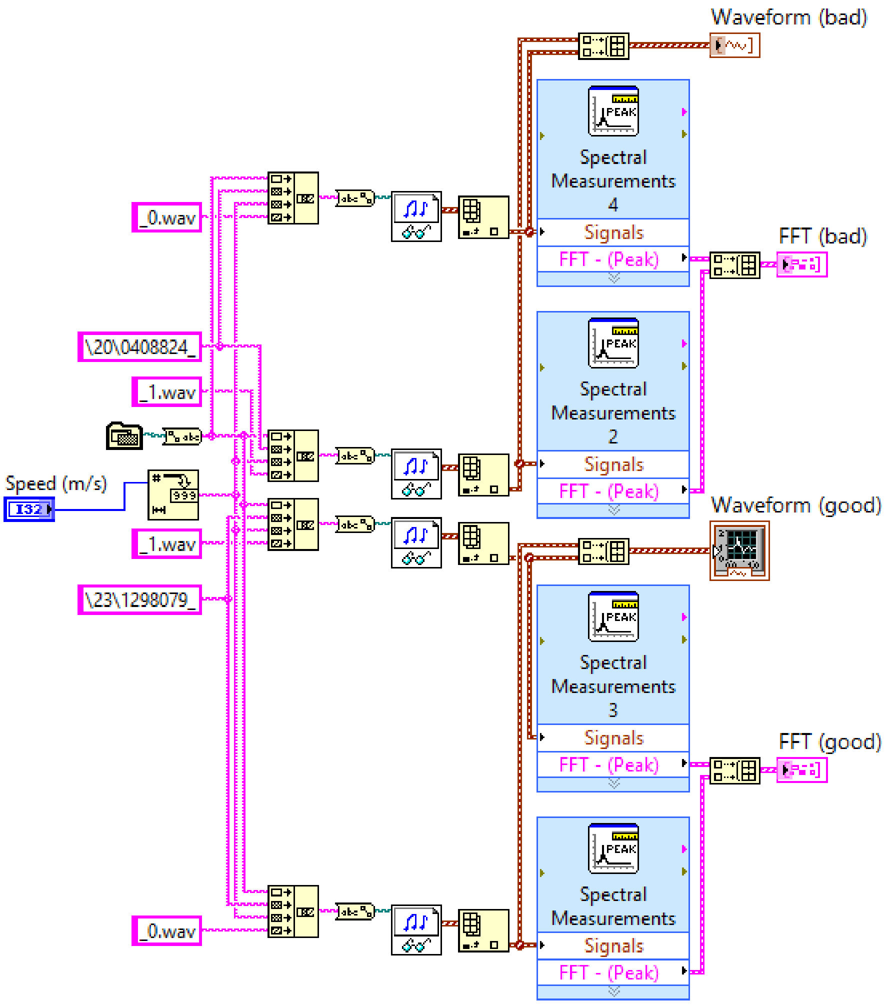


Figure 5.2 – “Audio Files Analysis” block diagram

However, due to the existence of multiple noises (other machines, ventilation system, people talking, music and the wind inside the wind tunnel), it was not possible to detect any relevant difference between reference and noisy anemometers’ data, rendering this acquisition method inapt for an ANN training.

5.1.2 Wind Tunnel: Accelerometer

Again, as done with the microphone, a VI was created to compute the signal FFT, this time to find manually the most relevant parameters that distinguish healthy from bad bearings (Figure 5.3). The block diagrams can be found on Appendix F.

After manually analyzing most of the available statistical parameters on the “Statistics” VI (Figure 5.4), it was concluded that, even if no fundamental fault frequencies were detected (not even with the help of envelope detection) the most distinguishable parameters were the variations of the standard deviation from the original signal (faulty bearings present more noise which translates in more spikes as well as a higher amplitude signal) and of the Root Mean Square (RMS) of the FFT in the 5-160Hz interval (since the fundamental frequencies are in this range, faulty bearings have a smaller RMS) with the speed of the anemometer.

Given that most of the data acquisition parameters can be controlled (namely the speed of the anemometer and the position of the accelerometer), this acquisition method was the one chosen to train the ANN model.

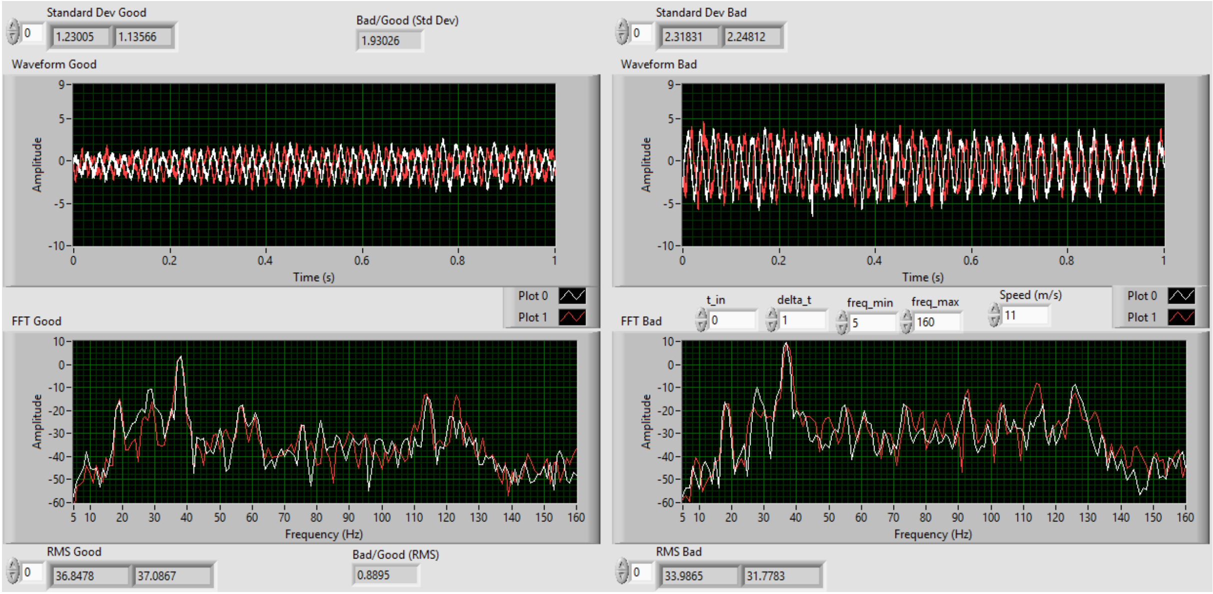


Figure 5.3 – "Accelerometer Data Analysis" interface. The first row of graphs represents the waveform of two 1 second samples of good (left) and faulty (right) bearings, at a speed of 11m/s, along with their standard deviation; the second row represents the FFTs of the same signals between 5 and 160Hz, along with their RMS on this range

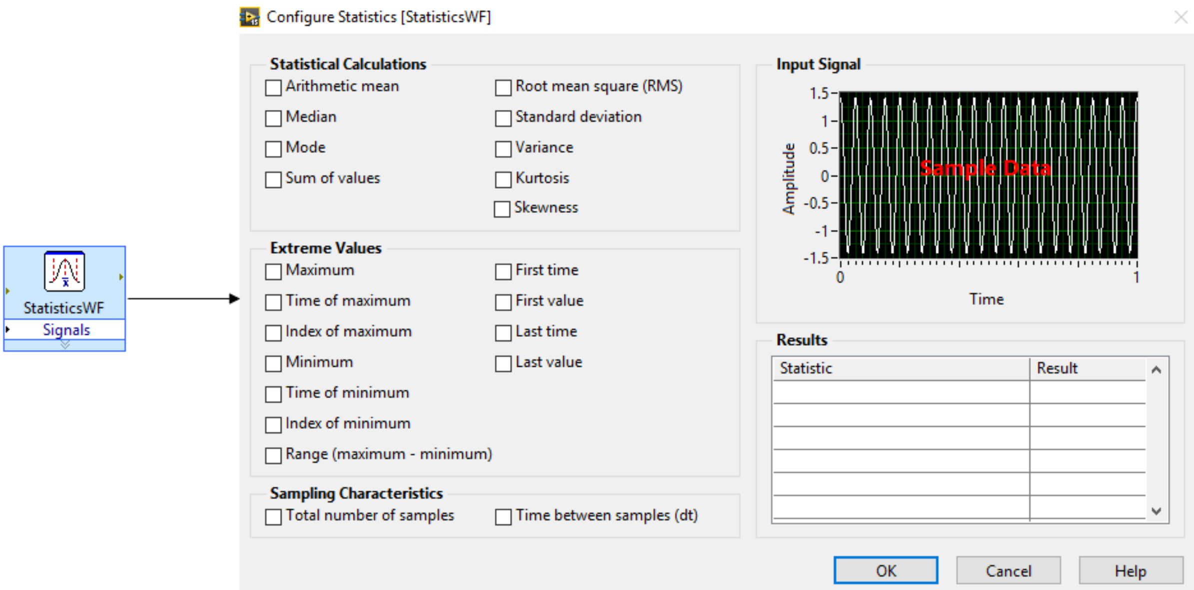


Figure 5.4 - Statistical parameters for waveform analysis

In order to improve the ability of the system to better detect variations within the fundamental frequencies, it would be preferable to use a DAQ with better resolution (e.g. NI 9234, with a resolution of 24 bits).

5.1.3 Microphone on Isolated Environment

Even though this method allowed the distinction of healthy and faulty bearings, it was inherently flawed: the spin induced to the anemometers was done by hand, without a precise control of the momentum applied. In addition, the position of the smartphone's microphone was not the same in all measurements. This prevented the data acquired from these measurements from being used to train an ANN model. Nevertheless, data from a reference anemometer and a noisy one was acquired and analyzed with the help of Audacity (Figure 5.5), an open source audio software [50].

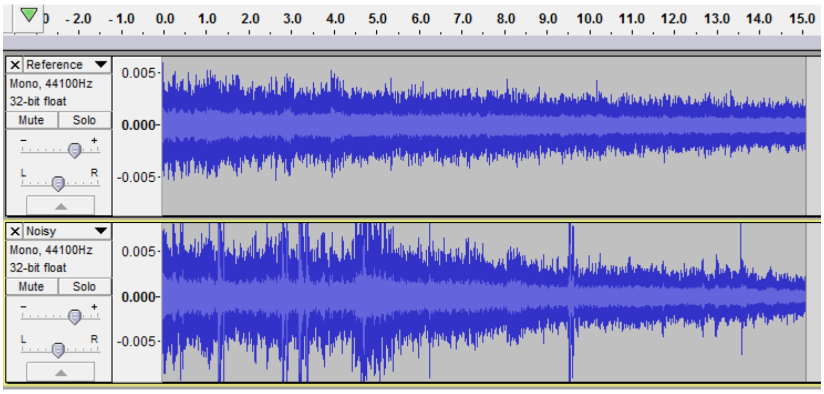


Figure 5.5 - Audio Analysis with Audacity. Waveforms of the audio data samples of 15 seconds from a good (top) and bad (bottom) bearing

Just by analyzing the waveforms obtained, it can be concluded that amplitude of the signal is higher for the noisy anemometer and also has more spikes – corresponding to the bearings.

