ALLAN EDUARDO FEITOSA

**CLASSIFICATION TECHNIQUES FOR ADAPTIVE
DISTRIBUTED NETWORKS AND AERONAUTICAL
STRUCTURES**

São Paulo
2018

ALLAN EDUARDO FEITOSA

**CLASSIFICATION TECHNIQUES FOR ADAPTIVE
DISTRIBUTED NETWORKS AND AERONAUTICAL
STRUCTURES**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo
2018

ALLAN EDUARDO FEITOSA

**CLASSIFICATION TECHNIQUES FOR ADAPTIVE
DISTRIBUTED NETWORKS AND AERONAUTICAL
STRUCTURES**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Prof. Dr. Cássio Guimarães Lopes

Co-orientador:

Prof. Dr. Vítor Heloiz Nascimento

São Paulo
2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Feitosa, Allan Eduardo

Classification Techniques for Adaptive Distributed Networks and
Aeronautical Structures / A. E. Feitosa -- versão corr. -- São Paulo, 2018.
72 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Processamento de sinais 2.Monitoramento de estruturas 3.Classificação
de danos 4.Filtragem adaptativa 5.Detecção em redes distribuídas
I.Universidade de São Paulo. Escola Politécnica. Departamento de
Engenharia de Sistemas Eletrônicos II.t.

To the people I love the most.

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor, Professor Cássio Guimarães Lopes, for conducting the structuring of this thesis, for his comprehensive review of my work and for his support.

I would also like to thank Professor Vítor Heloiz Nascimento, for his indispensable technical advice during the development of this thesis, for his support and for all the things he taught me.

I would also like to thank the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and EMBRAER for funding this master thesis (grants 2016/06529-1 and GDT0109-15).

I would like to thank all my friends that were with me during this important stage of my life for their company, trust and love. I am deeply grateful specially to Carlos Mori for his support and helping me write this thesis, also to Marianne Frida Hermann for her company and love, and Matheus Conti for cheering me up when I needed it.

The last word goes to my mother, Sonia, who is not among us anymore and to whom I am completely grateful for being the person that I have become, and I am sure that she would be proud, as she always was.

ABSTRACT

This master thesis is the result of a collaborative work between EMBRAER and the Escola Politécnica da USP for the study of structural health monitoring (SHM) techniques using sensors applied to aircraft structures. The goal was to develop classification techniques to discriminate between different events arising in the aircraft structure during tests; in the short term, improving the current SHM system used by EMBRAER, based on acoustic emission and, in the long term, fostering the development of a fully distributed system.

As a result of studying classification methods for immediate use, we developed two techniques: the Spectral Similarity and a Support Vector Machines (SVM) classifier. Both are unsupervised solutions, due to the unlabeled nature of the data provided. The two solutions were delivered as a final product to EMBRAER for prompt use in the existing SHM system.

By studying distributed solutions for future implementations, we developed a detection algorithm based on adaptive techniques. The main result was a special initialization for a maximum likelihood (ML) detector that yields an exponential decay rate in the error probability to a nonzero steady state, using adaptive diffusion estimation in a distributed sensor network. The nodes that compose the network must decide, locally, between two concurrent hypotheses concerning the environment state where they are inserted, using local measurements and shared estimates coming from their neighbors. The exponential performance does not depend on the adaptation step size value, provided it is sufficiently small. The results concerning this distributed detector were published in the journal *IEEE Signal Processing Letters*.

Keywords: *Structural Health Monitoring, Acoustic Emission, EMBRAER, Sensors, Classification, Detection, Discrimination, Adaptive Networks, Distributed Networks, Maximum Likelihood.*

RESUMO

Esta dissertação de mestrado é o resultado de um trabalho colaborativo entre a EMBRAER e a Escola Politécnica da USP no estudo de técnicas de monitoramento do estado de saúde de estruturas (*Structural Health Monitoring* - SHM) utilizando sensores em estruturas aeronáuticas. O objetivo foi desenvolver técnicas de classificação para discriminar entre diferentes eventos que surgem em estruturas aeronáuticas durante testes; para o curto prazo, aperfeiçoando o atual sistema de SHM utilizado pela EMBRAER, baseado em emissão acústica e, no longo prazo, fomentando o desenvolvimento de um sistema completamente distribuído.

Como resultado do estudo de métodos de classificação para uso imediato, desenvolvemos duas técnicas: a Similaridade Espectral e um classificador que utiliza *Support Vector Machines* (SMV). Ambas as técnicas são soluções não-supervisionadas, devido a natureza não rotulada dos dados fornecidos. As duas soluções foram entregues como um produto final para a EMBRAER para pronta utilização em seu atual sistema de SHM.

Ao estudar soluções completamente distribuídas para futuras implementações, desenvolvemos um algoritmo de detecção baseado em técnicas adaptativas. O principal resultado foi uma inicialização especial para um detector de máxima verossimilhança (*maximum likelihood* - ML) que possui uma taxa de decaimento exponencial na probabilidade de erro até um valor não nulo em regime estacionário, utilizando estimação adaptativa em uma rede distribuída. Os nós que compõem a rede devem decidir, localmente, entre duas hipóteses concorrentes com relação ao estado do ambiente onde eles estão inseridos, utilizando medidas locais e estimativas compartilhadas vindas de nós vizinhos. O desempenho exponencial não depende do valor do passo de adaptação, se este for suficientemente pequeno. Os resultados referentes a este detector distribuído foram publicados na revista internacional *IEEE Signal Processing Letters*.

Palavras-Chave: *Monitoramento de estruturas, Emissão Acústica, EMBRAER, Sensores, Classificação, Detecção, Discriminação, Redes Adaptativas, Redes Distribuídas, Máxima Verossimilhança.*

CONTENTS

List of Figures	iii
List of Tables	vii
1 Introduction	8
1.1 SHM via Acoustic Emissions	9
1.2 Distributed Adaptive Detection	13
2 Solutions for EMBRAER's current SHM system	16
2.1 Classification using spectral similarity	17
2.2 Classification using features and SVM	23
3 Distributed Maximum Likelihood Detector	29
3.1 The Decision Problem over Distributed Networks	30
3.2 Distributed detection using diffusion LMS	31
3.3 ML Detector Algorithm Analysis	36
3.3.1 Initial estimate and its effect on performance	36
3.3.2 ML detector analysis: formulation	40
3.3.3 ML detector analysis: approximations for small step size	44
3.4 Simulations	46
3.4.1 Equalization of performance	48

3.4.2	Performance of the three different statistics	49
3.4.3	Effects of the LMS step size value	50
3.4.4	Changing w_2 and the SNR	52
3.4.5	Changing the network order	55
3.5	Tracking considerations	55
4	Conclusion	57
	Appendix A	59
	$L_k(i)$ expression derivation for the diffusion LMS.	59
	Appendix B	61
	A non-recursive expression for $\mathcal{R}_{\tilde{w}}(i)$	61
	Appendix C	63
	First order Taylor expansion for $\sigma_k(i)$	63
	Appendix D	65
	Network with 100 nodes	65
	Appendix E	67
	Definition of features used in the SHM system	67
	References	68

LIST OF FIGURES

1	(a) Mechanical strains in an aeronautical structure. (b) Collection of sensors along the structure. (c) Sensor Network.	10
2	Example of an AE event and sensors. (a) Location of AE source and three sensor which receive the AE signal (hit) in different times. (b) Possible waveforms of the hits received at sensors. The delay in arrival among hits allows the estimation of the most probable location of the source.	11
3	Model for a typical acoustic emission (AE) waveform and some of its main features. The threshold is used by each sensor to detect a hit and to define some of its features.	12
4	Representation of a hypothetical scenario with 3 events and a sensor. The captured AE hits were named “A-ch ₁ ”, “B-ch ₁ ” e “C-ch ₁ ”.	17
5	Block diagram illustrating the spectral similarity calculation of two waveforms w_i e w_j . The vector W_i with dimension N represents the N -point discrete Fourier transform (DFT) of the waveform w_i , and $ W_i $ is the vector composed of the absolute values of each element $c_{i,k} \in \mathbb{C}$ of W_i , $1 \leq k \leq N$	18
6	Matrix showing the similarity coefficients (in color discretization) of the hypothetical scenario of Figure 4.	19
7	Metal plate used in the fatigue test with sensors. In its central region is located the rivet lines.	20

8	(a) Map of the events, damaged rivets and sensors. (b) Effect of the filtering of events.	21
9	Map of largest event clusters by number of similar events.	21
10	Map of the six largest event clusters of similarity by number of events, each one with an energy histogram. The energy that is considered is provided by the first sensor to receive a hit in the subset that defines the event. The numbers above the maps indicates the number of events in each cluster.	22
11	(a) Map of events of the test chosen to implement the classifier, with the separation in the “rivet” class for the events close to the damage, and for the first N event related to time. (b) same as (a) with zoom around the damage position.	25
12	Results for the one-class SVM classifier. (a) $N = 100$, (b) $N = 300$, (c) $N = 500$, (d) $N = 700$ e (e) $N = 900$. For each map, the number of classified events is displayed (classified as similar to the “rivet” class), related to the total of events (7744) and to the events inside the “rivet” boundary (982).	26
13	A simple network topology example. The lines represent connections between nodes. $\mathcal{N}_k$ represents the neighborhood of node k . (Source: [19])	30
14	Simulation to illustrate the effect of $E\{t_k(i)\}$ on the error probabilities for node $k = 1$ and for the two initializations $\mathbf{w}_{k=1}(-1) = \mathbf{0}$ and $\mathbf{w}_{k=1}(-1) = \bar{\mathbf{w}} = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2}$. (a) $E\{t_1(i)\}$ for each initialization and for each hypothesis with step size $\mu = 10^{-2}$. (b) Same as (a) with $\mu = 0.5 \cdot 10^{-2}$. (c) Resulting error probabilities related to (a). (d) Resulting error probabilities related to (b).	38

15	Network topology (a) and statistics (b)	47
16	Logarithm of the error probability for the 10 nodes, with initialization at $\mathbf{w}_k(-1) = \mathbf{0}$, showing the equalization effect of diffusion LMS on these probabilities.	48
17	Logarithm of maximum network error probability over time, with initializations (a) $\mathbf{w}_k(-1) = \mathbf{0}$ and (b) $\mathbf{w}_k(-1) = \bar{\mathbf{w}} = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2}$ for the three statistics $t_{k,\text{suf}}(i)$, $t_{k,\text{approx}}(i)$ and $t_k(i)$. $Q_k(i)$ and $L_k(i)U_{0:i}$ (only) were computed as if there were no diffusion ($A = I$), as proposed in [23].	49
18	Maximum error probability in log scale over time, with initialization $\mathbf{w}_k(-1) = \mathbf{0}$ (top) and $\mathbf{w}_k(-1) = \bar{\mathbf{w}} = (\mathbf{w}_1 + \mathbf{w}_2)/2$ (bottom). The curves are arranged, from top to bottom (in steady state), for $\mu \in \{2 \cdot 10^{-2}, 1 \cdot 10^{-2}, 0.5 \cdot 10^{-2}\}$. The maximum value is taken among all nodes at every time i	51
19	Logarithm of theoretical error probability for node $k = 1$ for 17 different values of μ and calculated from (3.44) for same node.	52
20	Simulations showing the theoretical maximum error probabilities in the network to illustrate the influence of the proximity between $\mathbf{w}_1$ and $\mathbf{w}_2$, using initializations $\mathbf{w}_k(-1) = \mathbf{0}$ and $\mathbf{w}_k(-1) = \bar{\mathbf{w}} = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2}$, for $\mathbf{w}_1$ and $\mathbf{w}_2$ given by (a) equation (3.45), (b) equation (3.46) and (c) equation (3.47) and lower SNR ($\sigma_k^2(k)$ multiplied by 500), with step size $\mu = 10^{-2}$. (d) Same as (c), with step size reduced to $\mu = 10^{-3}$	53
21	Logarithm of theoretical error probability for node $k = 1$ for (a) different SNR and (b) different $\mathbf{w}_2$, changing its first coordinate. The curves were calculated using the approximation in equation (3.44) for small step sizes and for the transient ($i \leq i_0$).	54

- 22 Log of maximum error probability for a network composed of 100 nodes, for two values of the step size: $\mu = 2 \cdot 10^{-2}$ and $\mu = 1 \cdot 10^{-2}$. Two initializations were used: (a) $w_k(-1) = \mathbf{0}$ and (b) $w_k(-1) = \bar{w}$. 56
- 23 Matrix showing the connections among nodes. Rows and columns represent the nodes (numerated from 1 to 100), and a position colored in yellow indicates a connection between the corresponding nodes of the row and the column, while blue indicates no connection. 65

LIST OF TABLES

1	Table that summarizes the results in Figure (12).	27
2	ML Detector using Diffusion LMS algorithm and special initialization	39
3	Number of operations performed by the detection algorithm per node. $\bar{K}$ represents the average node degree.	39
4	Values of α_k for each node k	66
5	Values of $\sigma_{u,k}^2$ for each node k	66
6	Values of $\sigma_{v,k}^2$ for each node k	66

1 INTRODUCTION

This master thesis is part of a collaborative project between Escola Politécnica da Universidade de São Paulo and EMBRAER, which aimed at developing methods for classification and detection of events in aircraft structures.

EMBRAER employs a centralized structural health monitoring (SHM) [1] system to determine the location of potential damages in aircraft structures, using signals captured during fatigue tests by sensors spread throughout the structure under observation. These signals may carry some information about the structure health and can be obtained, for instance, via acoustic emission¹ (AE) [1], which is a well-known *non-destructive* SHM technique employed in many industries, such as oil and gas extraction and pipeline transport, ship building and maintenance, civil engineering and, more specifically, aeronautics, in which EMBRAER is pioneering this specific application. Part of this work focused on proposing techniques that can enhance the efficiency of EMBRAER's current system, using the available data provided by structural fatigue testing.