By understanding that, with a proper method for measuring the data with a microphone in an isolated environment, the data acquired can be easily classifiable, a soundproof box sketch was designed on SolidWorks (Figure 5.6). This box is composed by two parts (connected with the help of four M12x25 ISO 4017 hexagonal head screws [51]), both with an internal coat of soundproofing foam panels [52]. The larger part includes a support for the shaft of the anemometer, which can be fixed with the help of three screws (e.g. M8x16 ISO 4017 hexagonal head screws [51]). This prevents from the head of the anemometer, along with the cups, from rotating. The data acquisition is done with a microphone attached to one of the anemometer's cups.

Considering the low binary needed, as well as an accurate speed control, an adequate choice would be a simple brushed DC motor.

The power transmission can be achieved in multiple ways. The sketch of Figure 5.6 uses a chain transmission between a pulley, connected to the shaft of the motor, and a shaft with the end-shape of the anemometer's plug. Since this is only a first sketch and no selection of a DC motor was done, nor a proper dimensioning of the transmission system, these components on the sketch are mainly illustrative.

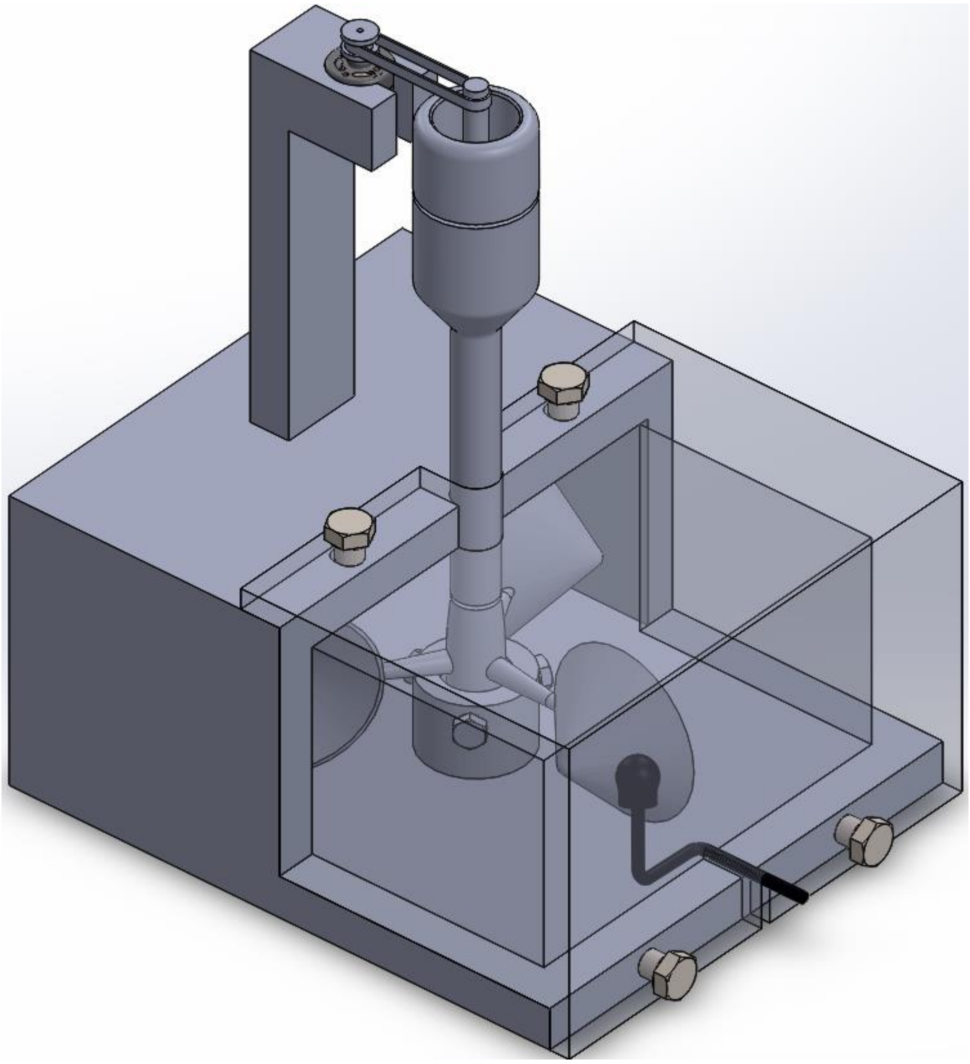


Figure 5.6 - Soundproof box sketch

5.2 Artificial Neural Network Implementation

With the data set selected (the accelerometer's data), it was necessary to implement the learning algorithm. The ANN model used was similar to the one created in section 3.2.3, but this time, to assure the best performance possible, normalization was included. All the VIs used to develop this ANN can be found on Appendix G. The implementation process is described on the following steps, along with the correspondent VI:

1. Split the 30sec data files in 1sec bins and extract the selected features (signal's standard deviation, FFT's RMS and speed) – "Extract Parameters" VI;
2. Extract the normalizing parameters from all the data bins created on step 1 – "Normalization Parameters" VI;
3. Normalize the data bins and create learning and test sets – "Data Normalization and Set Creation" VI
4. Train the ANN model with the learning set – "ANN Learning" VI;
5. Test the model with the test set – "ANN Test" VI.
6. Apply the model to new data – "Apply ANN" VI.

However, due to the shortage of anemometers available for calibration, only data from one reference anemometer and a noisy one was included in the learning process of the model. This may result in a model that doesn't adapt to new data sets, even if the success rate on the test set is high.

In this case, the database consisted on two samples of thirty seconds each at speed from 4m/s to 16m/s with an increment of 1m/s for each anemometer, resulting in 52 samples of thirty seconds. Considering the data is split in one second bins, this results in 1560 samples, where two thirds (1040) were used for learning and the other third (520) for testing.

For the learning process, three hidden nodes were used, with a learning rate of 1 and a regularization parameter of 1. To prevent over-fitting, the training set order is randomized before being implemented in the model training process. Also, the learning process was repeated multiple times, as due to the small learning set, the randomization of the training set order and of the weights' initialization can cause variations on the final value of the cost function.

5.2.1 Optimal Model Selection

To test the variations on the ANN models, 25 ANNs were trained, in batches of 5 with the same data sets. The results are shown in Table 4.

Table 4 - Different ANNs Results

Set	Cost Function	False Healthy (%)	False Noisy (%)	Error (%)
1	0.524	8.85	10.00	9.42
	0.415	4.23	3.08	3.65
	0.526	8.85	10.00	9.42
	0.371	0.77	1.15	0.96
	0.526	8.85	10.00	9.42
2	0.549	6.92	10.00	8.46
	0.379	0.77	1.92	1.35
	0.546	6.92	10.00	8.46
	0.398	1.15	1.92	1.54
	0.373	0.77	1.54	1.15
3	0.542	8.08	11.15	9.62
	0.533	8.08	10.77	9.42
	0.404	1.54	3.08	2.31
	0.379	0.77	1.92	1.35
	0.534	8.08	11.15	9.62
4	0.535	6.92	11.15	9.04
	0.527	6.92	11.54	9.23
	0.389	0.38	2.69	1.54
	0.527	0.38	2.69	1.54
	0.525	6.92	11.92	9.42
5	0.555	4.62	10.38	7.50
	0.422	1.15	1.15	1.15
	0.559	4.62	10.77	7.69
	0.390	0.38	0.77	0.58
	0.386	0.38	0.77	0.58

The optimal model is the one that minimizes the cost function (that represents the difference between the output of the model and the real output), without overfitting to the test data.

By analyzing the table, one can understand that the cost function stabilizes in two different local minima (Figure 5.7): one around 0.53~0.55 and another around 0.37~0.41, which result in different final errors. The cost of the model is also directly related to the error percentage, as one can understand by observing the percentage of errors for the models that stabilize at different local minima.

It can also be seen that, if changing the regularization parameter to zero, the cost function is even lower. However, this translates in an almost exact fit of the function to the training set, which usually results in worse results for new data.

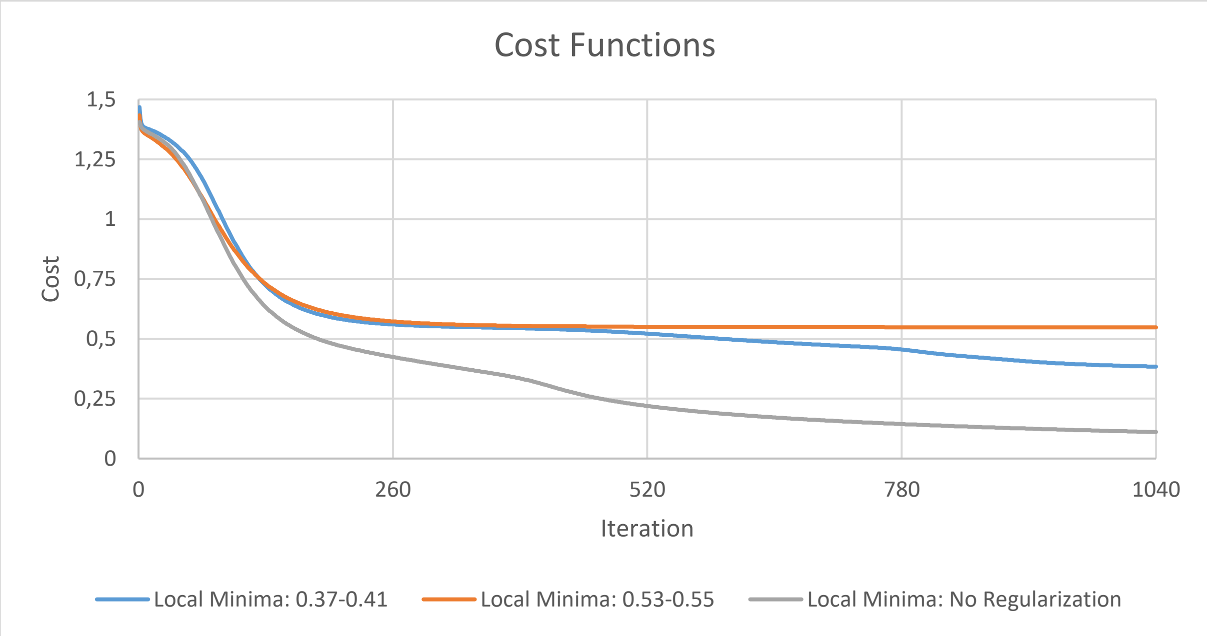


Figure 5.7 - Cost functions at different local minima

From Table 4, one can also analyze the error variations: for the reference anemometer’s data, between 0.38% and 8.85% of the data was misclassified as noisy (depending on the local minima). In the same way, for the noisy anemometer’s data, between 0.77% and 11.92% of the data was misclassified as reference. This results in a final classification error between 0.58% and 9.62%, which is very good considering the size of the learning set – never forgetting that this may be a result of over-fitting.

To guarantee the best results possible, the selected model was one which stabilized at the cost function’s local minima of 0.37~0.41. The results analyzed at the next section were obtained with this model.

5.3 Final Results

Finally, with the help of another VI – “Apply ANN” –, other anemometer’s data was applied to the working model (stabilized at the cost function’s lowest local minima) so that it could classify them. This VI has two parts that run sequentially: the first part extracts the parameters used in the ANN model from the data to be analysed, while the second part implements these same parameters in the ANN, which then classifies them by returning the percentage of bins classified as data from healthy bearings and from faulty bearings.

This VI was used to test five different anemometers, which were rotated by hand and heard beforehand, being categorized as “None” if no distinguishable noise was detected, “Noisy” if uncategorized noises were heard from inside or as “Weary” if the noises sounded as characteristic noises resulting from bearing weariness. Unfortunately, due to the lack of anemometers, no samples from a new anemometer without noise were acquired (although strangely the new anemometer sampled had noises). The results are on Table 5.

Table 5 - Results from Tested Anemometers

Anemometer	Status	Noise	Result
A	New	Noisy	91.31% Noisy
B	Used	Weary	73.93% Noisy
C	Used	Weary	82.15% Noisy
D	Used	None	99.88% Not-noisy
E	Used+Repaired	Noisy	83.45% Noisy

As seen on the previous table, all the anemometers with some kind of noise were classified as bad with a reasonable degree of certainty.

However, there are vulnerabilities present on this model: there are only two possible classifications, and even if the signal acquired doesn’t belong to one of the classes, it is classified nonetheless. As a result, if a measurement is not properly made (e.g. the accelerometer is positioned too far from a faulty bearing, resulting in low amplitude data), the bearing can be misclassified. One way to overcome this would be to use an algorithm capable of detecting outliers, such as One-Class SVM [53].

Another major problem when applying the ANN is the database available for testing. There is no other example of a reference anemometer apart from the one used to train the model. As such, there is no way to determine if the data obtained from another reference anemometer would be similar, even if the used anemometer without noise was classified as “Good” with a 99.88% certainty.

Nevertheless, the ANN was proven to work on three occasions: when detecting different waveforms, when applied to the test set and when classifying a small set of new anemometers. As such, the next step would be to acquire a larger database (with multiple examples from each class) and train an ANN with it.

For further investigational purposes, the data could be separated in 3 classes (“Reference”, “Weary” and “Noisy”), in order to detect what is the influence of each type of noise in the bearings’ status.

6 Conclusions and Future Works

6.1 Conclusions

This dissertation aimed to present a way to classify anemometer's bearings according to the noise produced by them when the anemometer is spinning, using Artificial Intelligent. To do so, multiple data measurement and acquisition devices were used to provide different databases. These methods included the usage of microphones and accelerometers, along with all the data acquisition and processing devices needed. The data acquired was then analysed using LabVIEW VIs along with MATLAB scripts to create an Artificial Neural Network capable of classifying the status of the bearings of each tested anemometer. From the different measurement methods, only one (using an accelerometer inside a wind tunnel) provided meaningful results, albeit with a small-scale database.

To know which frequency range to analyse, the fundamental fault frequencies were calculated for the different rotational speeds, and the maximum frequency value obtained was used to filter the data. Also, techniques such as Envelope Detection were applied to detect the fault frequencies. However, these faults were not detectable, which leads to the conclusion that, even if there are noises in the bearings, they do not translate to fundamental faults. Nevertheless, noisy bearings were classified as faulty while healthy ones had no uncharacteristic noise.

By incorporating Machine Learning in the data acquisition methods, it was possible to develop an Artificial Neural Network capable of classifying bearings as healthy or faulty, using parameters such as the variation of the signal's Standard Deviation and of the FFT's RMS with the rotational speed of the bearing. However, due to the small database set, it is not yet possible to apply this model to a larger data base with reliability.

It is also important to understand that, for the anemometers tested, the influence of the faulty bearings is not yet noticeable in their performance. However, applying a reliable version of the program created during this dissertation eases the detection of the status of each anemometer when working on the field, opening research ground on the influence of the bearings status on the anemometer's performance throughout their operational life. This can be done by taking measurements on the field with an accelerometer and use the data acquired in the model to identify the status of the bearings.

6.2 Future Works

It is important to remember that the database used to develop the model of this dissertation was very small, and it is advisable that, for future works, a larger number of anemometers should be calibrated for learning purposes. This larger database can also be used to increase the number of possible classifications, according to the type of sound present on the bearings, providing the opportunity to detect what is the influence of the different sounds on the bearings' status. Different algorithms could also be used, such as Support Vector Machines, with the possibility of comparing the results of each algorithm and opting for the one with the optimal results. Another way to look at the problem discussed in this dissertation would be to analyse the data as a regression problem regarding the operational life left for the bearings.

Considering the data acquisition module, a system with a higher resolution would be preferable. As such, using a DAQ such as NI 9234, with a resolution of 24 bits, coupled to the accelerometer, would result in an improved capability of detecting small variations of the signal, even though it would represent an investment five times higher.

Regarding acquisition of audible data, with the construction of the isolation box sketched on section 4.4 it would be possible to obtain clear data, even on the field. This, combined with the fact that audible data is easier to analyse, enables non-experts to take conclusions on the type of data they are dealing with.

Finally, a different approach would be the analysis of Acoustic Emission data. This, combined with the possibility of transforming this problem on a regression one, would provide more reliable data throughout the whole operational life of the bearings.

References

[1] M. Roser, "'Technological Progress' Published online at OurWorldInData.org," 2016. [Online]. Available: https://ourworldindata.org/technological-progress/. [Accessed: 12-Jun-2017].

[2] H. K. D. H. Bhadeshia, "Steels for bearings," *Prog. Mater. Sci.*, vol. 57, no. 2, pp. 268–435, 2012.

[3] B. P. Graney and K. Starry, "Rolling element bearing analysis crosses threshold," *Mater. Eval.*, vol. 70, no. 1, pp. 78–85, 2011.

[4] R. Company, "What is the rolling Bearings?," 2002. [Online]. Available: http://www.mygoodstuff.com/bearing.htm. [Accessed: 16-Jun-2017].

[5] D. O'Halloran and E. Kvochko, "Industrial Internet of Things : Unleashing the Potential of Connected Products and Services," *World Econ. Forum*, no. January, p. 40, 2015.

[6] G. Burdeshaw, "Methods For Monitoring Bearing Performance - Maintenance Technology," 2013. [Online]. Available: http://www.maintenancetechnology.com/2013/10/methods-for-monitoring-bearing-performance/. [Accessed: 07-Jun-2017].

[7] P. Maintenance, "Extend warning time and reduce the risk of bearing failure using SKF Acoustic Emission Enveloping."

[8] S. R. Gomes, "Manutenção Industrial - Aula 13 - Análise de Ruído com Estetoscópio - Máquina Ligada," 2016. [Online]. Available: http://manutencaodesistemasindustriais.blogspot.pt/2016/05/manutencao-industrial-aula-13-analise_17.html?m=0. [Accessed: 07-Jun-2017].

[9] S. M, B. Shah, D. . .D, and S. Jana, "A Novel Approach for Bearing Fault Detection and Classification using Acoustic Emission Technique," no. March 2017, 2012.

[10] SKF, "Basic condition monitoring products," 2017. [Online]. Available: http://www.skf.com/ph/products/condition-monitoring/basic-condition-monitoring-products/index.html. [Accessed: 07-Jun-2017].

[11] SKF, "SKF @ ptitude Analyst," 2016.

[12] G. Faults, O. Vibration, C. Systems, C. C. Vibration, and P. Technology, "Bearing Fault Detector," *Sensors (Peterborough, NH)*, no. 6, pp. 682–685.

[13] M. Lucas and Kittiwake, "Acoustic Emission: The Next Generation of Vibration Techniques," 2012. [Online]. Available: http://www.reliableplant.com/Read/28771/acoustic-emission-techniques. [Accessed: 14-Jun-2017].

[14] N. Hiremath and D. M. Reddy, "Bearing fault detection using acoustic emission signals analyzed by empirical mode decomposition," *Int. J. Res. Eng. Technol.*, vol. 3, no. 3, pp.

426–431, 2014.

[15] H. Instruments, "How does Acoustic Emission (AE) compare with Vibration?," p. 2006, 2006.

[16] Y. Qu, E. Bechhoefer, D. He, and J. Zhu, "A New Acoustic Emission Sensor Based Gear Fault Detection Approach," *Int. J. Progn. Heal. Manag.*, vol. 4, pp. 1–14, 2013.

[17] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, vol. 2016–March, no. July, pp. 3928–3937.

[18] R. Play, I. Clearance, and B. Bearings, "Radial Play (Internal Clearance) in Ball Bearings," no. 800, 2010.

[19] S. Tyagi and S. K. Panigrahi, "An improved envelope detection method using Particle Swarm Optimisation for rolling element bearing fault diagnosis," *J. Comput. Des. Eng.*, 2017.

[20] S. Abbasion, A. Rafsanjani, A. Farshidianfar, and N. Irani, "Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine," *Mech. Syst. Signal Process.*, vol. 21, no. 7, pp. 2933–2945, 2007.

[21] V. Vakharia, V. K. Gupta, and P. K. Kankar, "Ball Bearing Fault Diagnosis Using Supervised and Unsupervised Machine Learning Methods Ball Bearing Fault Diagnosis Using Supervised and Unsupervised Machine Learning Methods," vol. 20, no. February 2016, 2015.

[22] N. Tondon and A. Choudhury, "A review of vibration and acoustics measurement methods for the detection of defects in rolling element bearing," *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, 1999.

[23] J. J. Carr and J. M. (John M. Brown, *Introduction to biomedical equipment technology*. Prentice Hall, 1998.

[24] J. Garcia, "Dynamic vs. Ribbon vs. Condenser," 2013. [Online]. Available: http://criticalrecordingstudio.com/blog/dynamic-vs-ribbon-vs-condenser/. [Accessed: 18-May-2017].

[25] Yamaha Corporation, "Microphone types | PA Beginners Guide | Self Training | Training & Support | Yamaha," 2017. [Online]. Available: http://www.yamahaproaudio.com/global/en/training_support/selftraining/pa_guide_beginner/microphone/. [Accessed: 26-May-2017].

[26] A. . Fallis, *Handbook of Modern Sensors*. 2004.

[27] M. Huang, L. Jiang, P. K. Liaw, C. R. Brooks, R. Seele, and L. Dwaine, *Using Acoustic Emission in Fatigue and Fracture Materials Research*. 1998.

[28] D. Mba and R. Rao, "Development of Acoustic Emission Technology for Condition Monitoring and Diagnosis of Rotating Machines; Bearings, Pumps, Gearboxes, Engines and Rotating," *Shock Vib. Dig.*, vol. 38, no. 1, pp. 3–16, 2006.

[29] Z. Elman, "Acoustic Emission Apparatus and Data Acquisition," 2011.

[30] National Instruments, "How to Choose the Right DAQ Hardware for Your Measurement System - National Instruments," 2016. [Online]. Available: http://www.ni.com/white-paper/13655/en/. [Accessed: 27-May-2017].

[31] J. Mccarthy, "What is Artificial Intelligence ?," 2007.

[32] IBM, "IBM - What is big data?," 2017. [Online]. Available: https://www-

01.ibm.com/software/data/bigdata/what-is-big-data.html. [Accessed: 28-May-2017].

[33] T. M. Mitchell, *Machine Learning*. 1997.

[34] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. 2012.

[35] P. Domingos, "A Few Useful Things to Know About Machine Learning," *78 Commun. acm Commun. ACm*, vol. 55, no. 79, 2012.

[36] D. Nag, "Machine Learning = Representation + Evaluation + Optimization," 2016. [Online]. Available: https://medium.com/@devnag/machine-learning-representation-evaluation-optimization-fc7b26b38fdb. [Accessed: 28-May-2017].