In addition to the early interest in developing and describing classifiers using its current SHM system, EMBRAER has also been showing interest in fully distributed sensor networks throughout airplane structures, using local computation only, envisioning a feasible implementation in the long term. Consequently, studying alterna-

¹Although the term seems to refer to a pure sound phenomenon, acoustic emission refers more broadly to all elastic waves in a material produced through the release of stress energy by some source, such as the arising of cracks.

tives for such an approach constitutes a significantly part of this master thesis. The distributed algorithm proposed here is still far from being applicable to the real SHM problem, but it provides an initial step in that direction, allowing a deeper understanding of distributed classification algorithms, in particular proving that the usual compromise between convergence speed and steady state performance in on-line (adaptive) estimation can be avoided at least in some classification problems. The results generated an article that was published in a high impact factor journal, namely the *IEEE Signal Processing Letters* [2].

In light of the above, it is possible to divide the work of this master thesis in two separated (but correlated) lines of research. One focused on developing classifiers using EMBRAER's current SHM system, aiming at a short-term implementation, whilst the other focused on the fully distributed solutions for the long term. We now turn to describe ahead each of these lines.

1.1 SHM via Acoustic Emissions

Nowadays there is a large interest in developing SHM systems for aircraft structures, aiming at increasing their reliability and lifetime, and also at reducing their maintenance costs. In order to assess the life cycle of aircraft components, tests are made in critical structural elements to detect the appearance of possible damages (cracks, for example) during load cycles². Inspections can be made visually, which may demand the disassembling of structure parts in order to access the inspected element, or using sensors attached to the structure. An SHM system encompasses the installation of multiple sensors along the structure under test, receiving spatio-temporal signals that contain relevant information about the mechanical stress to which the structure is submitted (Figure 1). There are several techniques for SHM, such as acoustic emis-

²A load cycle is a type of structural test in which a specimen is periodically submitted to mechanical strains, performed by actuators, in order to simulate the stress that the specimen would experience in a real scenario.

sion, Foucault currents, radiography, to name a few [1, 3]. These are *non-destructive* techniques, since they do not cause damage to the material during monitoring.

Figure 1: (a) Mechanical strains in an aeronautical structure. (b) Collection of sensors along the structure. (c) Sensor Network.

Currently, EMBRAER employs, among others, an SHM system based on acoustic emission (AE) in its aircraft structures. The AE technique is based on mechanical elastic waves that propagate in a structure when a fault arises or changes — for instance, when a crack emerges or increases. Some advantages of using AE are its large area coverage, the need for only passive piezoelectric sensors to “hear” the structure (therefore, requires no additional power consumption besides that used in possible preamplifiers), its rich damage-related information provided in real time, and its high sensitivity, since AE waves are also emitted in the early stages of a damage. The structural testing performed by EMBRAER is taken with the structure on the ground, and it is subjected to load cycles employing mechanical actuators that emulate the structural efforts that a plane experiences when airborne. A number of piezoelectric sensors is placed along the structure under inspection (Figure 1) and these sensors collect AE signals (called *hits*³) and send them to a processing center. This centralized system groups signals into *events*⁴, based on delay differences in the arrival time among the sensors, and estimates the most probable source location of the observed AE signals (Figure 2). In such a test, the sensors receive not only signals from the source, but

³A hit is defined when an AE signal is captured by a sensor, which occurs when the signal excursion exceeds a predetermined voltage threshold (see Figure 3).

⁴An AE event is defined by the SHM system used by EMBRAER when three or more AE signals (hits) are collected by different sensors in the structure during a definite period of time.

also ones coming from diverse sources, such as the mechanical actuators and rivets in the structure. The signals received are therefore considerably noisy. Our goal is to distinguish between different type of events, such as fatigue cracks and rivet breaks.

Figure 2: Example of an AE event and sensors. (a) Location of AE source and three sensor which receive the AE signal (hit) in different times. (b) Possible waveforms of the hits received at sensors. The delay in arrival among hits allows the estimation of the most probable location of the source.

Several studies have been published concerning the use of AE for detecting damages and for classification [1, 3–11]. The straightforward method to do so is to use the associated AE *features* extracted from the signal waveform, such as amplitude, rise-time, duration and count of cycles. Figure 3 shows the definitions of the main *features* related to a generic model of the AE waveform. In addition to the *features* illustrated in the figure, another important one is the *energy*, which in the AE literature is defined as the integral of the rectified voltage over the duration of the AE hit.

In Chapter 2, we describe the development of two original techniques of damage classification using the AE data provided by EMBRAER. We also describe how each technique can be used in the SHM system currently used by EMBRAER. The first technique is called *Spectral Similarity*, an original development which uses mostly AE waveform spectrum information. This technique consists of an unsupervised classification that groups AE events in clusters by the similarity of their absolute spectra⁵. The

⁵Unsupervised classification means that the available data used to train the classifier is not labeled in the classes in which one wants to discriminate [12, 13]. This can occur for different reasons; for instance, due to lack of prior knowledge about an attribute that can discriminate (separates) the data.

Figure 3: Model for a typical acoustic emission (AE) waveform and some of its main features. The threshold is used by each sensor to detect a hit and to define some of its features.

second one is also an unsupervised classification technique using *Support Vector Machines* (SVM) [12,13] that decides if collected AE events are similar to a predetermined training data set. Briefly speaking, in this case the similarity is decided by creating an M -dimensional space formed by M AE features (see Figure 3) chosen to train the classifier, and defining a boundary hypersurface for decision in the M -space [12, 13].

The development of the two aforementioned techniques was important to extend the knowledge about the data collected by EMBRAER using its current SHM system, and also allowed to initiate studies that have great potential for progress, for instance in developing supervised methods of classification. The two techniques represent, therefore, an initial step towards this direction. Additionally, the study performed during this line of research helped to increase the understanding about the physical nature of the data provided by EMBRAER, which has also interest in such knowledge. For instance, the Spectral Similarity Method highlighted the abundance of information contained in the waveform spectrum, which is a very important finding since EMBRAER did not store the waveforms in an ordinary SHM test routine. Therefore, this work has potential to induce improvements in EMBRAER's current system. We expect that the adoption of our two proposed methodologies described in Chapter 2 will lead to novel

findings and the development of even more robust techniques.

1.2 Distributed Adaptive Detection

EMBRAER's SHM system is centralized, i.e., the sensors along the structure send all their information to a central processor. Besides centralized methods, there is a large interest in developing smart distributed sensor networks for SHM; several works have been published concerning sensor networks with some calculation capacity and wireless communication, looking at reducing cabling requirements [14–17]. Sensor networks fully distributed along the aircraft structure are envisioned, placed during manufacturing and capable of continuously collecting information in real time about the structure state throughout its lifetime, particularly during flight; hence, less cabling, thus less weight, is fundamental. A class of techniques that emerges as a well-suited solution is that of adaptive networks [18].

Detection using adaptive networks has received considerable attention due to its energy saving capability, less communication resource requirements, robustness to node and link failure, scalability and tracking performance [19–35], employing different detection strategies depending on the application. For instance, several works focus on distributed *Neyman-Pearson* (NP) detectors [36], an approach appropriate, for example, to detect the presence of a target in a radar, since this detector maximizes the probability of detection of an event given a maximum desirable probability of false alarm. Examples of adequate applications of distributed NP detectors are cognitive radios [23] and netted radar systems [37]. Another available choice is the use of a Bayesian approach, when prior probabilities are set for the possible outcomes. When one can assume equal prior probabilities for the possible alternatives, the Bayesian detector reduces to the *maximum likelihood* (ML) detector [36], which is applied, for instance, in signal multiple-input multiple-output (MIMO) communication systems [38–42]. An ML detector is also used in problems in which the prior proba-

bilities for the hypotheses are uncertain — the ML detector, in this case, minimizes the worst-case error probability [43, thm. 1.2.1], because the maximum error probability in detection occurs when the prior probabilities are equally likely.

This thesis proposes an ML detector over a distributed network, which decides between two concurrent hypotheses about the environment state — in EMBRAER’s context, for instance, it could be imagined as a detector that decides if an AE source is due to one or another damage in the structure; or even decides between actual damage or noise. The decision algorithm is based on the estimation technique *diffusion* LMS [19,44], which was applied in [23–31] to design distributed NP detectors. These works use a *diffusion* learning process among the nodes [45,46], but there are several others relying on the *consensus* technique [32–35].

Several works have been published on distributed detection. Diffusion solutions such as the one proposed in [23] take advantage of well-known connections between estimation and detection [36, p. 140] to derive a distributed detection algorithm. A distributed adaptive detection algorithm robust in the presence of impulsive noise is developed in [25], whereas [26] describes an algorithm for a node-specific distributed detector. The works [27–31] study the steady state of diffusion detectors in the slow adaptation regime, whereas [32,33] consider a consensus-based distributed detection approach and show that its error probabilities decay asymptotically at an exponential rate with time.

In contrast with the existing works that focused on steady state performance of distributed detectors [27–33], we studied, for an ML detector, the transient performance. The proposed ML detector attains an exponential decay in the error probability during the transient, due to a specific algorithm initialization. Furthermore, the transient performance of this detector is almost indifferent to the value of the LMS step size, provided it is sufficiently small. The complete description of this distributed detector is covered in Chapter 3.

The results obtained in studying the convergence of the distributed and adaptive ML detector were published in the *IEEE Signal Processing Letters* [2], as aforementioned. It is also important to emphasize that, as this study represented a line of research that aimed at the long term, this ML detector model is not bounded to the requirement of a direct and immediate implementation in the SHM system currently employed by EMBRAER, but instead allowed us to investigate novel and efficient solutions that can stimulate new implementations and exploit new results.

2 SOLUTIONS FOR EMBRAER'S CURRENT SHM SYSTEM

Due to the unsupervised nature of the data provided by EMBRAER, the two techniques proposed ahead hold the typical attributes of unsupervised machine learning, which are:

- a) It is not possible, at first, to determine if the discrimination performed by such classifiers is the one sought, based on the classification results only;
- b) the nature of the separation¹ created by the classifier is, at first, unknown.

These limitations are overcome when it is possible to implement a supervised classifier, whose data are labeled with high degree of confidence in every statistical class. Given the adversity in labeling data in certain scenarios in real applications, which is the case for AE data acquisition, such a procedure can be quite complex or even unfeasible. Hence, unsupervised learning techniques gain considerable importance, as they can provide valuable clues about the nature of the events one wants to discriminate; such clues must be investigated externally to the classification. Also, they can suggest novel ways to label the data. Having these ideas in mind, we developed the following unsupervised techniques which aim at expanding the knowledge about AE events and improve the current system used by EMBRAER.

¹Separation means that the data can be divided into different classes through decision boundaries in a multidimensional space created by the classifier. The space dimension depends on the number of characteristics (*features*) used to produce the classifier. See [12, 13] for more about decision boundaries in classification problems.

2.1 Classification using spectral similarity

The spectral similarity technique represents a substantial result relative to the use of real data provided by EMBRAER. Besides, this technique was originally developed during this master degree project. It can be further used and studied to produce even more promising unsupervised discrimination methods.

To explain this technique, Figure 4 illustrates a hypothetical situation with a test specimen where three events, represented by the star symbols, are observed by “Sensor 1”. Suppose the events “A” and “C” have similar waveforms and happen very close to each other, while event “B” happens in a different position and has not a similar waveform related to “A” and “B”. Assume that the temporal order in which the events occur is “A, B, C”. The goal is to use the similarity technique to group the events with some degree of spectral likeness among them.

Figure 4: Representation of a hypothetical scenario with 3 events and a sensor. The captured AE hits were named “A-ch₁”, “B-ch₁” e “C-ch₁”.

The sensors used during EMBRAER's tests are piezoelectric [47], and due to variations in the frequency response among the sensors and to the various propagation paths of the AE signals in the material, the similarity technique is more appropriate for comparing hits received in the same sensor, as the example illustrated in Figure 4.

The similarity technique requires the waveforms of the EA hits, and consists of calculating the norm of the difference, two by two, of the absolute normalized spectra of these signals². Since we use spectral information, such technique requires no synchronization between the waveforms. Figure 5 illustrates the calculation procedure for two generic signals w_i e w_j , which represent the waveform of the AE hits; the calculation of the spectrum is represented by the "FFT" blocks in the figure, and only their absolute value is used. The power spectrum of each waveform is normalized before subtracting one from another, and finally, the negative value of the Euclidean norm of the difference is taken and named as *spectral similarity coefficient*. As a consequence of the definition, the coefficient value lies between $-\sqrt{2}$ and 0, representing, respectively, the minimum and maximum spectral similarity between two hits³.

Figure 5: Block diagram illustrating the spectral similarity calculation of two waveforms w_i e w_j . The vector W_i with dimension N represents the N -point discrete Fourier transform (DFT) of the waveform w_i , and $|W_i|$ is the vector composed of the absolute values of each element $c_{i,k} \in \mathbb{C}$ of W_i , $1 \leq k \leq N$.

In Figure 6 are illustrated the similarity coefficients of each pair of hits from the hypothetical example of Figure 4, in form of a symmetric matrix and using a value

²The spectra are normalized in order to compare them in such a fashion that eliminates possible differences in energy magnitude among the waveform spectra.