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 13, no. 1. MIT Press, 2016.

[38] T. O. Ayodele, "Types of Machine Learning Algorithms," *New Adv. Mach. Learn.*, pp. 19–49, 2010.

[39] S. Raschka, *Python Machine Learning*, 1st ed. Birmingham, UK: Packt Publishing, 2015.

[40] S. Raschka, "What is the Role of the Activation Function in a Neural Network?," 2016. [Online]. Available: http://www.kdnuggets.com/2016/08/role-activation-function-neural-network.html. [Accessed: 30-May-2017].

[41] R. S. Sutton and A. G. Barto, "Reinforcement learning : an introduction," *Neural Networks IEEE Trans.*, vol. 9, no. 5, p. 1054, 2013.

[42] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, vol. 9, no. 2. 1995.

[43] National Instruments, "LabVIEW - National Instruments," 2017. [Online]. Available: http://www.ni.com/en-us/shop/labview.html. [Accessed: 13-Jun-2017].

[44] A. Y. Ng, "Machine Learning (free online course on Coursera)," 2017. [Online]. Available: https://www.coursera.org/learn/machine-learning. [Accessed: 31-May-2017].

[45] D. Graupe, *Principles of Artificial Neural Networks*. 2007.

[46] A. Y. Ng, "Unsupervised Feature Learning and Deep Learning Tutorial: Multi-layer Neural Network." [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/. [Accessed: 31-May-2017].

[47] T. Clima, "thiesclima.com - Products," 2017. [Online]. Available: https://www.thiesclima.com/en/Products/Wind-First-Class/. [Accessed: 12-Jun-2017].

[48] GSMArena, "Xiaomi Redmi Note 3 - Full phone specifications," 2016. [Online]. Available: http://www.gsmarena.com/xiaomi_redmi_note_3-7863.php. [Accessed: 12-Jun-2017].

[49] INEGI, "INEGI accredited for the calibration of anemometers," 2015. [Online]. Available: http://www.inegi.up.pt/instituicao/noticias_detalhe.asp?ano=2015&idm=1&idsubm=9&id=32&noticiaid=711&LN=EN. [Accessed: 03-Jun-2017].

[50] Audacity Team, "Audacity® | Free, open source, cross-platform audio software for multi-track recording and editing.," 2017. [Online]. Available: http://www.audacityteam.org/. [Accessed: 12-Jun-2017].

[51] S. Morais, *Desenho Técnico Básico*, 25th ed. 2007.

[52] Soundproofingtips.com, “Top 10 Soundproofing Materials » Soundproofing Tips,” 2017. [Online]. Available: http://www.soundproofingtips.com/soundproofing-materials/. [Accessed: 12-Jun-2017].

[53] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, and Sckit-learn, “Estimating the Support of a High-Dimensional Distribution,” *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

Appendix A *Thies™* Wind Transmitter "First Class" Advanced

Summary of

Cup Anemometer Classification

According to IEC 61400-12-1 (2005-12) Classification Scheme

Description of Anemometer

Manufacturer: Adolf Thies GmbH&Co.KG
Hauptstrasse 76
37083 Göttingen

Identification: 4.3351.00.000; SN.0806002
4.3351.00.000; SN 0806005
Thies First Class Advanced

Dimension:

Body diameter:	50 mm;	Body length:	95 mm
Total length:	290 mm;	Shaft diameter:	18 mm
Top:	38 mm		
Rotor diameter:	240 mm;	Cup diameter:	80 mm
Cup tilt angle:	3.5 deg;	Flaps (approx):	28 x 31 mm

Reference:

Deutsche WindGuard Wind Tunnel Services GmbH AK 08 1662.01
Measuring period: 09.2007 – 05.2008
Test site: Varel, Germany
Wind Tunnel: Deutsche WindGuard GmbH, Varel

Procedure:

The classification is based on numerical integration of the differential equation which describes the response of a cup anemometer to fluctuating wind speeds. The chosen spectra of the wind speed time series was a *Kaimal* spectrum for non-isotropic condition (for Class A classification) and a *von Karman* spectrum for isotropic conditions (for class B classification). The time series have been generated with a software tool provided by Risø - National Laboratory, Denmark. Other parameters which influence the response of an anemometer in fluctuating wind conditions are:

- Of axis response for different tilt angles
- Friction changes in bearings due different ambient temperatures
- Driving and braking torque of the cups during rotation
- Inertia of the rotor
- Air density

All relevant parameters have been measured in the wind tunnel of Deutsche WindGuard GmbH. The driving and braking forces used in the numerical model have been derived from the measured step response of the tested anemometer and a cup torque model.

In addition, results of the field comparison are presented in this summary.

Ammonit

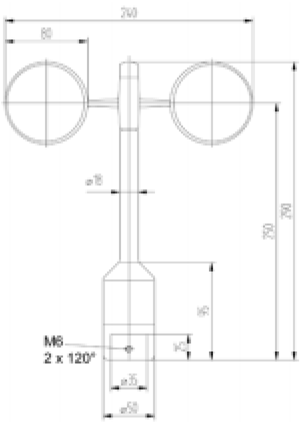
Anemometer Thies First Class Advanced S11100 / S11100H

Accredited according to: IEC 61400-12-1 (2005-12), CLASS A 0.9, B 3.0 & S 0.5
MEASNET, ISO 17713-1, CLASSCUP

Classification:
IEC 61400-12-1 (2005-12)
Class A Classification Index A 0.9
Class B Classification Index B 3.0
Class S Classification Index S 0.5

Linearity (MEASNET):
r > 0.999 99 (4 ... 20 m/s)

Dimensional drawing



Optically-scanned cup anemometer

The anemometer Thies First Class Advanced gives outstanding performance. It is the only anemometer on the market that complies with all the requirements of IEC 61400-12-1 (2005-12), Class S 0.5. Its performance ratings have even been improved on the previous Thies First Class anemometer, which was rated the best of its kind according to the CLASSCUP / ACCUWIND Study, (Risø-R-1563-EN, Table 4-4).

It gives optimal dynamic performance with the following characteristics:

- High accuracy
- Minimal deviation from cosine line
- Excellent behaviour to turbulences
- Minimum overspeeding
- Small distance constant
- Low start up value
- Low power consumption
- Digital output

Measurement of power curves and site assessment reports are the main tasks for this instrument. The patented design is the result of long testing in the wind tunnel. The anemometer requires only low maintenance thanks to its low-inertia and ball-bearing cup star.

The sensor is designed for measuring the horizontal wind velocity in the fields of meteorology, climate research, site assessment, and the measurement of capacity characteristics of wind power systems (power curves).

For winter operation the anemometer is equipped with electronically regulated heating to guarantee smooth running of the ball bearings and prevent icing of shaft and slot.

Calibration

Wind speed is determined by the linear function of the frequency output f:

wind speed [m/s] = slope [m] × f [Hz] + offset [m/s]

Manufacturers instructions: Slope = 0.046 m, Offset = 0.21 m/s

For wind site assessment, anemometers have to be calibrated according to MEASNET in a wind tunnel to achieve highest accuracy. After calibration, use slope and offset values according to the calibration protocol.

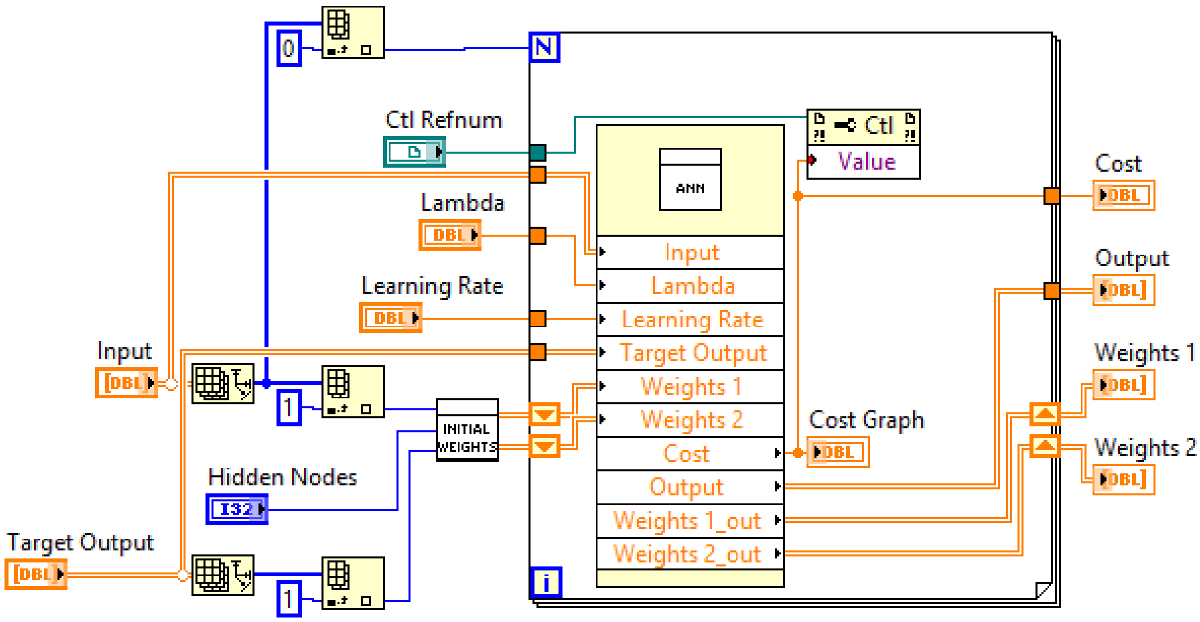
Comparison of performance of anemometers

Cup anemometer	Class A	Class B	
NRG Class 1*	1.01	8.44	Information as stated acc. to CLASSCUP & ACCUWIND Study (Table 4-4 horizontal wsp definition Risø R-1563-EN)
Risø P2546**	1.31	6.18	
Vaisala WAA151	1.7	11.1	
Vector L100	1.8	4.5	
Thies First Class	1.5	2.9	
Thies First Class Advanced	0.9	3.0	IEC 61400-12-1 (2005-12) according to Deutsche WindGuard

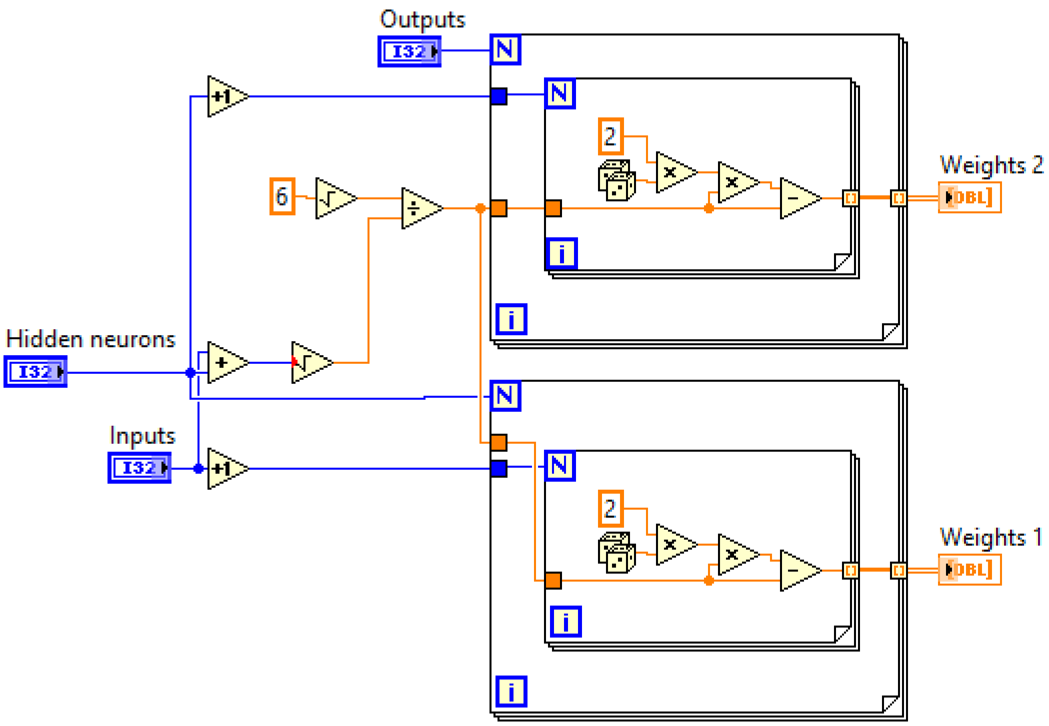
* NRG Class 1 is a ball bearing version of NRG #40C. In 2012 DTU Wind Energy Department classified the anemometer acc. to ACCUWIND and IEC 61400-12-1.
** Classification from 2004 performed in FDI-LT5 wind tunnel, which is not MEASNET-certified.