³The minimum similarity value, $-\sqrt{2}$, is obtained when $|w_i|$ and $|w_j|$ are Euclideanly orthogonal.

discretization by colors to display the similarity intensity. Yellow represents the highest values, occurring in the auto-similarity between hits (similarity of a hit to itself). The similarity between hits A- ch_1 e C- ch_1 is displayed in orange, denoting a relatively high similarity, which indicates the likeness of their waveforms (Figure 4). Thus, it is possible to observe that event B is only slightly similar to the others.

Figure 6: Matrix showing the similarity coefficients (in color discretization) of the hypothetical scenario of Figure 4.

Having calculated the spectral similarity coefficients, it is possible to group similar events into clusters. However, such an action is dependent on a threshold, defined between -2 e 0 , in order to decide for similarity between two hits. This threshold is chosen using an empirical procedure, in such a way that the classifier creates neither a large nor a small number of different clusters.

The similarity technique was applied in a considerable number of data from fatigue tests, provided by EMBRAER. Hereupon we present the obtained results concerning one of those tests. This one was conducted in a metal plate, which is part of an airplane structure, as depicted in Figure 7. The test consisted in submitting the plate to physical strain in order to observe the appearance of cracks along its surface. In its central region, lines of rivets are disposed, where the cracks were concentrated after the test. We can also verify in the same figure the positions of the sensors.

Figure 7: Metal plate used in the fatigue test with sensors. In its central region is located the rivet lines.

From the estimates of the AE sources and from the position of the sensors and the cracks — acquired via visual inspection at the end of the fatigue test — we created the map of events of Figure 8a. The blue dots in that map represent the position estimates of the sources. The cracks were found concentrated in the rivet region, and we cataloged the damaged rivets either as “broken” or “with crack” (in this case, the rivet is not yet broken). In Figure 8b we have the same map, however using a filter of events developed in a previous stage of this cooperative work between EMBRAER and USP to select the events estimated by the SHM system with higher confidence [48]. This filter removes the less probable true positive events based on the relative position of the sensor related to the estimated position of the source.

After grouping the events into clusters using the Spectrum Similarity technique, we ordered them by the number of events. In Figure 9 we show the six largest clusters following that order. In Figure 10, we provide, for the same clusters, a histogram showing the distribution of the feature “energy” for each corresponding cluster. This feature was chosen in order to investigate the presence of possible energy variations among the clusters. It is important to note that to define an event and its position estimate, as illustrated in Figure 2, only a subset of sensors is used (at least three),

Figure 8: (a) Map of the events, damaged rivets and sensors. (b) Effect of the filtering of events.

and each sensor has its own collection of features from their respective hits. Hence, we choose the energy provided by the sensor which received the hit first, in order to simplify the analysis, and because we assume that this sensor is the closest to the AE source.

Figure 9: Map of largest event clusters by number of similar events.

Comparing the histograms of energy distribution of each cluster in Figure 10, it is possible to identify significant variations among them. For instance, the first cluster has an energy distributed between the values 0 e 35, while the fifth cluster has an energy distribution until 300. Further differences can be found in each histogram by

Figure 10: Map of the six largest event clusters of similarity by number of events, each one with an energy histogram. The energy that is considered is provided by the first sensor to receive a hit in the subset that defines the event. The numbers above the maps indicates the number of events in each cluster.

inspecting its values.

The presented result shows that the grouping of events using the spectral similarity technique reveals some physical nature related to the energy contained in the events. We must recall that this method of grouping normalizes the energies of every hit, allowing us to compare the similarity between normalized spectra. Thus, the energetic information is not taken into account during the similarity calculation. On the other hand, Figure 10 suggests that the energy of the clusters can be also used to discriminate them. Therefore, this is an important conclusion, because it reveals that this technique can separates the clusters energetically, without considering the energy in the hits.

This conclusion suggests that there could exist a physical basis for the discrimination performed by the spectral similarity technique. To properly corroborate such a hypothesis, it is necessary to investigate the causes of this separation, looking for identifying:

- a) If each cluster is related to a different damage in the structure;
- b) if some clusters are related to the same damage, which would reveal different

ways that a damage can stimulate, via acoustic emission, the structure;

- c) if there is a temporal connection between the different clusters, i.e., if some clusters arise only in certain time intervals, or if some of them are composed by events spread all along the time of test.

From what we exposed, we proposed a methodological procedure to implement the spectral similarity technique in EMBRAER's current SHM system, in order to verify the hypothesis we suggested above. Such a procedure, which was forwarded directly to the EMBRAER's team, consists of:

1. Using a test specimen, if possible, with a limited number of known cracks and relatively distant to each other, in order to avoid superposition of events. It is also desirable that the known cracks could be determined with a relative high level of confidence;
2. Dividing the data acquisition (waveforms and AE features) into phases with defined time interval, in order to verify, in each phase, the grouping into clusters and the energy distribution among them, and to compare the results of each phase, in order to investigate the temporal evolution of clusters.
3. Inspecting, if possible, the test specimen at the end of each phase, and recording every possible modifications in the structure.

The spectral similarity technique, as already mentioned, is original and has a considerable chance of being improved during its practical use and though novel findings that can arise during its application in EMBRAER's current SHM system.

2.2 Classification using features and SVM

Alternatively to the spectral similarity technique, we studied the possibility of using the AE features directly in a unsupervised classification method. This can be justified, firstly, due to the fact that the features summarize several characteristics of the AE

events and, therefore, represent a quite efficient way of storing information. Secondly, such features have traditionally been used in AE tests for classification. Furthermore, machine learning solutions have already been used in a considerable number of SHM techniques [1]. In this master project, we implemented a classifier that uses Support Vector Machines (SVM) [12, 13], which was chosen due to its robustness, easy implementation and scalability. Briefly, the SVM can be understood as a decision boundary classifier.

In the test chosen to validate an SVM implementation, it was known the damage position (a broken rivet) in a metal plate under mechanical strains. With the features collected from the events defined by EMBRAER's SHM system, we pursued a discrimination between events similar to those occurred close to the broken rivet and those that may not be similar. In order to do that we chose an one-class SVM classifier [49], which consists of deciding if a test data set is similar (or included) to a previously established training data set.

In the test context, it is important to note that even an event occurring close to a rivet might be noise or another unknown source, and an event distant from the rivet position might be caused by the rivet itself, due to a localization error. In the following procedure we describe attempts to distinguish between those possibilities. To define a train data set, we chose a number N of events whose location estimates were close to the local where the broken rivet was positioned. For that, it was defined, empirically, a radius of 40mm around the rivet location, which we named "rivet boundary", and the first N events (in relation to time) whose locations were estimated inside the region defined by that boundary were labeled as belonging to the class "rivet"⁴. Figure 11 illustrates this labeling procedure performed in the position estimates for a given N .

To perform the classification training, we used the features *rise time*, *count*, *energy*,

⁴It is important to clarify that this class label is only a reference for the proximity of the events to the damage, since there is no real labels, and also that the classifier is unsupervised even though we had created an artificial label for the training data.

Figure 11: (a) Map of events of the test chosen to implement the classifier, with the separation in the “rivet” class for the events close to the damage, and for the first N event related to time. (b) same as (a) with zoom around the damage position.

duration, amp, rms, asl, pcnts, strength and *absenergy*⁵, collected by the sensor that first received a hit from the subset that defined the event, supposing that this sensor is the closest to the AE source (as we did before for the feature energy used to create the histograms in Figure 10). The SVM classifier was implemented with the function *fitsvm* of MATLAB, using a Gaussian boundary. Figure 12 shows the results for five different classifiers, changing the number of events N used to train the classifiers. The events that were not used for training were separated to compose the test data set. For example, Figure 12a shows the results for $N = 100$, in which the similar events are displayed in blue; 254 events of a total of 7744 were classified as similar, and 244 of these were inside the “rivet” boundary, which contained 982 events. The count of the results for every classifier is summarized in Table 1.

As can be observed in Figure 12, the larger the number N of events used to train the classifier, the larger the number of test events classified as similar⁶. Besides, they tend to be concentrated in specific regions (shown in blue in the maps). Since this SVM classifier is an unsupervised machine learning technique, additional examination

⁵See Appendix E for the definition of these features.

⁶This happens probably because a small number N of events is statistically insufficient to establish a similarity between the training data and the data tested.

Figure 12: Results for the one-class SVM classifier. (a) $N = 100$, (b) $N = 300$, (c) $N = 500$, (d) $N = 700$ e (e) $N = 900$. For each map, the number of classified events is displayed (classified as similar to the “rivet” class), related to the total of events (7744) and to the events inside the “rivet” boundary (982).

Table 1: Table that summarizes the results in Figure (12).

N	Total of rivet events	Inside boundary
100	254	244
300	822	763
500	1122	893
700	1196	930
900	1410	934

external to the classification is necessary, in order to verify the nature behind that similarity. Is is possible to raise some hypotheses about such a nature:

- a) These events could be related to other unknown damages located in the indicated regions (in blue), and could be similar to the known damage;
- b) They could be events of a distinct nature, although the classifier detected them as similar;
- c) These events could be nothing more than additional events related to the known damage, but their location estimates were systematically calculated erroneously by EMBRAER's SHM system.

A way to verify these hypotheses would be to perform an inspection of the plate after test, investigating the structural state of these regions where the similar events were concentrated.

Due to the difficulty or even perhaps the impossibility of extracting fully labeled data using the current system — i.e., knowing with high degree of confidence to which class belongs each data point — unsupervised techniques as the one we described above acquire great relevance, since they can provide important clues about the nature of the events we want to discriminate. Also, such techniques can suggest means of developing novel methods that lead to better labeling the data, which is fundamental since we lack labeled data. From these considerations and from the presented results, we propose to EMBRAER the following methodological procedure to employ

the SVM classifier proposed here:

1. From a test that is being taken, and in which the position of at least one damage is known, one chooses arbitrarily a boundary, and a certain number N of events located inside the boundary region are concentrated.
2. With these N events, one trains an SVM classifier using the features of the first sensor that detected the hit of each event (closest to the source location).
3. The rest of the data are used as a test set and are classified by the trained SVM.
4. The results are analyzed and one looks for regions with larger concentration of events classified as similar.
5. The test specimen is inspected in order to verify if the classification results have some physical interpretation.
6. Back to step (1), the classifier is updated with possible resets of the boundary and the number of events used to train the classifier.

The procedure described above would allow us to investigate if there is some physical interpretation concerning the similarities indicated by the classifier. If such connections exist, and supposing that they can be apprehended by inspection of the test specimen, novel characteristics of the AE events might become known, and therefore, it would allow us to improve the classifier techniques.

3 DISTRIBUTED MAXIMUM LIKELIHOOD DETECTOR

In this chapter, we describe the development of our distributed ML detector. Our formulation is general, i.e., SHM in aeronautical structures can be understood as one of several envisioned applications of distributed networks. We present a data model for distributed networks and formulate an ML detection problem, based on the adaptive diffusion LMS algorithm, and compute a local statistic at each node of the network that is mathematically tractable and asymptotically optimal in terms of error probability as time increases. In this formulation, the nodes that compose the network must decide between two concurrent hypotheses concerning the environment where they are inserted, using local measurements and shared estimates coming from their neighbors. For the developed statistic, we show that a special initialization of the proposed algorithm yields an exponential decay rate on error probabilities in transient. We show analytically and by simulations that the exponentially decaying error rate performance does not depend on the LMS step size value, provided it is sufficiently small.

Notation used throughout this chapter: We use lowercase boldface for vector quantities and normal font for scalars. Uppercase is used for matrices. Time-dependent quantities are denoted by indices i, j in parentheses. The operator $E\{\cdot\}$ denotes expectation, the operator T transposition. We use indices k, l in subscript to denote nodes. $A \otimes B$ denotes the Kronecker product of matrices A and B . I_D denotes the identity matrix with dimension D . $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, C)$ represents a Gaussian random vector $\mathbf{x}$ with mean $\bar{\mathbf{x}} = E\{\mathbf{x}\}$ and covariance matrix C .

3.1 The Decision Problem over Distributed Networks

In a generic distributed network with K nodes, as presented in Figure 13, each node k at every time instant i has access to a scalar observation $d_k(i)$ of its environment and to a *deterministic* regressor vector¹ $\mathbf{u}_k(i) \in \mathbb{R}^M$, and it must be capable of making a decision about the state of its environment based on these observations. Each node communicates only with its neighbors, and there is no fusion center to where the data in the nodes can be sent and processed. By allowing every node to receive additional information from its neighbors, the quality of its local decision process can be improved.

Figure 13: A simple network topology example. The lines represent connections between nodes. $\mathcal{N}_k$ represents the neighborhood of node k . (Source: [19])

The following data model

$$d_k(i) = \mathbf{u}_k^T(i) \mathbf{w}_s + v_k(i), \quad k = 1, \dots, K, \quad (3.1)$$

relates the observable variables $d_k(i)$ and $\mathbf{u}_k(i)$ to an unknown parameter vector $\mathbf{w}_s \in \mathbb{R}^M$, where $v_k(i) \sim \mathcal{N}(0, \sigma_{v,k}^2)$ is random noise. Nodes must decide between two hypotheses, H_1 and H_2 , about the unknown parameter:

$$\mathbf{w}_s = \begin{cases} \mathbf{w}_1 & \text{if } H_1, \\ \mathbf{w}_2 & \text{if } H_2. \end{cases} \quad (3.2)$$

¹The regressors $\mathbf{u}_k(i)$ can be interpreted as one realization of a random process, observed until time instant i , since at time i each node k has already observed $u_k(0), \dots, u_k(i)$. Thus, a detector developed to this problem seeks the best decision rule for these particular regressors.

Let $p(H_1)$ and $p(H_2)$ be the respective prior probabilities, i.e., the probability of occurrence of each hypothesis. We assume that the two possibilities are known *a priori*, although it is not known which of the two is in place. Hence, our formulation is different from [23–26], where $\mathbf{w}_2 = \mathbf{0}$ and an NP detector is derived.

We wish to minimize the error probability p_e in detection, defined as

$$p_e = p(\text{decides } H_1 | \text{occurs } H_2)p(H_2) + p(\text{decides } H_2 | \text{occurs } H_1)p(H_1), \quad (3.3)$$

and since in many problems the prior probabilities $p(H_1)$ and $p(H_2)$ are unknown or difficult to estimate, we consider instead the worst-case error probability

$$\min_{\text{decision boundary}} \max_{0 \leq p(H_1) \leq 1} p_e,$$

whose solution is the maximum likelihood (ML) detector, which considers $p(H_1) = p(H_2) = 1/2$. The theorem and its proof used for this conclusion can be consulted in [43, thm. 1.2.1], which we enunciate in the following:

Theorem (Minimax Theorem): *The ML detector minimizes the maximum possible average probability of error when the input distribution is unknown if the conditional error probability $p_{e,ML/H_i}$ is independent of i .*

3.2 Distributed detection using diffusion LMS

We look for a fully distributed ML detector using communication among connected nodes. In order to motivate a procedure to achieve this goal, we start by collecting all data of all nodes up to time i as follows:

$$\mathbf{d}_{0:i} = \text{col}\{d_1(i), \dots, d_K(i), \dots, d_1(0), \dots, d_K(0)\}, \quad ((i+1)K \times 1)$$

$$\mathbf{U}_{0:i} = \text{col}\{\mathbf{u}_1^T(i), \dots, \mathbf{u}_K^T(i), \dots, \mathbf{u}_1^T(0), \dots, \mathbf{u}_K^T(0)\}, \quad ((i+1)K \times M)$$

$$\mathbf{v}_{0:i} = \text{col}\{v_1(i), \dots, v_K(i), \dots, v_1(0), \dots, v_K(0)\}, \quad ((i+1)K \times 1)$$

where the col operator stacks its arguments column-wise. Therefore, the data model in (3.1) can be summarized as

$$\mathbf{d}_{0:i} = U_{0:i} \mathbf{w}_s + \mathbf{v}_{0:i}, \quad (3.4)$$

which describes a global linear model for the network data under the Gaussian uncertainty $\mathbf{v}_{0:i} \sim \mathcal{N}(\mathbf{0}, \Sigma_{0:i})$, where

$$\begin{aligned} \Sigma_{0:i} &= E\{\mathbf{v}_{0:i} \mathbf{v}_{0:i}^T\} = I_{i+1} \otimes \Sigma, & ((i+1)K \times (i+1)K) \\ \Sigma &= \text{diag}(\sigma_{v,1}^2, \dots, \sigma_{v,K}^2). & (K \times K) \end{aligned} \quad (3.5)$$

For the ML detection problem, the decision criterion for (3.4) that minimizes the error probability is given as [36, p. 78]

$$\frac{p(\mathbf{d}_{0:i}|H_2)}{p(\mathbf{d}_{0:i}|H_1)} \underset{H_1}{\overset{H_2}{\gtrless}} \frac{p(H_1)}{p(H_2)} = 1,$$

in which $p(\mathbf{d}_{0:i}|H_1)$ and $p(\mathbf{d}_{0:i}|H_2)$ are the probability density functions of the observed data conditioned upon each hypothesis. A global sufficient statistic for this ML detection problem is [36, p. 115]

$$t_{\text{global}}(i) = \mathbf{d}_{0:i}^T \Sigma_{0:i}^{-1} U_{0:i} (\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2} (\mathcal{E}_2 - \mathcal{E}_1) \underset{H_1}{\overset{H_2}{\gtrless}} 0, \quad (3.6)$$

with $\mathcal{E}_s = \mathbf{w}_s^T (U_{0:i}^T \Sigma_{0:i}^{-1} U_{0:i}) \mathbf{w}_s$, for $s = 1, 2$.

Following [23], we define local statistics $t_k(i)$ for each node for a distributed detection. In order to produce $t_k(i)$ that do not depend on global quantities, we use the connection between estimation and detection [36] to define local statistics written in terms of local estimates of the unknown parameter $\mathbf{w}_s$. This connection can be verified, for instance, producing a global statistic in terms of the *Minimum Variance Unbiased Estimate* (MVUE) for the unknown parameter $\mathbf{w}_s$, which is the unbiased estimate with the least variance possible. The existence of the MVUE for $\mathbf{w}_s$ is guaranteed

since (3.4) is a general linear data model, which is given as [50]

$$\mathbf{w}_{\text{MVUE}}(i) = (U_{0:i}^T \Sigma_{0:i}^{-1} U_{0:i})^{-1} U_{0:i}^T \Sigma_{0:i}^{-1} \mathbf{d}_{0:i}. \quad (3.7)$$

We can rewrite equation (3.6) in terms of the MVUE in (3.7), and using the fact that $\Sigma_{0:i}^{-1} = (\Sigma_{0:i}^{-1})^T$ by construction of matrix $\Sigma_{0:i}$ (see (3.5)), as

$$t_{\text{global}}(i) = \mathbf{w}_{\text{MVUE}}^T(i) U_{0:i}^T \Sigma_{0:i}^{-1} U_{0:i} (\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2} (\mathcal{E}_2 - \mathcal{E}_1) \underset{H_1}{\overset{H_2}{\geq}} 0. \quad (3.8)$$

The structure of (3.8) motivates, in our development, the definition of local statistics $t_k(i)$ based on local estimates for $\mathbf{w}_s$. Consider that each node k is able to obtain local linear estimates $\mathbf{w}_k(i)$ of the unknown parameter $\mathbf{w}_s$, at every time i , of the form

$$\mathbf{w}_k(i) = L_k(i) \mathbf{d}_{0:i}, \quad (3.9)$$

where $L_k(i) \in \mathbb{R}^{M \times (i+1)K}$ is a matrix that defines the local linear estimator². Multiplying each side of (3.4) by $L_k(i)$, we have

$$\mathbf{w}_k(i) = L_k(i) U_{0:i} \mathbf{w}_s + L_k(i) \mathbf{v}_{0:i} = \tilde{U}_{0:i} \mathbf{w}_s + \tilde{\mathbf{v}}_{0:i}, \quad (3.10)$$

which also describes a linear model under Gaussian uncertainty

$$\tilde{\mathbf{v}}_{0:i} = L_k(i) \mathbf{v}_{0:i} \sim \mathcal{N}(\mathbf{0}, L_k(i) \Sigma_{0:i} L_k(i)^T) = \mathcal{N}(\mathbf{0}, \tilde{\Sigma}_{0:i}).$$

Thus, we can define a local sufficient statistic for (3.10) in the same fashion as (3.6), which is given as

$$\begin{aligned} t_{k,\text{suf}}(i) &= \mathbf{w}_k^T(i) \tilde{\Sigma}_{0:i}^{-1} \tilde{U}_{0:i} (\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2} (\mathcal{E}_{k,2} - \mathcal{E}_{k,1}) \underset{H_1}{\overset{H_2}{\geq}} 0 \\ &= \mathbf{w}_k^T(i) Q_k(i) (\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2} (\mathcal{E}_{k,2} - \mathcal{E}_{k,1}) \underset{H_1}{\overset{H_2}{\geq}} 0, \end{aligned} \quad (3.11)$$

²For instance, if we define $L_k(i) = (U_{0:i}^T \Sigma_{0:i}^{-1} U_{0:i})^{-1} U_{0:i}^T \Sigma_{0:i}^{-1}$, the estimate $\mathbf{w}_k(i)$ in (3.9) becomes the MVUE. However, in that fashion $\mathbf{w}_k(i)$ would depend on global quantities, which is not what we pursue for the local estimates.

in which we defined the local quantities

$$Q_k(i) = \tilde{\Sigma}_{0:i}^{-1} \tilde{U}_{0:i} = \left(L_k(i) \Sigma_{0:i} L_k^T(i) \right)^{-1} L_k(i) U_{0:i},$$

$$\mathcal{E}_{k,s} = \mathbf{w}_s^T (\tilde{U}_{0:i}^T \tilde{\Sigma}_{0:i}^{-1} \tilde{U}_{0:i}) \mathbf{w}_s = \mathbf{w}_s^T (L_k(i) U_{0:i})^T Q_k(i) \mathbf{w}_s, \quad s = 1 \text{ or } 2. \quad (3.12)$$

In order to obtain the local estimate $\mathbf{w}_k$ in (3.9) in a distributed way, we follow [23–26] and use the diffusion LMS algorithm, an estimation technique given by the equations

$$\boldsymbol{\psi}_k(i) = \mathbf{w}_k(i-1) + \mu_k \mathbf{u}_k(i) [d_k(i) - \mathbf{u}_k^T(i) \mathbf{w}_k(i-1)], \quad (3.13)$$

$$\mathbf{w}_k(i) = \sum_{l=1}^K a_{k,l} \boldsymbol{\psi}_l(i), \quad \sum_{l=1}^K a_{k,l} = 1, \quad a_{k,l} \geq 0, \quad \forall k, l, \quad (3.14)$$

where $\boldsymbol{\psi}_k(i)$ are intermediate estimates, μ_k is a positive step size and the scalar $a_{k,l}$ represents the weighted contribution of $\boldsymbol{\psi}_l(i)$ from node l to estimate $\mathbf{w}_k(i)$ of node k . Using the diffusion LMS update, the resulting expression for $L_k(i)$, for each node k and time i , is given as

$$L_k(i) = \left[\sum_{l=1}^K a_{k,l} \mu_l \mathcal{W}_k(i) \quad \sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \right], \quad (3.15)$$

where $\mathcal{W}_k(i) = [\mathbf{0} \ \mathbf{0} \ \dots \ \mathbf{u}_k(i) \ \dots \ \mathbf{0}] \in \mathbb{R}^{M \times K}$. A complete derivation for (3.15) can be consulted in Appendix A.

The computation of the quantities $Q_k(i)$ is quite demanding (see (3.12)), since it would imply the exchanging of the matrices $L_k(i)$ in the diffusion LMS update, whose dimensions increase with every new measurement, as can be verified in (3.15). In order to simplify the computations and make them amenable for distributed implementations, [23] proves, for diffusion LMS, that $Q_k(i)$ in (3.11) can be approximated by the identity matrix, if the following assumptions are taken:

1. No diffusion update is used to compute $Q_k(i)$, that is, $a_{k,l} = 1$ for $l = k$ and $a_{k,l} = 0$ for $k \neq l$ in (3.14). Work [23] shows that this approximation yields good results

through simulations (see Figures 5, 6 and 7 in Section VI of [23]).

2. Sufficiently small step size. The proof is quite extensive, and can be verified in Section III-D of [23]. We only repeat here the main result of that proof (which depends on the assumption (1) above):

$$L_k(i)\Sigma_{0:i}L_k^T(i) \approx \beta_k L_k(i)U_{0:i}, \text{ for sufficient small } \mu_k,$$

in which β_k is a constant for the node k .

Considering these two assumptions, the local matrix $Q_k(i)$ reduces to:

$$Q_k(i) = (L_k(i)\Sigma_{0:i}L_k^T(i))^{-1}L_k(i)U_{0:i} = (\beta_k L_k(i)U_{0:i})^{-1}L_k(i)U_{0:i} = \frac{1}{\beta_k}I_M.$$

Since the performance of a detector is not affected by multiplication by a constant [36], an approximation for $Q_k(i)$ is simply

$$Q_k(i) = I_M.$$

Taking this approximation in (3.11), we obtain the following approximated optimal local statistic:

$$t_{k,\text{approx}}(i) = \mathbf{w}_k^T(i)(\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2}(\mathcal{E}_{k,\text{approx},2} - \mathcal{E}_{k,\text{approx},1}), \quad (3.16)$$

with $\mathcal{E}_{k,\text{approx},s} = \mathbf{w}_s^T(L_k(i)U_{0:i})^T\mathbf{w}_s$. The computation of $L_k(i)U_{0:i}$ is still quite demanding, because of its dependence on the matrix $L_k(i)$ and on every node having access to the regressor $u_k(i)$ at all the other nodes and for all time i . Therefore, we must also find a reasonable substitute for that quantity. Looking at (3.10) and observing that $L_k(i)$ represents the diffusion LMS update by choosing it as our estimation technique, after convergence of the algorithm it is expected that $L_k(i)U_{0:i} \approx I$. In fact, [23] argues that this is the case for the diffusion LMS in the limit $i \rightarrow \infty$ (see

Sections III-E and IV-B of [23]). Thus, in steady state, the statistic in (3.16) reduces to

$$t_{k,\text{approx}}(i \rightarrow \infty) = \mathbf{w}_k^T(i \rightarrow \infty)(\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2}(\mathbf{w}_2^T \mathbf{w}_2 - \mathbf{w}_1^T \mathbf{w}_1).$$

Hence, approximating $\mathcal{E}_{k,\text{approx},s}$ by $\mathbf{w}_s^T \mathbf{w}_s = \|\mathbf{w}_s\|_2^2$ in (3.16), we propose a local approximate statistic $t_k(i)$, for every k and i , given as

$$t_k(i) = \mathbf{w}_k^T(i)(\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2}(\|\mathbf{w}_2\|_2^2 - \|\mathbf{w}_1\|_2^2) \stackrel{H_2}{\underset{H_1}{\gtrless}} 0, \quad (3.17)$$

which is an asymptotically optimal local statistic ($t_k \approx t_{k,\text{suf}}$ as $i \rightarrow \infty$) assuming that the estimation algorithm converges³ and the two assumptions taken previously (p. 34). In Section 3.4, we illustrate, by simulations, how these approximations proposed in the local statistics influence the detection performance, comparing them to the optimal case when we use the optimal statistic in (3.11). As we stated above, the performance in steady state is the same using either the optimal statistic in (3.11) or the approximated statistic in (3.17).