Ammonit
Measurement GmbH
Wrangelstrasse 100, D-10997 Berlin, Germany
T: +49 30 6003188-0, E: info@ammonit.com
measuring wind and solar power
www.ammonit.com

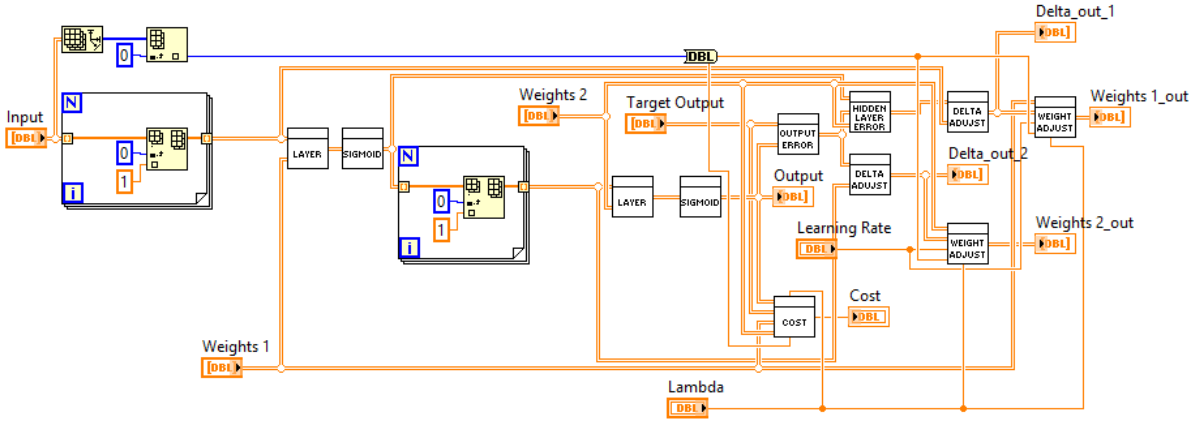
Appendix B "Test ANN" Sub-VIs



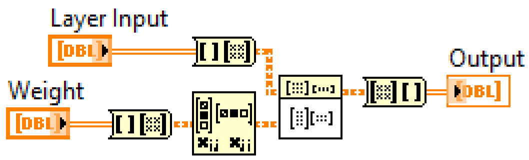
"Create ANN LV" Block Diagram



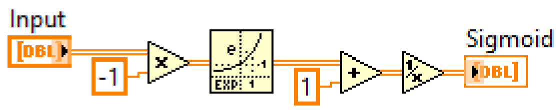
"Initial Weights" Block Diagram



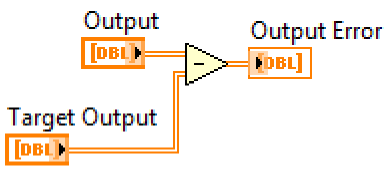
"ANN" Block Diagram



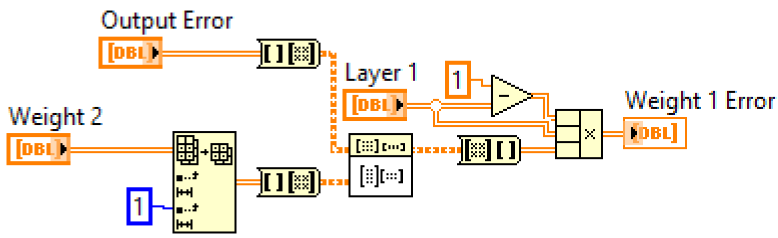
"Layer" Block Diagram



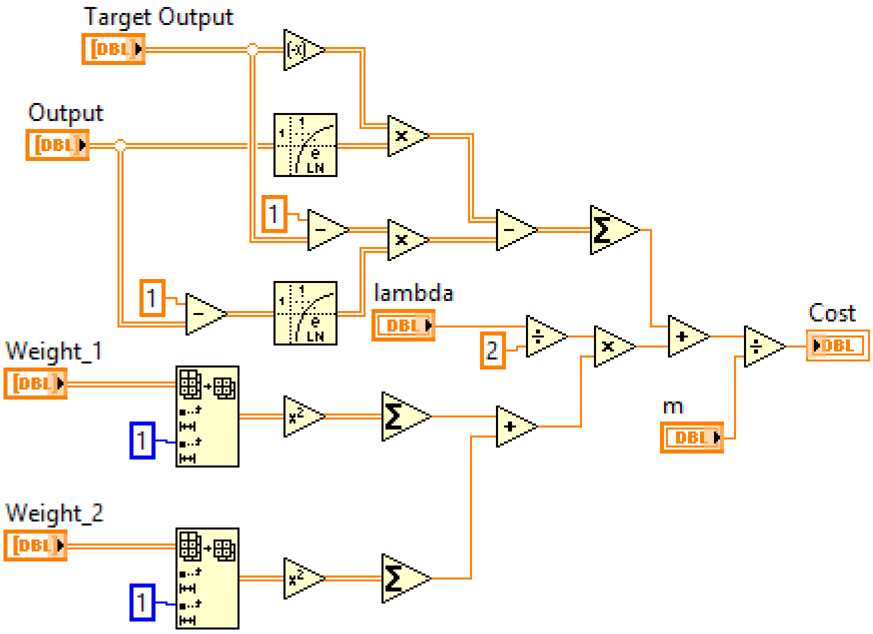
"Sigmoid" Block Diagram



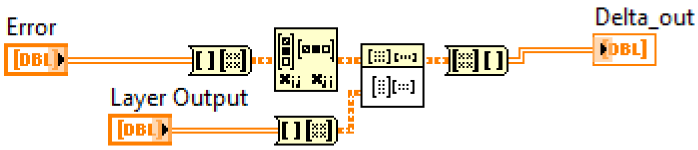
"Output Error" Block Diagram



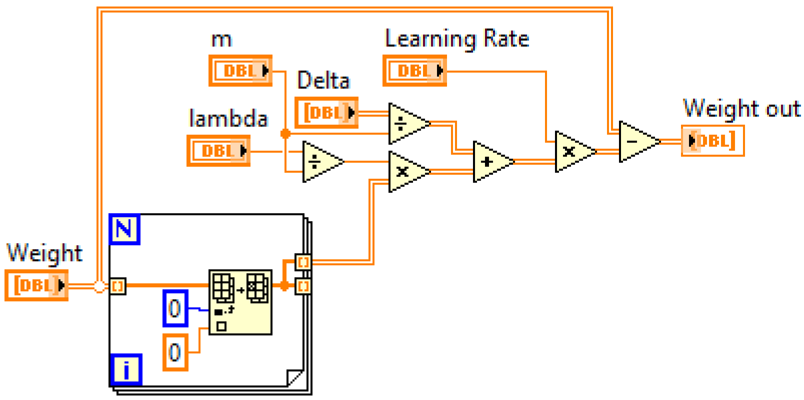
"Hidden Layer Error" Block Diagram



"Cost Function" Block Diagram



"Delta Adjust" Block Diagram



"Weight Adjust" Block Diagram

Appendix C PCB 352A24 Accelerometer

Model Number 352A24	ICP® ACCELEROMETER	Revision: C ECN #: 27171

Performance	ENGLISH	SI	
Sensitivity(± 10 %)	100 mV/g	10.2 mV/(m/s²)	
Measurement Range	± 50 g pk	± 490 m/s² pk	
Frequency Range(± 5 %)	1.0 to 8000 Hz	1.0 to 8000 Hz	
Frequency Range(± 10 %)	0.8 to 10,000 Hz	0.8 to 10,000 Hz	
Frequency Range(± 3 dB)	0.4 to 12,000 Hz	0.4 to 12,000 Hz	
Resonant Frequency	≥ 30 kHz	≥ 30 kHz	
Broadband Resolution(1 to 10,000 Hz)	0.0002 g rms	0.002 m/s² rms	[1]
Non-Linearity	≤ 1 %	≤ 1 %	[2]
Transverse Sensitivity	≤ 5 %	≤ 5 %	
Environmental			
Overload Limit(Shock)	± 5000 g pk	± 49,050 m/s² pk	
Temperature Range(Operating)	-65 to +250 °F	-54 to +121 °C	
Temperature Response	See Graph	See Graph	
Electrical			
Excitation Voltage	18 to 30 VDC	18 to 30 VDC	
Constant Current Excitation	2 to 20 mA	2 to 20 mA	
Output Impedance	≤ 300 ohm	≤ 300 ohm	
Output Bias Voltage	8 to 12 VDC	8 to 12 VDC	
Discharge Time Constant	0.4 to 1.5 sec	0.4 to 1.5 sec	
Settling Time(within 10% of bias)	<8 sec	<8 sec	
Spectral Noise(1 Hz)	80 µg/√Hz	785 (µm/s²)/√Hz	
Spectral Noise(10 Hz)	15 µg/√Hz	147 (µm/s²)/√Hz	[1]
Spectral Noise(100 Hz)	4 µg/√Hz	39 (µm/s²)/√Hz	[1]
Spectral Noise(1 kHz)	1 µg/√Hz	9.8 (µm/s²)/√Hz	[1]
Electrical Isolation(Base)	≥ 10^8 ohm	≥ 10^8 ohm	
Physical			
Sensing Element	Ceramic	Ceramic	
Sensing Geometry	Shear	Shear	
Housing Material	Anodized Aluminum	Anodized Aluminum	
Sealing	Epoxy	Epoxy	
Size (Height x Length x Width)	0.19 in x 0.48 in x 0.28 in	4.8 mm x 12.2 mm x 7.1 mm	
Weight	0.03 oz	0.8 gm	[1]
Electrical Connector	3-56 Coaxial Jack	3-56 Coaxial Jack	
Electrical Connection Position	Side	Side	
Mounting	Adhesive	Adhesive	

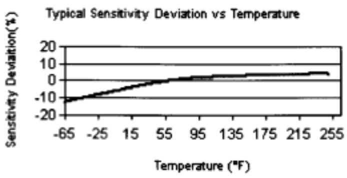
OPTIONAL VERSIONS

Optional versions have identical specifications and accessories as listed for the standard model except where noted below. More than one option may be used.