3.3 ML Detector Algorithm Analysis

3.3.1 Initial estimate and its effect on performance

Since we do not know what is the real value of $\mathbf{w}_s$ when the local estimation algorithms start ($i = -1$), a natural choice for the initial estimate is $\mathbf{w}_k(-1) = \mathbf{0}$, as was done in [23–26]. From (3.17), this choice causes the statistics to become, for every node k ,

$$t_k(-1) = -\frac{1}{2}(\|\mathbf{w}_2\|_2^2 - \|\mathbf{w}_1\|_2^2).$$

However, we show in the sequel that this may not be the best initialization in terms of the rate of convergence of the error probability, because the performance of the detector is significantly affected by the initial value $t_k(-1)$.

³A full discussion about conditions for convergence of adaptive detectors using diffusion LMS and deterministic regressors is available in [23] in sections IV-B and IV-C, concerning the mean and the mean-square convergences respectively.

In (3.17), the detector decides whether $t_k(i)$ is in the upper half-plane (decide H_2) or in the lower half-plane (decide H_1), relatively to the threshold of decision at zero. Let us use γ to refer to this threshold. Theoretically, when the statistics $t_k(i)$ are initialized far from $\gamma = 0$, the ML detector only decides for one of the hypotheses in the first iterations, which means that it decides correctly 50% of the time, since the prior probabilities are equally likely. Therefore, it can take several iterations for the error probability to decrease, which will only occur when $t_k(i)$ is close enough to γ . On the other hand, if we start the statistic exactly on $\gamma = 0$, the filter updates will make $t_k(i)$ move in the correct direction on average right from the start, and the error probability will start reducing promptly. In order to accelerate this decay in this fashion, we can choose $\mathbf{w}_k(-1)$ so that $t_k(-1) = 0$. An initial estimate for that can be found by setting the left-hand side of (3.17) to zero, for instance,

$$\mathbf{w}_k(-1) = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2} = \bar{\mathbf{w}}. \quad (3.18)$$

This initialization is possible to implement since we suppose both $\mathbf{w}_1$ and $\mathbf{w}_2$ to be known.

In order to illustrate the intuitive idea exposed above, we performed preliminary simulations to show the effect of choosing $\mathbf{w}_k(-1) = \mathbf{0}$ or $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ on $E\{t_k(i)\}$ and on the error probability $p_{e,k}(i)$ using theoretical expressions to produce the results⁴. The results can be seen in Figures 14a to 14d, choosing node $k = 1$ of the simulated network for simplicity⁵. In Figure 14a we use an LMS step size value of $\mu = 10^{-2}$, and we can see that $E\{t_1(i)\}$ with initialization $\mathbf{w}_1(-1) = \mathbf{0}$ starts far from $\gamma = 0$, and only when $E\{t_1(i)\}$ for the two hypotheses gets close to the threshold, $p_{e,1}(i)$ in Figure 14c begins to decrease. On the other hand, choosing $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ causes a noticeably faster decrease in the initial iterations.

⁴The expressions used to perform the simulations are (3.19), (3.29), (3.34) and (3.35).

⁵The complete topological and statistical conditions used in this simulation is presented in Section 3.4.

Figure 14: Simulation to illustrate the effect of $E\{t_k(i)\}$ on the error probabilities for node $k = 1$ and for the two initializations $w_{k=1}(-1) = 0$ and $w_{k=1}(-1) = \bar{w} = \frac{w_1 + w_2}{2}$. (a) $E\{t_1(i)\}$ for each initialization and for each hypothesis with step size $\mu = 10^{-2}$. (b) Same as (a) with $\mu = 0.5 \cdot 10^{-2}$. (c) Resulting error probabilities related to (a). (d) Resulting error probabilities related to (b).

Figures 14b and 14d show the same for a step size value $\mu = 0.5 \cdot 10^{-2}$. It is noticeable that, even though the expected values converge more slowly in Figure 14b compared to Figure 14a (due to the smaller step size value), when the filters are initialized with $\bar{w}$ the convergence rate of the error probability in Figure 14d is the same as that in Figure 14c (and in addition reaches a lower steady state error probability). It is also important to notice that when the initialization $w_k(-1) = 0$ is used, the time $p_{e,1}(i)$ remains at 50% (-0.3010 in log scale) in Figure 14c is longer than in Figure 14d.

These features are simply a glance at the far-reaching consequences resulting of choosing $w_k(-1) = \bar{w}$. As will be shown later, with this choice the error probability converges exponentially fast to a value as small as desired during the transient.

The algorithm is summarized in Table 2, and Table 3 presents the number of operations performed by the algorithm per node⁶. Since the number of neighbors of each node in the network can vary, we use the average number of connection among nodes (average node degree) $\bar{K} = \frac{1}{K} \sum_{l=1}^K K_l$, where K_l is the number of nodes connected to node l , $1 \leq l \leq K$.

Table 2: ML Detector using Diffusion LMS algorithm and special initialization

For every node $k = 1, 2, \dots, K$ of the network, set $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$, and for every time instant $i \geq 0$, repeat:
Learning update: for every node k , obtain $\{d_k(i), \mathbf{u}_k(i)\}$ and repeat $\boldsymbol{\psi}_k(i) = \mathbf{w}_k(i-1) + \mu_k \mathbf{u}_k(i)[d_k(i) - \mathbf{u}_k^T(i) \mathbf{w}_k(i-1)],$
Diffusion update: for every node k , repeat $\mathbf{w}_k(i) = \sum_{l=1}^K a_{k,l} \boldsymbol{\psi}_l(i), \quad (a_{k,l} \text{ defined as in (3.14)})$
ML decision: for every node k , repeat $t_k(i) = \mathbf{w}_k^T(i)(\mathbf{w}_2 - \mathbf{w}_1) - \frac{1}{2}(\ \mathbf{w}_2\ _2^2 - \ \mathbf{w}_1\ _2^2),$ $t_k(i) \underset{H_1}{\overset{H_2}{\gtrless}} 0,$

Table 3: Number of operations performed by the detection algorithm per node. $\bar{K}$ represents the average node degree.

Algorithm step	$\times$	$+$
Learning update	$2M + 1$	$2M$
Diffusion update	$M\bar{K}$	$M\bar{K} - M$
ML decision	M	M
Totals	$3M + 1 + M\bar{K}$	$2M + M\bar{K}$

⁶In the computation, the subtraction of the quantity $\frac{1}{2}(\|\mathbf{w}_2\|_2^2 - \|\mathbf{w}_1\|_2^2)$ in the ML decision (see Table 2) is accounted as a single constant subtraction, since this quantity can be evaluated before the start of the algorithm (the values of $\mathbf{w}_1$ and $\mathbf{w}_2$ are known by hypothesis). Also, the comparison to zero is not accounted since it could be performed as a single signal bit check.

3.3.2 ML detector analysis: formulation

We now derive analytical expressions that describe the ML detector performance in terms of the error probability over time. For each node k , we can calculate the error probability at every time i using the following expression, which is derived from (3.19) and using the Gaussian distribution assumption on noise:

$$p_{e,k}(i) = \frac{1}{2} \mathcal{Q}\left(\frac{E\{t_k(i)|H_2\}}{\sigma_k(i)}\right) + \frac{1}{2} \mathcal{Q}\left(\frac{-E\{t_k(i)|H_1\}}{\sigma_k(i)}\right), \quad (3.19)$$

where $\sigma_k(i)$ is the standard deviation of $t_k(i)$ and $\mathcal{Q}(z_0) = \text{Prob}(Z > z_0)$ for a standard Gaussian variable Z :

$$\mathcal{Q}(z_0) = \frac{1}{\sqrt{2\pi}} \int_{z_0}^{\infty} \exp\left(-\frac{z^2}{2}\right) dz.$$

Expression (3.19) provides a way of determining the algorithm performance as a function of time. To completely understand how the initialization $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ influences this relation, we must develop expressions for $E\{t_k(i)|H_s\}$, $s = 1, 2$ and for $\sigma_k(i)$. For that we need an expression for $E\{\mathbf{w}_k(i)\}$ and for the covariance matrix of $\mathbf{w}_k$, which can be obtained from equations (3.13) and (3.14). Rewriting those equations in terms of the error quantities

$$\tilde{\mathbf{w}}_k(i) = \mathbf{w}_k(i) - \mathbf{w}_s, \quad \tilde{\boldsymbol{\psi}}_k(i) = \boldsymbol{\psi}_k(i) - \mathbf{w}_s, \quad s = 1 \text{ or } 2,$$

the LMS expressions (3.13) and (3.14) become

$$\tilde{\boldsymbol{\psi}}_k(i) = \tilde{\mathbf{w}}_k(i-1) + \mu_k \mathbf{u}_k(i) [v_k(i) - \mathbf{u}_k^T(i) \tilde{\mathbf{w}}_k(i-1)], \quad (3.20)$$

$$\tilde{\mathbf{w}}_k(i) = \sum_{l=1}^K a_{k,l} \tilde{\boldsymbol{\psi}}_l(i). \quad (3.21)$$

Let us define the matrix $A = (a_{k,l}) \in \mathbb{R}^{K \times K}$, such that $A \mathbb{1}_K = \mathbb{1}_K$ in which $\mathbb{1}_K$ denotes a vector with entries equal to one in $\mathbb{R}^K$ (see the definition of $a_{k,l}$ in (3.14)),

and also

$$\mathcal{A} = A \otimes I_M. \quad (KM \times KM) \quad (3.22)$$

Additionally, the following global expressions will be useful for our analysis:

$$\begin{aligned} \tilde{\mathbf{w}}(i) &= \text{col}\{\tilde{\mathbf{w}}_1(i), \tilde{\mathbf{w}}_2(i), \dots, \tilde{\mathbf{w}}_K(i)\}, & (KM \times 1) \\ \tilde{\boldsymbol{\psi}}(i) &= \text{col}\{\tilde{\boldsymbol{\psi}}_1(i), \tilde{\boldsymbol{\psi}}_2(i), \dots, \tilde{\boldsymbol{\psi}}_K(i)\}, & (KM \times 1) \\ \mathcal{D}(i) &= \text{diag}\{\mathbf{u}_1(i)\mathbf{u}_1^T(i), \dots, \mathbf{u}_K(i)\mathbf{u}_K^T(i)\}, & (KM \times KM) \\ \mathcal{U}(i) &= \text{diag}\{\mathbf{u}_1^T(i), \dots, \mathbf{u}_K^T(i)\}. & (K \times KM) \end{aligned} \quad (3.23)$$

Setting $\mu_k = \mu$ for simplicity, we can rewrite (3.20) and (3.21) as a single global expression

$$\tilde{\mathbf{w}}(i) = \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]\tilde{\mathbf{w}}(i-1) + \mu\mathcal{A}\mathcal{U}^T(i)\mathbf{v}(i), \quad (3.24)$$

with $\mathbf{v}(i) = \text{col}\{v_1(i), \dots, v_K(i)\} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Afterward, collect the local statistics $t_k(i)$ for all nodes into the global statistics vector

$$\mathbf{t}(i) = \text{col}\{t_1(i), t_2(i), \dots, t_K(i)\} \in \mathbb{R}^K,$$

and define the auxiliary quantities

$$W = I_K \otimes (\mathbf{w}_2 - \mathbf{w}_1), \quad (KM \times K) \quad (3.25)$$

$$\boldsymbol{\delta} = \mathbb{1}_K \otimes \frac{(\mathbf{w}_2 - \mathbf{w}_1)}{2}. \quad (KM \times 1) \quad (3.26)$$

With that, we can write the network statistics in (3.17) in the following compact expression:

$$\mathbf{t}(i) = \begin{cases} W^T(\tilde{\mathbf{w}}(i) - \boldsymbol{\delta}) & \text{if } H_1, \\ W^T(\tilde{\mathbf{w}}(i) + \boldsymbol{\delta}) & \text{if } H_2. \end{cases} \quad (3.27)$$

The expected value of $\mathbf{t}(i)$ can be calculated using the expected value of (3.24), which is (recall that the regressors $\{\mathbf{u}_k(i)\}$ are deterministic)

$$E\{\tilde{\mathbf{w}}(i)\} = \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]E\{\tilde{\mathbf{w}}(i-1)\}. \quad (3.28)$$

Calculating the recursion in (3.28) starting from $i = 0$, we obtain

$$E\{\mathbf{t}(i)\} = \begin{cases} W^T(\mathcal{F}(i, 0)E\{\tilde{\mathbf{w}}(-1)|H_1\} - \boldsymbol{\delta}) & \text{if } H_1, \\ W^T(\mathcal{F}(i, 0)E\{\tilde{\mathbf{w}}(-1)|H_2\} + \boldsymbol{\delta}) & \text{if } H_2, \end{cases} \quad (3.29)$$

in which we defined the following matrix⁷:

$$\mathcal{F}(i, j) = \begin{cases} \prod_{n=j}^i \mathcal{A}[I_{KM} - \mu\mathcal{D}(i+j-n)] & \text{if } i \geq j, \\ I_{KM} & \text{if } i < j. \end{cases} \quad (3.30)$$

Choosing $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$, we have $E\{\tilde{\mathbf{w}}(-1)|H_1\} = \boldsymbol{\delta}$ and $E\{\tilde{\mathbf{w}}(-1)|H_2\} = -\boldsymbol{\delta}$, resulting that (3.29) is now given as

$$E\{\mathbf{t}(i)\} = \begin{cases} W^T(\mathcal{F}(i, 0) - I_{KM})\boldsymbol{\delta} & \text{if } H_1, \\ -W^T(\mathcal{F}(i, 0) - I_{KM})\boldsymbol{\delta} & \text{if } H_2, \end{cases} \quad (3.31)$$

so that $E\{t_k(i)|H_2\} = -E\{t_k(i)|H_1\}$ for every node k and time i . Therefore, expression (3.19) can be simplified to

$$p_{e,k}(i) = \mathcal{Q}\left(\frac{E\{t_k(i)|H_2\}}{\sigma_k(i)}\right) = \mathcal{Q}(c_k(i)), \quad (3.32)$$

in which we defined $c_k(i) = E\{t_k(i)|H_2\}\sigma_k^{-1}(i)$.