RH - RoHS Compliant
Supplied Accessory : Model RH030A10 Coax Cable, 10 ft (3 m), 3-56 plug to 10-32 plug, RoHS compliant. replaces Model 030A10

NOTES:
[1] Typical.
[2] Zero-based, least-squares, straight line method.
[3] See PCB Declaration of Conformance PS023 for details.

CE [3]



SUPPLIED ACCESSORIES:
Model 030A10 Coax Cable, 10 ft (3 m), 3-56 plug to 10-32 plug. (1)
Model 039A28 Removal Tool (1)
Model 080A109 Petro Wax (1)
Model ACS-1 NIST traceable frequency response (10 Hz to upper 5% point). (1)

Entered: JH	Engineer: EB	Sales: KJL	Approved: BM	Spec Number:
Date: 8-23-07	Date: 8-23-07	Date: 8-23-07	Date: 8-23-07	12320

All specifications are at room temperature unless otherwise specified.
In the interest of constant product improvement, we reserve the right to change specifications without notice.
ICP® is a registered trademark of PCB Group, Inc.

PCB PIEZOTRONICS™
VIBRATION DIVISION
3425 Walden Avenue, Depew, NY 14043

Phone: 716-684-0001
Fax: 716-685-3886
E-Mail: vibration@pcb.com

Appendix D NI™ USB-6009 DAQ Device

NATIONAL INSTRUMENTS

Technical Sales

(866) 531-6285
orders@ni.com

Requirements and Compatibility | Ordering Information | Detailed Specifications

For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2014-11-06 07:14:12.0

Low-Cost, Bus-Powered Multifunction DAQ for USB

12- or 14-Bit, Up to 48 kS/s, 8 Analog Inputs

- 8 analog inputs at 12 or 14 bits, up to 48 kS/s
- 2 analog outputs at 12 bits, software-timed
- 12 TTL/CMOS digital I/O lines
- One 32-bit, 5 MHz counter
- Digital triggering
- Bus-powered
- 1-year warranty

Overview

With recent bandwidth improvements and new innovations from National Instruments, USB has evolved into a core bus of choice for measurement applications. The NI USB-6008 and USB-6009 are low-cost DAQ devices with easy screw connectivity and a small form factor. With plug-and-play USB connectivity, these devices are simple enough for quick measurements but versatile enough for more complex measurement applications.

Back to Top

Requirements and Compatibility

OS Information

- Mac OS X
- Windows 2000/XP
- Windows 7
- Windows CE
- Windows Mobile
- Windows Vista 32-bit
- Windows Vista 64-bit

Driver Information

- NI-DAQmx
- NI-DAQmx Base

Software Compatibility

- ANSI C/C++
- LabVIEW
- LabWindows/CVI
- Measurement Studio
- SignalExpress
- Visual Basic .NET
- Visual C#

Back to Top

Comparison Tables

Product	Analog Inputs	Input Resolution	Max Sampling Rate (kS/s)	Analog Outputs	Output Resolution	Output Rate (Hz)	Digital I/O Lines	32-Bit Counter	Triggering
USB-6008	8 single-ended/4 differential	12	10	2	12	150	12	1	Digital
USB-6009	8 single-ended/4 differential	14	48	2	12	150	12	1	Digital

Back to Top

Appendix E PCB Model 480E09 Signal Conditioner

Model Number 480E09	BATTERY-POWERED SIGNAL CONDITIONER	Revision: V ECN #: 45339

Performance	ENGLISH	SI	
Channels	1	1	
Frequency Range(-5 %)(x1, x10 Gain)	0.15 to 100,000 Hz	0.15 to 100,000 Hz	[6]
Frequency Range(-10 %)(x100 Gain)	0.15 to 50,000 Hz	0.15 to 50,000 Hz	[7]
Voltage Gain(± 2 %)	1:1	1:1	
Voltage Gain(± 2 %)	1:10	1:10	
Voltage Gain(± 2 %)	1:100	1:100	
Fault/Bias Monitor/Meter(± 1 V)(midscale)	13 VDC	13 VDC	
Environmental			
Temperature Range	32 to 120 °F	0 to 50 °C	
Electrical			
Excitation Voltage(To Sensor)	25 to 29 VDC	25 to 29 VDC	[1]
Constant Current Excitation(To Sensor)	2.0 to 3.2 mA	2.0 to 3.2 mA	[2]
Discharge Time Constant	>7 sec	>7 sec	[3][4]
DC Offset(Maximum)	<30 mV	<30 mV	[3]
Spectral Noise(1 Hz)(Gain 1)	.25 μV/√Hz	-132 dB	
Spectral Noise(10 Hz)(Gain 1)	.07 μV/√Hz	-143 dB	
Spectral Noise(100 Hz)(Gain 1)	.05 μV/√Hz	-146 dB	
Spectral Noise(1 kHz)(Gain 1)	.04 μV/√Hz	-148 dB	
Spectral Noise(10 kHz)(Gain 1)	.03 μV/√Hz	-150 dB	
Broadband Electrical Noise(1 to 10,000 Hz)(Gain x1)	3.25 μV rms	-110 dB/rms	
Spectral Noise(1 Hz)(Gain 10)	2.2 μV/√Hz	-113 dB	
Spectral Noise(10 Hz)(Gain 10)	2.0 μV/√Hz	-114 dB	
Spectral Noise(100 Hz)(Gain 10)	1.1 μV/√Hz	-119 dB	
Spectral Noise(1 kHz)(Gain 10)	.55 μV/√Hz	-125 dB	
Spectral Noise(10 kHz)(Gain 10)	.3 μV/√Hz	-130 dB	
Broadband Electrical Noise(1 to 10,000 Hz)(Gain x10)	49 μV/rms	-86 dB/rms	
Spectral Noise(1 Hz)(Gain 100)	20 μV/√Hz	-94 dB	
Spectral Noise(10 Hz)(Gain 100)	19 μV/√Hz	-94 dB	
Spectral Noise(100 Hz)(Gain 100)	12 μV/√Hz	-98 dB	
Spectral Noise(1 kHz)(Gain 100)	5.5 μV/√Hz	-105 dB	
Spectral Noise(10 kHz)(Gain 100)	2 μV/√Hz	-114 dB	
Broadband Electrical Noise(1 to 10,000 Hz)(Gain x100)	569 μV/rms	-65 dB/rms	
Power Required(Standard)	Internal Battery	Internal Battery	
Internal Battery(Type)	9V	9V	
Battery Life(Standard Alkaline)	50 hours	50 hours	
Power Required(Alternate)	DC power	DC power	
DC Power	15 mA	15 mA	[5]
Internal Battery(Quantity)	3	3	
DC Power	18 to 30 VDC	18 to 30 VDC	[5][1]
Physical			
Electrical Connector(Input, sensor)	BNC Jack	BNC Jack	
Electrical Connector(Output, scope)	BNC Jack	BNC Jack	
Electrical Connector(External Power, DC)	3.5 mm Diameter Miniature Jack	3.5 mm Diameter Miniature Jack	
Electrical Connector(Battery Charger)	#722 Switchcraft Jack	#722 Switchcraft Jack	
Size (Depth x Height x Width)	2.4 in x 4.0 in x 2.9 in	6.1 cm x 10 cm x 7.4 cm	
Weight(Including Batteries)	0.7 lb	0.3 kg	

OPTIONAL VERSIONS

Optional versions have identical specifications and accessories as listed for the standard model except where noted below. More than one option may be used.

NOTES:

[1]Excitation voltage to sensor limited by optional DC power voltage.
[2]Through internal current limiting regulator.
[3]With 1M ohm load.
[4]Un-buffered output, read out device input impedance affects discharge time constant and low frequency response of unit.
[5]Provided by optional external DC power supply.
[6]Low frequency response specified into 1M ohm load.
[7]After Serial Number 24699, previously HFR was 100kHz.
[8]See PCB Declaration of Conformance PS024 for details.

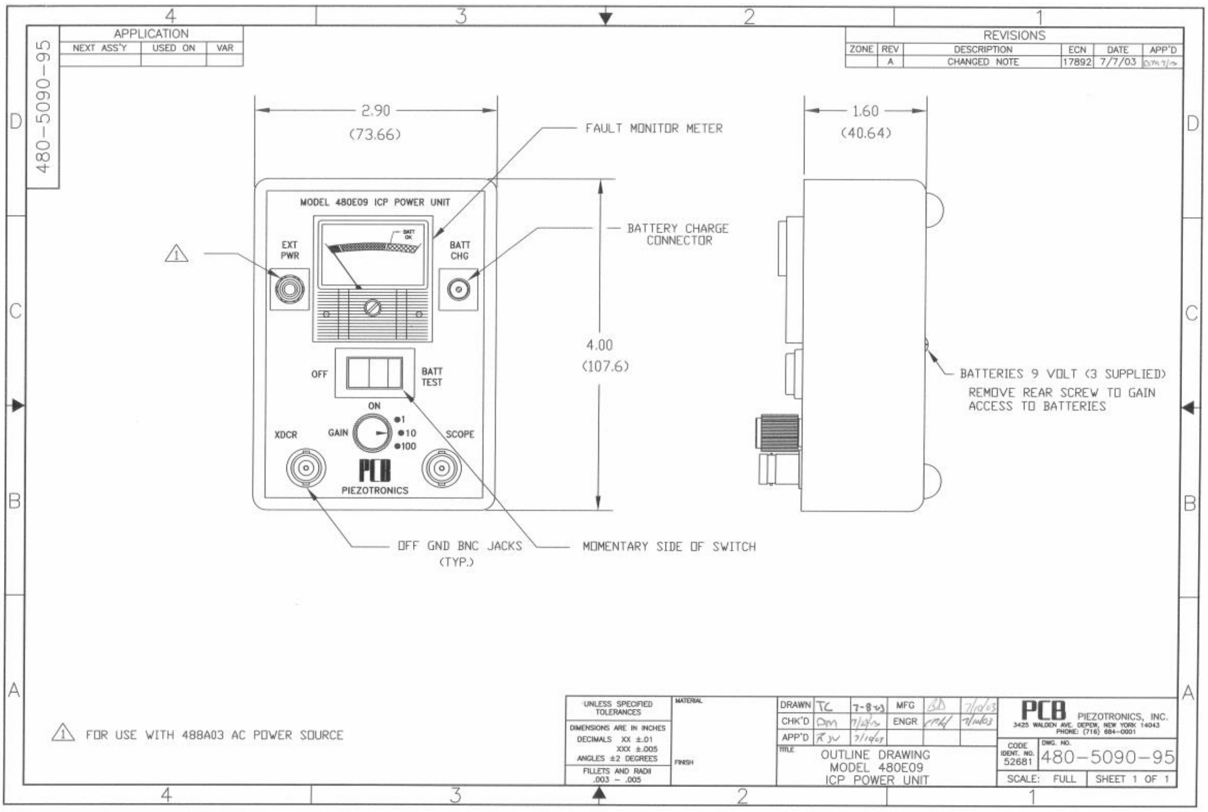
Entered: LK	Engineer: CPH	Sales: ML	Approved: JWH	Spec Number:
Date: 5/23/2016	Date: 5/23/2016	Date: 5/23/2016	Date: 5/23/2016	480-5090-80