We now need an expression for $\sigma_k(i)$. The covariance matrices of $\mathbf{t}(i)$ and $\tilde{\mathbf{w}}(i)$ are defined by:

$$\begin{aligned} \mathcal{R}_{\mathbf{t}}(i) &= E\{(\mathbf{t}(i) - \boldsymbol{\alpha})(\mathbf{t}(i) - \boldsymbol{\alpha})^T\}, \quad (K \times K) \\ \mathcal{R}_{\tilde{\mathbf{w}}}(i) &= E\{(\tilde{\mathbf{w}}(i) - \boldsymbol{\beta})(\tilde{\mathbf{w}}(i) - \boldsymbol{\beta})^T\}, \quad (KM \times KM) \end{aligned}$$

where $\boldsymbol{\alpha} = E\{\mathbf{t}(i)\}$ and $\boldsymbol{\beta} = E\{\tilde{\mathbf{w}}(i)\}$. In the following, we assume arbitrarily that

⁷The definition of $\mathcal{F}(i, j)$ as a two variable function is useful for derivations we present in Subsection 3.3.3.

H_2 occurs in order to derive an expression for $\mathcal{R}_t(i)$ in terms of $\mathcal{R}_{\tilde{w}}(i)$:

$$\begin{aligned}
\mathcal{R}_{t|H_2}(i) &= E\{(t(i)|H_2)(t(i)|H_2)^T\} - E\{t(i)|H_2\}E\{t(i)|H_2\}^T \\
&= E\{W^T\tilde{w}(i)\tilde{w}^T(i)W + W^T\tilde{w}(i)\delta^TW + W^T\delta\tilde{w}^T(i)W + W^T\delta\delta^TW\} \\
&\quad - E\{W^T\tilde{w}(i) + W^T\delta\}E\{\tilde{w}^T(i)W + \delta^TW\} \\
&= W^TE\{\tilde{w}(i)\tilde{w}^T(i)\}W - W^TE\{\tilde{w}(i)\}E\{\tilde{w}^T(i)\}W \\
&= W^T\mathcal{R}_{\tilde{w}}(i)W.
\end{aligned}$$

Conduction a similar derivation, it holds also that $\mathcal{R}_{t|H_1}(i) = W^T\mathcal{R}_{\tilde{w}}(i)W$. Therefore, we have

$$\mathcal{R}_t(i) = W^T\mathcal{R}_{\tilde{w}}(i)W. \quad (3.33)$$

The statistic variance $\sigma_k^2(i)$ is obtained, by definition, from the k th diagonal entry of the matrix $\mathcal{R}_t(i)$. Thus, it holds from (3.33), and recalling the definition of matrix W in (3.25), that

$$\sigma_k^2(i) = (\mathbf{w}_2 - \mathbf{w}_1)^T R_{\tilde{w}_k}(i) (\mathbf{w}_2 - \mathbf{w}_1), \quad (3.34)$$

in which $R_{\tilde{w}_k}(i)$ is the k th $M \times M$ diagonal block of matrix $\mathcal{R}_{\tilde{w}}(i)$ and also the covariance matrix of $\tilde{w}_k(i)$. We now obtain a non-recursive expression for $\mathcal{R}_{\tilde{w}}(i)$, in order to compute approximations to $c_k(i)$ in (3.32) in the next subsection. For every $i \geq 0$, from (3.24) and (3.28) and setting $\mathcal{R}_{\tilde{w}}(-1) = 0$, we have⁸

$$\mathcal{R}_{\tilde{w}}(i) = \mu^2 \sum_{j=0}^i \mathcal{F}(i, j+1) \mathcal{H}(j) \mathcal{F}^T(i, j+1), \quad (3.35)$$

in which we defined $\mathcal{H}(j) = \mathcal{A}\mathcal{U}^T(j)\Sigma\mathcal{U}(j)\mathcal{A}^T$ (recall the definitions of $\mathcal{A}$ and $\mathcal{U}(i)$ in (3.22) and (3.23), respectively, and that Σ is the covariance matrix of the noise $\mathbf{v}_k(i)$). As aforementioned, (3.35) is used to compute approximations of $c_k(i)$ in the next subsection, in order to describe how the initialization $\bar{w}$ affects the transient performance of the detection algorithm.

⁸A complete derivation of (3.35) can be consulted in Appendix B.

3.3.3 ML detector analysis: approximations for small step size

In this subsection, we show that $c_k(i)$ in (3.32) is asymptotically independent of the step size as $\mu \rightarrow 0$. In other words, the rate of convergence of the error probability $p_{e,k}(i)$ does not depend on the step size, provided it is small enough. This rate of convergence analysis holds for the transient. For a fixed nonzero μ , the error probability eventually reaches a steady state. It can be shown from (3.31)-(3.35) (see also [30]), that the steady state performance of the detector improves as μ is reduced, so our result means that the diffusion detector with initialization $\bar{w}$ can have an error probability as small as desired in steady state, with no degradation in the convergence rate.

For any given μ_0 for which diffusion LMS converges, let $i_0 < \infty$ be such that $p_{e,k}(i)$ did not reach the steady state for all $0 < \mu < \mu_0$, for all nodes k . We begin our analysis by finding approximations for the numerator and denominator of $c_k(i)$ in (3.32), valid for small μ , and for the initial convergence phase, $0 \leq i \leq i_0 < \infty$. Expanding $\mathcal{F}(i, j)$ in (3.30) as a polynomial on μ , we obtain

$$\mathcal{F}(i, j) = \mathcal{B}_0(i, j) + \mu \mathcal{B}_1(i, j) + \dots + \mu^{i-j+1} \mathcal{B}_{i-j+1}(i, j), \quad (3.36)$$

in which the $\mathcal{B}_n(i, j)$ are functions of $\mathcal{A}$ and $\mathcal{D}(n)$, for $j \leq n \leq i$. In particular, expanding (3.30), the first two coefficients of (3.36) are

$$\mathcal{B}_0(i, j) = \mathcal{A}^{i-j+1}, \quad \mathcal{B}_1(i, j) = - \sum_{n=j}^i \mathcal{A}^{i-n+1} \mathcal{D}(n) \mathcal{A}^{n-j}. \quad (3.37)$$

Since i_0 is finite, the $\mathcal{B}_n(i, j)$ are uniformly bounded for $0 \leq i \leq i_0$, and we can approximate (3.36) by a first-degree polynomial on μ . Substituting in (3.31), the resulting expression, conditioned on H_2 , is

$$\begin{aligned} E\{\mathbf{t}(i)|H_2\} &\approx -W^T(\mathcal{A}^{i-j+1} - \mu \sum_{n=j}^i \mathcal{A}^{i-n+1} \mathcal{D}(n) \mathcal{A}^{n-j} - I_{KM})\boldsymbol{\delta} \\ &= \mu W^T \sum_{n=0}^i \mathcal{A}^{i-n+1} \mathcal{D}(n) \mathcal{A}^{n-j} \boldsymbol{\delta} - W^T \mathcal{A}^{i-j+1} \boldsymbol{\delta} + W^T \boldsymbol{\delta}. \end{aligned} \quad (3.38)$$

Note that (recall the definitions of $\mathcal{A}$ in (3.22) and $\boldsymbol{\delta}$ in (3.26))

$$\begin{aligned}\mathcal{A}^i \boldsymbol{\delta} &= \mathcal{A}^{i-1} \mathcal{A} \boldsymbol{\delta} = \mathcal{A}^{i-1} \left(A \otimes I_M \right) \left(\mathbb{1}_K \otimes \frac{\mathbf{w}_2 - \mathbf{w}_1}{2} \right) \\ &= \mathcal{A}^{i-1} \left(\left(A \mathbb{1}_K \right) \otimes \left(I_M \frac{w_2 - w_1}{2} \right) \right),\end{aligned}$$

using the fact that $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$. Since $A \mathbb{1}_K = \mathbb{1}_K$ by construction of matrix A , we have

$$\mathcal{A}^i \boldsymbol{\delta} = \mathcal{A}^{i-1} \left(\mathbb{1}_K \otimes \frac{\mathbf{w}_2 - \mathbf{w}_1}{2} \right) = \mathcal{A}^{i-1} \boldsymbol{\delta},$$

and repeating this argument, it holds, therefore, that $\mathcal{A}^i \boldsymbol{\delta} = \boldsymbol{\delta}$, $\forall i \geq 0$. With this result and using it in (3.38), we can simplify that expression to

$$E\{\mathbf{t}(i)|H_2\} \approx \mu W^T \sum_{n=0}^i \mathcal{A}^{i-n+1} \mathcal{D}(n) \boldsymbol{\delta} = \mu \boldsymbol{\epsilon}(i), \quad (3.39)$$

in which we defined $\boldsymbol{\epsilon}(i) = W^T \sum_{n=0}^i \mathcal{A}^{i-n+1} \mathcal{D}(n) \boldsymbol{\delta} \in \mathbb{R}^K$ which does not depend on μ . Thus, we can find, for each node k , the expected value of $t_k(i)$ as follows:

$$E\{t_k(i)|H_2\} \approx \mu \epsilon_k(i), \quad 0 \leq i \leq i_0, \quad (3.40)$$

in which $\epsilon_k(i)$ represents the k th entry of vector $\boldsymbol{\epsilon}(i)$.

Now we find a corresponding first-degree approximation (on μ) for $\sigma_k(i)$. For that we use the fact that $\mathcal{R}_{\tilde{\mathbf{w}}}(i)$ in (3.35) can be expanded as a polynomial on μ , which can be verified substituting (3.36) in (3.35), resulting

$$\mathcal{R}_{\tilde{\mathbf{w}}}(i) = \sum_{n=1}^{i+1} \mu^{2n} \mathcal{P}_{2n}(i), \quad (3.41)$$

where the matrices $\mathcal{P}_{2n}(i) \in \mathbb{R}^{KM \times KM}$ correspond to the polynomial coefficients and do not depend on μ . From (3.34), (3.35) and (3.41), $\sigma_k^2(i)$ can also be calculated as

$$\sigma_k^2(i) = \sum_{n=1}^{i+1} \mu^{2n} (\mathbf{w}_2 - \mathbf{w}_1)^T P_{2n,k}(i) (\mathbf{w}_2 - \mathbf{w}_1), \quad (3.42)$$

in which we defined the matrix $P_{2n,k}(i)$ as the k th $M \times M$ diagonal block of matrix

$\mathcal{P}_{2n}(i)$.

Using a first order Taylor expansion for $\sigma_k(i) = \sqrt{\sigma_k^2(i)}$ around $\mu = 0$, we obtain the approximation for small step size⁹

$$\sigma_k(i) \approx \mu \sqrt{(\mathbf{w}_2 - \mathbf{w}_1)^T P_{2,k}(i) (\mathbf{w}_2 - \mathbf{w}_1)}, \quad (3.43)$$

and, since $P_{2,k}$ is the term related to μ^2 in (3.42), we can find by inspection that $\mathcal{P}_2(i) = \sum_{j=0}^i \mathcal{B}_0(i, j+1) \mathcal{H}(j) \mathcal{B}_0(i, j+1)^T$. Thus, $c_k(i)$ can be calculated, for sufficiently small step size, using (3.40) and (3.43), as follows:

$$c_k(i) \approx \frac{\epsilon_k(i)}{\sqrt{(\mathbf{w}_2 - \mathbf{w}_1)^T P_{2,k}(i) (\mathbf{w}_2 - \mathbf{w}_1)}}, \quad i \leq i_0. \quad (3.44)$$

Expression (3.44) does not depend on the step size μ . Therefore, the error probability in (3.32) is also approximately independent of μ during the transient ($i \leq i_0$).

3.4 Simulations

In this section we present the simulations used to verify the features of our proposed ML detector. We consider a network formed by $K = 10$ nodes, with topology depicted in Figure 15a. In order to obtain regressors $\mathbf{u}_k(i)$ with full covariance matrices, we generate 10 signals using first order Markov processes (which are used, among other applications, to model certain communication channels)¹⁰, with power $\sigma_{u,k}^2$ and correlation α_k , from which we form vectors of dimension $M = 10$. The expression for the first-order auto-regressive filter, at each node k , is given as

$$H_k(z) = \frac{\sqrt{\sigma_{u,k}^2 (1 - \alpha_k^2)}}{1 - \alpha_k z^{-1}}.$$

⁹A complete derivation of this result can be consulted in Appendix C.

¹⁰First-order auto-regressive models are usually employed to generate signals with known correlation properties in simulations of adaptive filtering algorithms [51, 52]. We used them in order to avoid overly simple uncorrelated regressors.

The noise present in the measurements has power $\sigma_{v,k}^2$. The weights $a_{k,l}$ in (3.14) are obtained using the Metropolis rule [53]

$$\begin{cases} a_{k,l} = \frac{1}{\max(n_k, n_l)}, & \text{if nodes } k \text{ and } l \text{ (} k \neq l \text{) are linked,} \\ a_{k,l} = 0, & \text{if nodes } k \text{ and } l \text{ (} k \neq l \text{) are not linked,} \\ a_{k,k} = 1 - \sum_{m=1, m \neq k}^K a_{k,m}, & \text{if } k = l, \end{cases}$$

in which n_k and n_l are the number of connections of nodes k and l . This composition results in A being a double stochastic matrix, such that $A = A^T$. Figure 15 summarizes the network topology and statistics.¹¹ We choose $\mathbf{w}_1$ and $\mathbf{w}_2$ close to each other in order to test an unfavorable detection situation:

$$\mathbf{w}_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T \quad \text{and} \quad \mathbf{w}_2 = [0.97 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T, \quad (3.45)$$

since in this situation the detection problem is more challenging: for larger differences, the performance of our algorithm would only improve. We provide simulations in which we show how the proximity between $\mathbf{w}_1$ and $\mathbf{w}_2$ alters the performance later in this section.

Figure 15: Network topology (a) and statistics (b)

¹¹ $\alpha_k \in \{0.145; 0.42; 0.92; 0.89; 0.76; 0.11; 0.38; 0.14; 0.85; 0.05\}$

$\sigma_{u,k}^2 \in \{0.9; 0.64; 0.6; 0.54; 0.9; 0.4; 0.96; 0.74; 0.82; 0.92\}$

$\sigma_{v,k}^2 \in \{0.09; 0.06; 0.008; 0.07; 0.08; 0.05; 0.07; 0.08; 0.04; 0.05\}$

3.4.1 Equalization of performance

We first show the phenomenon of equalization present in the error probabilities in Figure (16). As can be seen, the probabilities values at nodes are very similar, despite the diversity of network statistics. This effect is caused by the performance equalization of the diffusion LMS algorithm, and such an effect is well described in works [19,44]. In the following simulations, we depict the maximum node error probability $p_{e,\max}(i) = \max_k p_{e,k}(i)$, for every time instant i , and because of the equalization effect, it is not likely that the maximum value be substantially different from the error probabilities at each node.

Figure 16: Logarithm of the error probability for the 10 nodes, with initialization at $w_k(-1) = \mathbf{0}$, showing the equalization effect of diffusion LMS on these probabilities.

3.4.2 Performance of the three different statistics

First, we illustrate how the three statistics discussed in Section 3.2, namely $t_{k,\text{suf}}(i)$ (3.11), $t_{k,\text{approx}}(i)$ (3.16) and $t_k(i)$ (3.17), are different in performance, using a step size of $\mu = 10^{-2}$. Figures 17a and 17b show the simulation result for the maximum error probability $p_{e,\text{max}}(i)$ in the network.

Figure 17: Logarithm of maximum network error probability over time, with initializations (a) $\mathbf{w}_k(-1) = \mathbf{0}$ and (b) $\mathbf{w}_k(-1) = \bar{\mathbf{w}} = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2}$ for the three statistics $t_{k,\text{suf}}(i)$, $t_{k,\text{approx}}(i)$ and $t_k(i)$. $Q_k(i)$ and $L_k(i)U_{0:i}$ (only) were computed as if there were no diffusion ($A = I$), as proposed in [23].

Recalling from Subsection 3.2 that in order to compute $Q_k(i)$ and $L_k(i)U_{0:i}$ (see (3.11)), we assume, and only for these computations, no diffusion update ($A = I$). In Figure 17a, the initialization is $\mathbf{w}_k(-1) = \mathbf{0}$ for all statistics, and the detector with $t_k(i)$ only achieves the performance of the detectors using $t_{k,\text{suf}}(i)$ and $t_{k,\text{approx}}(i)$ in steady state, as expected from our analysis in Section 3.2. In Figure 17b, on the other hand, we used the initialization $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ for all statistics, and the result is that our detector with this initialization avoids the initial “dead time” in which the error probability does not decrease. This happens only for $t_k(i)$ because the initialization $\bar{\mathbf{w}}$ was proposed to accelerate the performance of the detector using particularly this statistic (see Section 3.3.1). The three detectors converge to the same steady state condition, as expected from our analysis.

3.4.3 Effects of the LMS step size value

The plots in Figure 18 show how the choice of initial estimate changes the detector performance in terms of maximum error probability. The step size has a large influence when the initialization is at $\mathbf{w}_k(-1) = \mathbf{0}$. However, when initialization is $\bar{\mathbf{w}}$, the plots in Figure 18 show that smaller values of μ achieve a slightly faster decrease in the maximum error probability in the first iterations.

The aforementioned non-intuitive behavior may be explored via the theoretical curves obtained from (3.19) using (3.31) and (3.34). Figure 19 shows the logarithm of theoretical error probabilities over time for 17 different values of the step size, from 10^{-2} to $10^{-2}/2^8$, top to bottom, for the same network setting as before (Figure 15). As the curves show, smaller values of the step size attain lower error probabilities for a given time i ; in other words, they achieve better performance (the plots are not smooth because the regressor was obtained from a single realization of an AR(1) model). The value of the step size defines the steady state of $p_e(i)$, which is smaller as the step size decreases, as predicted in [30].

Figure 18: Maximum error probability in log scale over time, with initialization $w_k(-1) = \mathbf{0}$ (top) and $w_k(-1) = \bar{w} = (w_1 + w_2)/2$ (bottom). The curves are arranged, from top to bottom (in steady state), for $\mu \in \{2 \cdot 10^{-2}, 1 \cdot 10^{-2}, 0.5 \cdot 10^{-2}\}$. The maximum value is taken among all nodes at every time i .

Another feature that can be noticed is that the curves in Figure 19 start by following the same exponential decrease rate for all values of μ . As time increases, the curves detach from one another and converge to constant values; however, for smaller step sizes, the error probability keeps decreasing for a longer time. In fact, this exponential asymptotic rate (as $\mu \rightarrow 0$) represents the curve that is obtained when $p_e(i)$ is calculated with expression (3.32) using the result in (3.44), marked with Δ in Figure 19. That curve represents an exponential decay rate in terms of error probability, and our analysis shows that smaller values of step sizes better approximate this decay rate. Therefore, for known regressors one can compute *offline* the number of iterations to reach a desired maximum error probability, based solely in expressions (3.32) and (3.44).

Figure 19: Logarithm of theoretical error probability for node $k = 1$ for 17 different values of μ and calculated from (3.44) for same node.

3.4.4 Changing w_2 and the SNR

We show that the performance results are qualitatively the same for any pair of values for w_1 and w_2 . The difference is in the magnitude of the steady state error probability, and on the rate of convergence, which both change with the SNR. To illustrate that, we provide the following simulations in Figures 20a to 20d. Figure 20a shows the maximum error probabilities $p_{e,\max}(i)$, simulated with same network settings (Figure 15), using initializations $w_k(-1) = \bar{w}$ and $w_k(-1) = 0$. Due to a very low $p_{e,\max}(i)$ in steady state that results in some of the plots, which would require a very large number of realizations to estimate with confidence, these plots were obtained using theoretical expressions (equations (3.32) and (3.44)). The values of w_1 and w_2 for Figure 20a are given by (3.45).

In Figure 20b, we maintained the same w_1 and changed w_2 to

$$w_2 = [0.9 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T, \quad (3.46)$$

and in Figure 20c, we change it to the randomly chosen vector

$$w_2 = [-1.350 \ 3.035 \ 0.725 \ -0.063 \ 0.715 \ -0.205 \ -0.124 \ 1.490 \ 1.409 \ 1.417]. \quad (3.47)$$

Figure 20: Simulations showing the theoretical maximum error probabilities in the network to illustrate the influence of the proximity between w_1 and w_2 , using initializations $w_k(-1) = \mathbf{0}$ and $w_k(-1) = \bar{w} = \frac{w_1 + w_2}{2}$, for w_1 and w_2 given by (a) equation (3.45), (b) equation (3.46) and (c) equation (3.47) and lower SNR ($\sigma_k^2(k)$ multiplied by 500), with step size $\mu = 10^{-2}$. (d) Same as (c), with step size reduced to $\mu = 10^{-3}$.

Therefore, we tested the effect of choosing very similar solutions against when they are diverse. We use a step size of $\mu = 10^{-2}$ for both Figures 20a and 20b. We see that $p_{e,\max}(i)$ starts decaying quickly for $w_k(-1) = \bar{w}$ already in the first iterations, while for $w_k(-1) = \mathbf{0}$ $p_{e,\max}(i)$ initially remains at its maximum value (50%), and decays a number of iterations later. Hence, the effect of choosing $w_k(-1) = \bar{w}$ over $w_k(-1) = \mathbf{0}$ is observed in both scenarios. Also, the effect of changing w_2 affects the steady state error probability, as we can see in these figures.

Figure 20c shows a simulation with w_2 given by (3.47), a step size $\mu = 10^{-2}$, but the noise variances in all nodes were multiplied by 500, otherwise the steady state error probability would be too small and poor to compare. In Figure 20d, the step size of

Figure 20c was changed to $\mu = 10^{-3}$. In both figures, we still observe the same general behavior. Also, the figures show that reducing the step size lowers the error probability in steady state. If the initialization $\mathbf{w}_k(-1) = \mathbf{0}$ is used, reducing μ increases the time that $p_{e,\max(i)}$ stays at 50% (blue curves).

Our detection method is useful in any situation, but particularly so when the SNR (measured by the norm of $\mathbf{w}_1 - \mathbf{w}_2$ divided by the noise standard deviation) is small, that is, for the most challenging detection problems. This is because in this situation, our algorithm allows the use of a small step size (and thus a low error probability) without sacrificing convergence speed (Figures 20a and 20b). For large SNR, we avoid the initial stalling in convergence, but particularly in this case the use of a small step size is not necessary, since the steady state error probability would be small even for large step sizes (Figures 20c and 20d).

We also provide in Figure 21 simulation results in which we varied the SNR and the proximity between $\mathbf{w}_1$ and $\mathbf{w}_2$. The curves represent the log theoretical error probabilities for node $k = 1$ calculated using the $c_k(i)$ approximation for $\mu \rightarrow 0$ in equation (3.44). In Figure 21a, we simulated the network with the same setting that we have been using, varying only the variance of noise in nodes σ_k^2 .

Figure 21: Logarithm of theoretical error probability for node $k = 1$ for (a) different SNR and (b) different $\mathbf{w}_2$, changing its first coordinate. The curves were calculated using the approximation in equation (3.44) for small step sizes and for the transient ($i \leq i_0$).

When the SNR is higher ($0.5 \times \sigma_k^2$), the exponential decay rate is increased. The opposite occurs when the SNR is lower ($2 \times \sigma_k^2$). Figure 20 shows how the decay rate is altered by changing the first coordinate of $\mathbf{w}_2$, recalling that the w_1 and w_2 are given by (3.45).

We must recall that the actual experimental curves of error probability will detach from these curves in Figures 21a and 20 as i increases, because the approximation for small step sizes is valid only for the transient. This is what we showed in Figure 19, where the smaller the step size value, the better the approximation holds.

3.4.5 Changing the network order

We simulated our detection algorithm so far using a small network in order to allow an easy visualization and to facilitate its statistical and topological description. However, the main results presented concerning the developed ML detector are qualitatively the same regardless of the network size. To show that, we simulated a new network with 100 nodes, comparing the two initializations $\mathbf{w}_k(-1) = \mathbf{0}$ and $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ for two values of the step size, namely $\mu = 2 \cdot 10^{-2}$ and $\mu = 1 \cdot 10^{-2}$. The values of w_1 and w_2 are given by (3.45). The topology and the statistical properties are displayed in Appendix D. Figures 22a and 22b show the simulation results, where the early decay in the error probability for $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$ can be also seen.

3.5 Tracking considerations

One of the most important benefits of using distributed adaptive networks is their tracking capability. However, the step size must be large enough to allow fast tracking of changes in the environment¹². The proposed algorithm has a performance that

¹²Strictly speaking, there is an optimal step size value to track these changes. If the step size is too large, there is a large variance, and if it is too small, the bias becomes too large [51]. However, this optimal choice will probably be larger than the step size value used in our detection algorithm, which should be as small as possible to improve the performance.

Figure 22: Log of maximum error probability for a network composed of 100 nodes, for two values of the step size: $\mu = 2 \cdot 10^{-2}$ and $\mu = 1 \cdot 10^{-2}$. Two initializations were used: (a) $\mathbf{w}_k(-1) = \mathbf{0}$ and (b) $\mathbf{w}_k(-1) = \bar{\mathbf{w}}$.

improves as the step size is reduced, as shown by several simulation we presented. However, if an abrupt change occurs in the parameter $\mathbf{w}_s$, the algorithm will take a long time to recover. However, it is easy to determine if an abrupt change occurs (for example, if the true hypothesis changes from H_1 to H_2). Since the algorithm is using a small step size, if $\mathbf{w}_s$ changes abruptly, the node estimation error in diffusion LMS, defined as $e_k(i) = \mathbf{d}_k(i) - \mathbf{u}_{k,i}^T \mathbf{w}_{k,i-1}$ (see (3.13)), will have a sharp increase. Supposing it is detected, the nodes can switch their current estimates back to $\bar{\mathbf{w}}$, recovering the advantages of our method. Other simple solutions would be to reset the estimates at $\bar{\mathbf{w}}$ periodically at the nodes or to verify if there are changes in the node statistics $t_k(i)$.

Yet another option, albeit with higher complexity, would be to use combinations of filters, some with larger step sizes, others with lower step sizes. A final solution would be to use the universal distributed algorithm of [54], and take advantage of the multiple nodes in the network, using some nodes with small step size (to guarantee small probability of error in steady state) and other nodes with larger step size (to guarantee good tracking behavior).