CE [8]

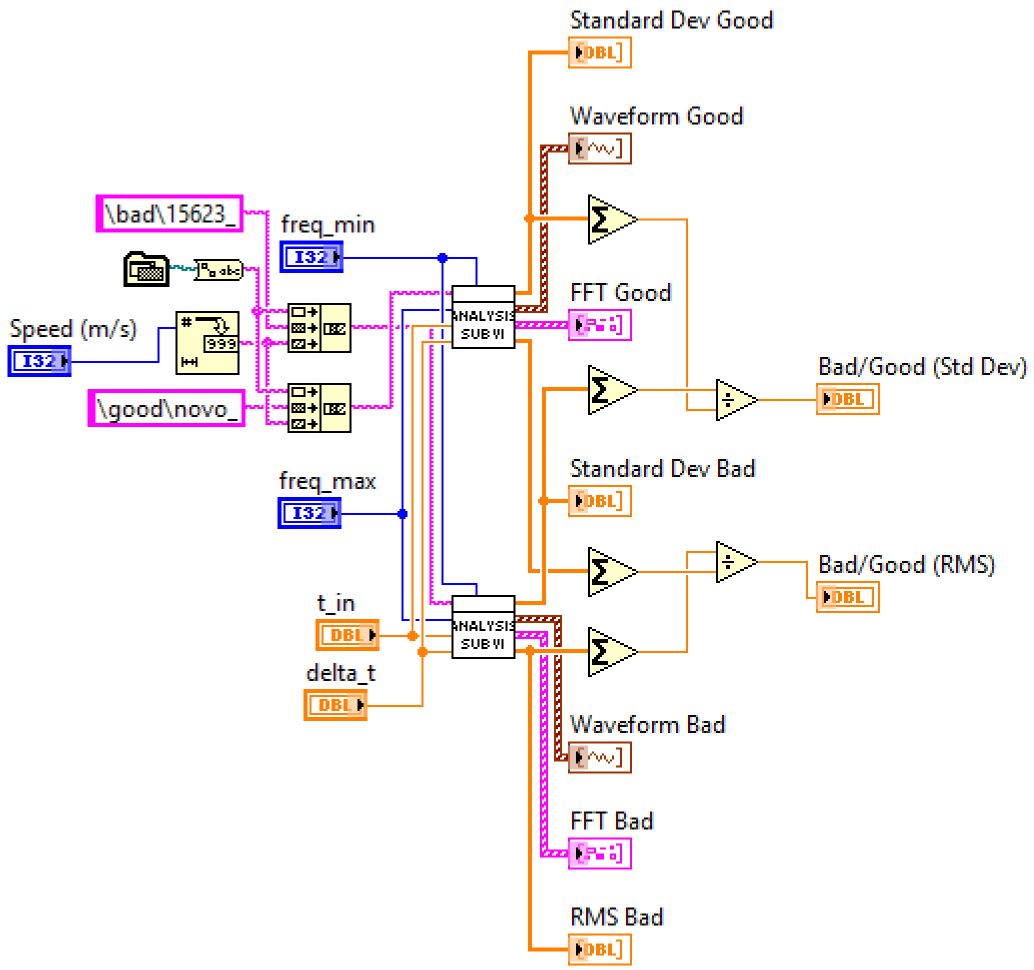
All specifications are at room temperature unless otherwise specified.
In the interest of constant product improvement, we reserve the right to change specifications without notice.
ICP® is a registered trademark of PCB Group, Inc.

PCB PIEZOTRONICS
3425 Walden Avenue, Depew, NY 14043

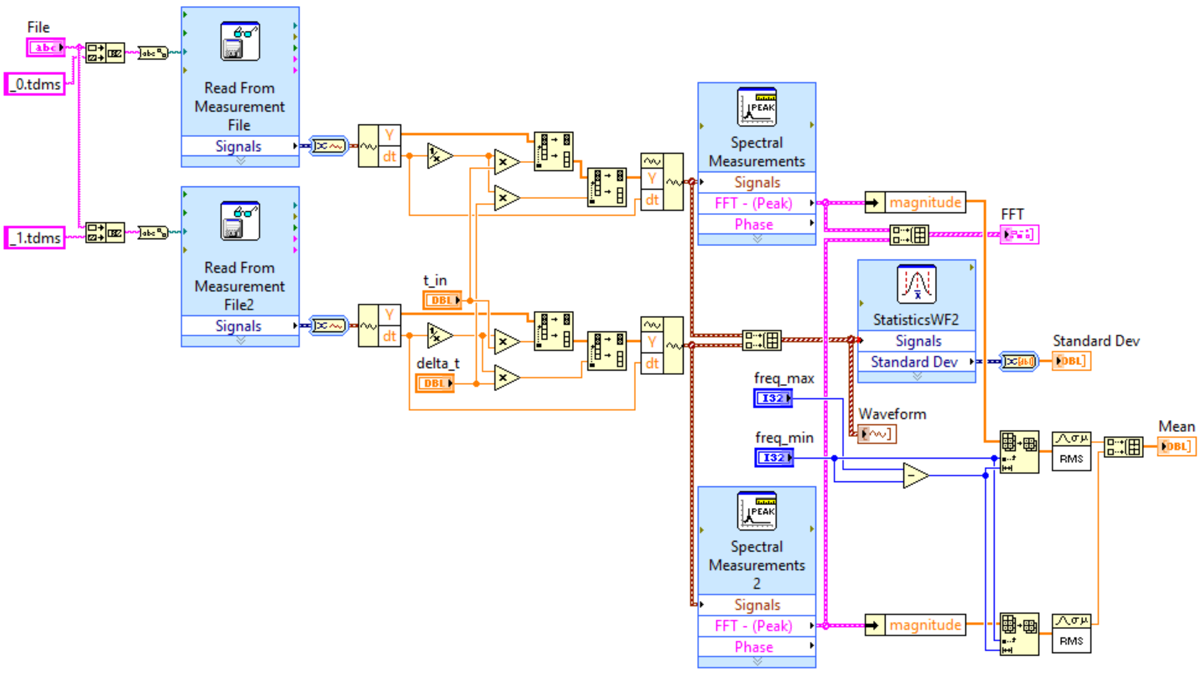
Phone: 716-684-0001
Fax: 716-684-0987
E-Mail: info@pcb.com



Appendix F Accelerometer's Data ANN VIs

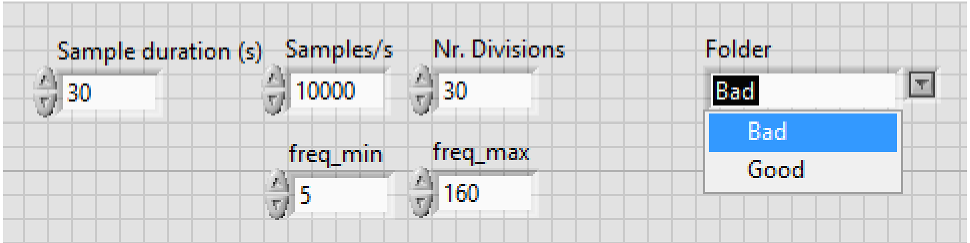


Accelerometer DAQ VI

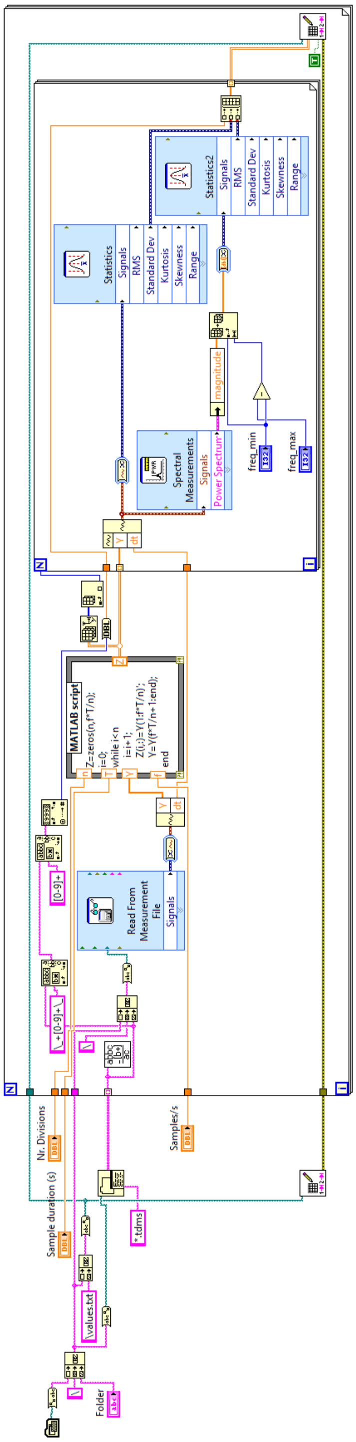


Analysis Sub-VI Interface

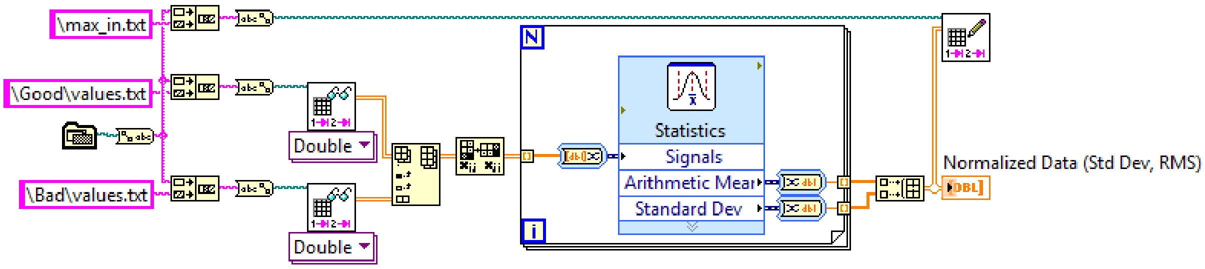
Appendix G ANN Final Model VIs



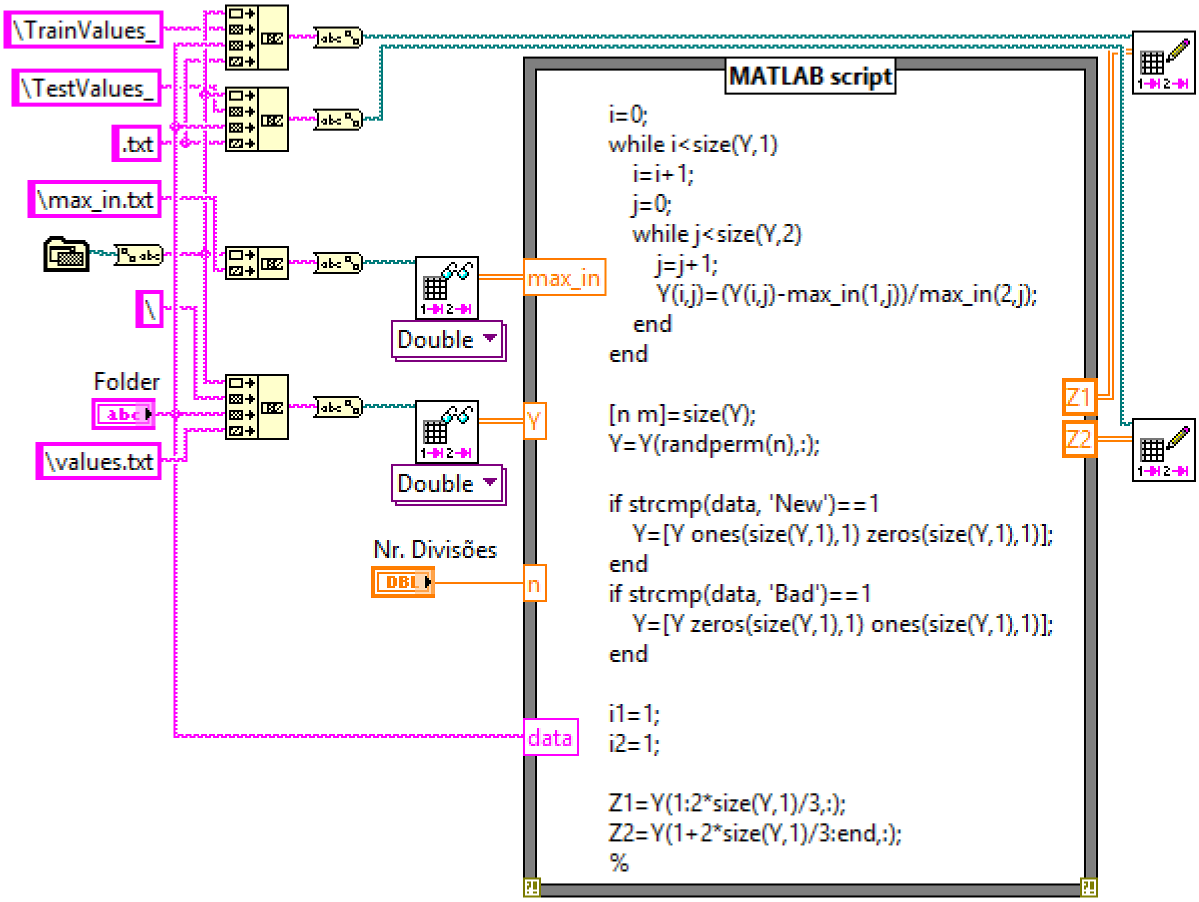
“Extract Parameters” Interface



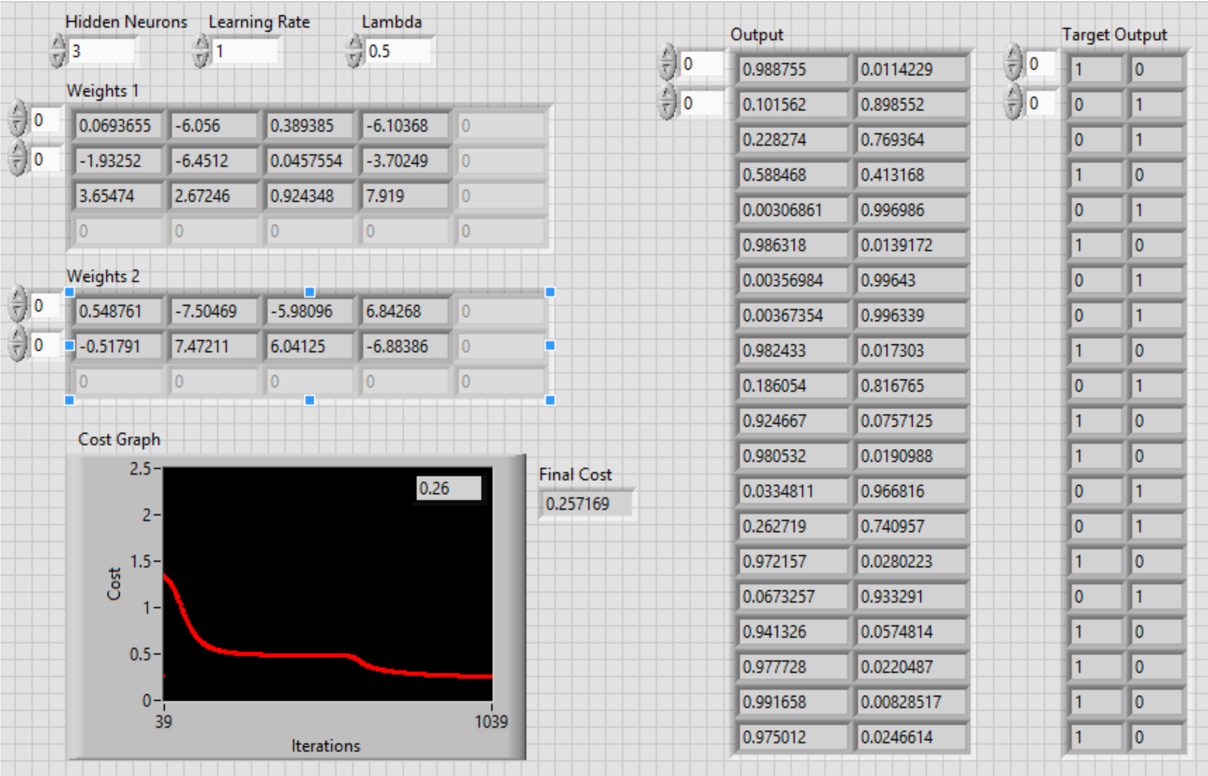
“Extract Parameters” Block Diagram



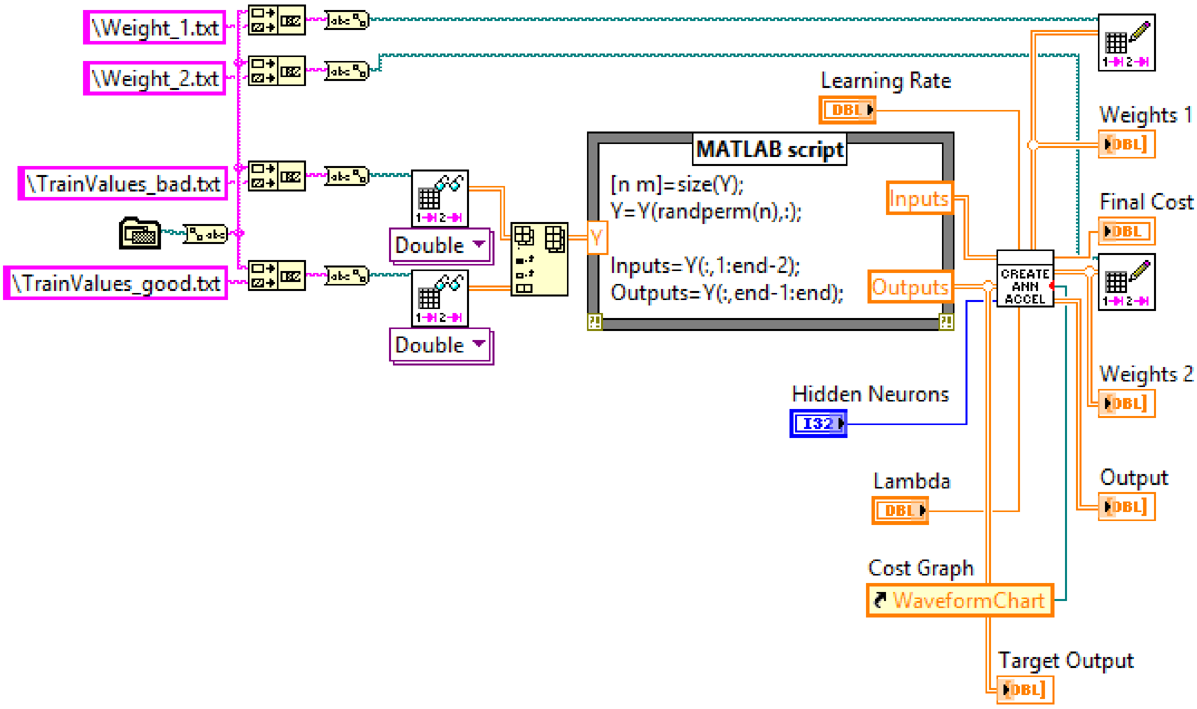
"Normalization Parameters" Block Diagram



"Data Normalization and Sets Creation" Block Diagram

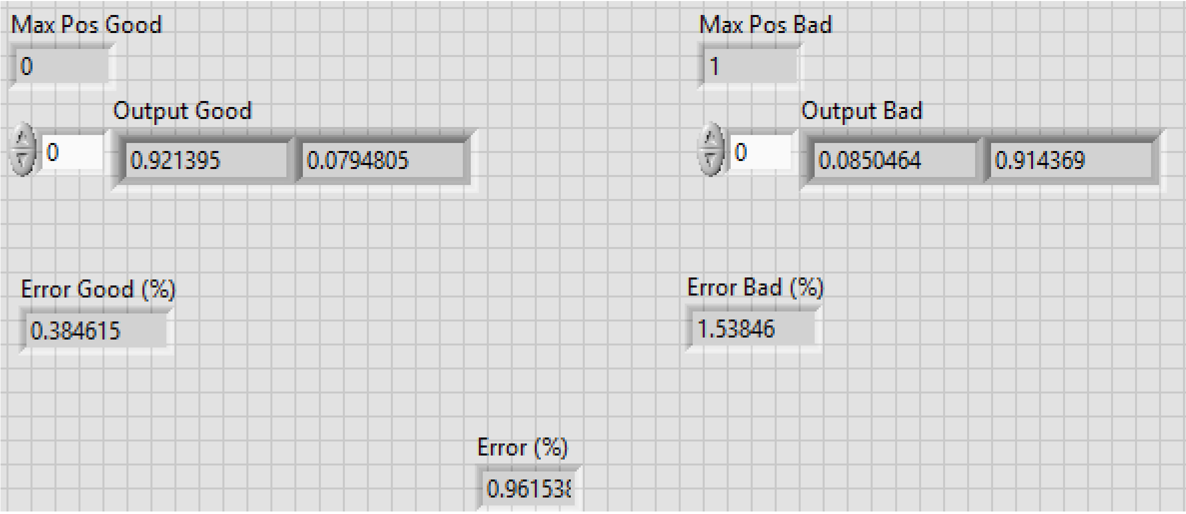


"ANN Learning" Interface

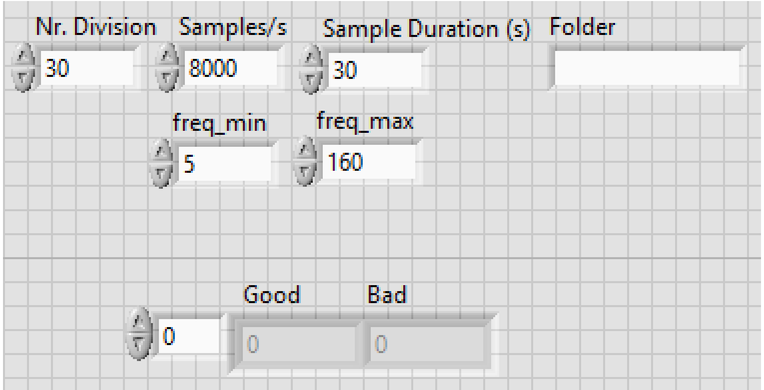


"ANN Learning" Block Diagram

The "Create ANN Accel" VI is the same as the "Create ANN LV" VI, found on Appendix B.



"Test ANN" Interface



"Apply ANN" Interface