4 CONCLUSION

During this master degree project, we developed solutions for the short term using the available data provided by the SHM system currently used by EMBRAER, pursuing its improvement and a better understanding of the data and its nature. Also, we developed long term solutions, envisioning novel and efficient distributed detection implementations, aiming at its inherent advantages such as less weight (considering wireless implementations), network scalability, less communication resources requirements and robustness to node and link failure, which recommend them to be used in different SHM systems, including those applied to aircraft structures.

The research on the current centralized SHM method used by EMBRAER resulted in the development of the two unsupervised classification techniques described in Chapter 2, that is, the Spectral Similarity and the SVM classifier. The first technique consists in grouping events by the similarity of their normalized spectral information, and requires the waveforms of the signals detected at the sensors, while the second consists of using AE features to decide if events are similar to a predetermined training data set. Both techniques are unsupervised classifiers, and looking for turning them in even more useful tools with potential for advancing and developing future studies, we proposed two methodological procedures describing how to apply them in controlled tests, aiming at studying and investigating AE events from the point of view of their physical interpretation. Both procedures focused on attaining a better understanding about acoustic emission, aiming at (i) developing feasible methods for labeling the acquired data, (ii) implementing supervised or semi-supervised classifiers, and (iii) ap-

plying the respective techniques in real structural tests in order to improve them with new findings. These techniques were delivered to EMBRAER at the end of the corresponding stage of this cooperative work, in form of MATLAB codes, to be used in EMBRAER's SHM system. EMBRAER acknowledged that the work planned for this stage was fully achieved.

Concerning our long term research line, we developed a distributed and adaptive ML detector that (i) accelerates the convergence of diffusion detection algorithms; (ii) allows exponential convergence of the error probability in stationary environments; (iii) allows the designer to predict, if enough *a priori* information is available, after how many iterations a given performance will be attained. These results were published in the *IEEE Signal Processing Letters*. Future steps of this work on distributed detectors include (i) the study of similar solutions when the prior probabilities for each hypotheses are different—which are known as *Maximum A Posteriori Probability* (MAP) detectors [36, p. 80]; (ii) an extension to more than two hypotheses; (iii) an extension to stochastic regressors; (iv) the study of NP detectors investigating the possibility of similar solutions and performance.

APPENDIX A

$L_k(i)$ expression derivation for the diffusion LMS.

Equations (3.13) and (3.14) can be rewritten in a single expression as follows:

$$\begin{aligned}
 \mathbf{w}_k(i) &= \sum_{l=1}^K a_{k,l} \boldsymbol{\psi}_l(i) \\
 &= \sum_{l=1}^K a_{k,l} \left[\mathbf{w}_l(i-1) + \mu_l \mathbf{u}_l(i) [d_l(i) - \mathbf{u}_k^T(i) \mathbf{w}_l(i-1)] \right] \\
 &= \sum_{l=1}^K a_{k,l} \left[\mu_l \mathbf{u}_l(i) d_l(i) + [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] \mathbf{w}_l(i-1) \right] \\
 &= \sum_{l=1}^K a_{k,l} \mu_l \mathbf{u}_l(i) d_l(i) + \sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] \mathbf{w}_l(i-1). \quad (\text{A.1})
 \end{aligned}$$

We can rewrite the first term in (A.1) as

$$\sum_{l=1}^K a_{k,l} \mu_l \mathbf{u}_l(i) d_l(i) = \sum_{l=1}^K a_{k,l} \mu_l \mathcal{W}_k(i) \mathbf{d}(i), \quad (\text{A.2})$$

in which we define

$$\begin{aligned}
 \mathcal{W}_k(i) &= [\mathbf{0} \ \mathbf{0} \ \dots \ \mathbf{u}_k(i) \ \dots \ \mathbf{0}], \quad (M \times K) \\
 \mathbf{d}(i) &= \text{col}\{d_1(i), \dots, d_K(i)\}.
 \end{aligned}$$

The second term in (A.1) can be rewritten as

$$\sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \mathbf{d}_{0:i-1}, \quad (\text{A.3})$$

since, by the definition in (3.9), $\mathbf{w}_k(i-1) = L_l(i-1) \mathbf{d}_{0:i-1}$. We can re Using (A.2)

and (A.3), we can rewrite (A.1) as

$$\begin{aligned}
\mathbf{w}_k(i) &= \sum_{l=1}^K a_{k,l} \mu_l \mathcal{W}_k(i) \mathbf{d}(i) + \sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \mathbf{d}_{0:i-1} \\
&= \sum_{l=1}^K a_{k,l} \left[\mu_l \mathcal{W}_k(i) \mathbf{d}(i) + [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \mathbf{d}_{0:i-1} \right] \\
&= \sum_{l=1}^K a_{k,l} \begin{bmatrix} \mu_l \mathcal{W}_k(i) & [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \end{bmatrix} \begin{bmatrix} \mathbf{d}(i) \\ \mathbf{d}_{0:i-1} \end{bmatrix} \\
&= \left[\sum_{l=1}^K a_{k,l} \mu_l \mathcal{W}_k(i) \quad \sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \right] \mathbf{d}_{i:0}, \quad (\text{A.4})
\end{aligned}$$

since $\mathbf{d}_{0:i} = \text{col}\{\mathbf{d}(i), \mathbf{d}_{i:i-1}\}$. Therefore, comparing (A.4) with (3.9), we must have

$$L_k(i) = \left[\sum_{l=1}^K a_{k,l} \mu_l \mathcal{W}_k(i) \quad \sum_{l=1}^K a_{k,l} [I_M - \mu_l \mathbf{u}_l(i) \mathbf{u}_l^T(i)] L_l(i-1) \right], \quad (\text{A.5})$$

which is the expression given by (3.15).

APPENDIX B

A non-recursive expression for $\mathcal{R}_{\tilde{\mathbf{w}}}(i)$

The covariance matrix of $\tilde{\mathbf{w}}(i)$ is defined as

$$\mathcal{R}_{\tilde{\mathbf{w}}}(i) = E\{\tilde{\mathbf{w}}(i)\tilde{\mathbf{w}}^T(i)\} - E\{\tilde{\mathbf{w}}(i)\}E\{\tilde{\mathbf{w}}(i)\}^T. \quad (\text{B.1})$$

To fully develop this expression, we need expressions (3.23) and (3.28):

$$\tilde{\mathbf{w}}(i) = \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]\tilde{\mathbf{w}}(i-1) + \mu\mathcal{A}\mathcal{U}^T(i)\mathbf{v}(i),$$

$$E\{\tilde{\mathbf{w}}(i)\} = \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]E\{\tilde{\mathbf{w}}(i-1)\}.$$

From (3.23), we obtain

$$\begin{aligned} E\{\tilde{\mathbf{w}}(i)\tilde{\mathbf{w}}^T(i)\} &= \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]E\{\tilde{\mathbf{w}}(i-1)\tilde{\mathbf{w}}^T(i-1)\}[I_{KM} - \mu\mathcal{D}(i)]^T\mathcal{A}^T \\ &\quad + \mu^2\mathcal{A}\mathcal{U}^T(i)E\{\mathbf{v}(i)\mathbf{v}^T(i)\}\mathcal{U}(i)\mathcal{A}^T. \end{aligned} \quad (\text{B.2})$$

As $\mathbf{v}(i) = \text{col}\{v_1(i), \dots, v_K(i)\} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, substituting (B.2) and (3.28) in (B.1), we have

$$\begin{aligned} \mathcal{R}_{\tilde{\mathbf{w}}}(i) &= \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]\left(E\{\tilde{\mathbf{w}}(i-1)\tilde{\mathbf{w}}^T(i-1)\} \right. \\ &\quad \left. - E\{\tilde{\mathbf{w}}(i-1)\}E\{\tilde{\mathbf{w}}^T(i-1)\}\right)[I_{KM} - \mu\mathcal{D}(i)]^T\mathcal{A}^T \\ &\quad + \mu^2\mathcal{A}\mathcal{U}^T(i)\Sigma\mathcal{U}(i)\mathcal{A}^T \\ &= \mathcal{A}[I_{KM} - \mu\mathcal{D}(i)]\mathcal{R}_{\tilde{\mathbf{w}}}(i-1)[I_{KM} - \mu\mathcal{D}(i)]^T\mathcal{A}^T + \mu^2\mathcal{A}\mathcal{U}^T(i)\Sigma\mathcal{U}(i)\mathcal{A}^T, \end{aligned} \quad (\text{B.3})$$

which is a recursive expression to calculate the covariance of $\tilde{\mathbf{w}}(i)$. Recalling the

definition of $\mathcal{F}(i, j)$ in (3.30):

$$\mathcal{F}(i, j) = \begin{cases} \prod_{n=j}^i \mathcal{A}[I_{KM} - \mu \mathcal{D}(i+j-n)] & \text{if } i \geq j, \\ I_{KM} & \text{if } i < j, \end{cases}$$

we can calculate the recursion in (B.3), setting $\mathcal{R}_{\tilde{w}}(-1) = 0$, as follows:

$$\begin{aligned} \mathcal{R}_{\tilde{w}}(0) &= \mu^2 \mathcal{A} \mathcal{U}^T(0) \Sigma \mathcal{U}(0) \mathcal{A}^T = \mu^2 \mathcal{H}(0), \\ \mathcal{R}_{\tilde{w}}(1) &= \mathcal{A}[I_{KM} - \mu \mathcal{D}(1)] \mathcal{R}_{\tilde{w}}(0) [I_{KM} - \mu \mathcal{D}(1)]^T \mathcal{A}^T + \mu^2 \mathcal{A} \mathcal{U}^T(1) \Sigma \mathcal{U}(1) \mathcal{A}^T \\ &= \mu^2 \left(\mathcal{F}(1, 1) \mathcal{H}(0) \mathcal{F}^T(1, 1) + \mathcal{H}(1) \right), \\ \mathcal{R}_{\tilde{w}}(2) &= \mathcal{A}[I_{KM} - \mu \mathcal{D}(2)] \mathcal{R}_{\tilde{w}}(1) [I_{KM} - \mu \mathcal{D}(2)]^T \mathcal{A}^T + \mu^2 \mathcal{A} \mathcal{U}^T(2) \Sigma \mathcal{U}(2) \mathcal{A}^T \\ &= \mu^2 \left(\mathcal{F}(2, 1) \mathcal{H}(0) \mathcal{F}^T(2, 1) + \mathcal{F}(2, 2) \mathcal{H}(1) \mathcal{F}^T(2, 2) + \mathcal{H}(2) \right), \\ \mathcal{R}_{\tilde{w}}(3) &= \mathcal{A}[I_{KM} - \mu \mathcal{D}(3)] \mathcal{R}_{\tilde{w}}(2) [I_{KM} - \mu \mathcal{D}(3)]^T \mathcal{A}^T + \mu^2 \mathcal{A} \mathcal{U}^T(3) \Sigma \mathcal{U}(3) \mathcal{A}^T \\ &= \mu^2 \left(\mathcal{F}(3, 1) \mathcal{H}(0) \mathcal{F}^T(3, 1) + \mathcal{F}(3, 2) \mathcal{H}(1) \mathcal{F}^T(3, 2) \right. \\ &\quad \left. + \mathcal{F}(3, 3) \mathcal{H}(2) \mathcal{F}^T(3, 3) + \mathcal{H}(3) \right), \\ &\dots \\ \mathcal{R}_{\tilde{w}}(i) &= \mu^2 \left(\sum_{j=0}^{i-1} \mathcal{F}(i, j+1) \mathcal{H}(j) \mathcal{F}^T(i, j+1) + \mathcal{H}(i) \right). \end{aligned}$$

From the definition of $\mathcal{F}(i, j)$, it holds that

$$\mathcal{F}(i, j+1) = I_{KM}, \quad \text{if } i = j.$$

Therefore, a non-recursive expression for the covariance matrix of $\tilde{w}(i)$ can be given as

$$\mathcal{R}_{\tilde{w}}(i) = \mu^2 \sum_{j=0}^i \mathcal{F}(i, j+1) \mathcal{H}(j) \mathcal{F}^T(i, j+1), \quad (\text{B.4})$$

with $\mathcal{H}(j) = \mathcal{A} \mathcal{U}^T(j) \Sigma \mathcal{U}(j) \mathcal{A}^T$.

APPENDIX C

First order Taylor expansion for $\sigma_k(i)$

First, we recall that the statistic variance can be given by (3.42), which is

$$\sigma_k^2(i) = \sum_{n=1}^{i+1} \mu^{2n} (\mathbf{w}_2 - \mathbf{w}_1)^T P_{2n,k}(i) (\mathbf{w}_2 - \mathbf{w}_1),$$

where we defined the matrix $P_{2n,k}(i)$ as the k th $M \times M$ diagonal block of matrix $\mathcal{P}_{2n}(i)$, which appears in equation (3.41):

$$\mathcal{R}_{\tilde{\mathbf{w}}}(i) = \sum_{n=1}^{i+1} \mu^{2n} \mathcal{P}_{2n}(i).$$

Let us name the scalar $(\mathbf{w}_2 - \mathbf{w}_1)^T P_{2n,k}(i) (\mathbf{w}_2 - \mathbf{w}_1)$ as $\phi_{2n,k}(i)$, which does not depend on μ . Then (3.42) becomes

$$\sigma_k^2(i) = \sum_{n=1}^{i+1} \mu^{2n} \phi_{2n,k}(i). \quad (\text{C.1})$$

Let us define

$$\sigma_k(i) = \sqrt{\sigma_k^2(i)}. \quad (\text{C.2})$$

A first order approximation for (C.2) around $\mu = 0$ is given by

$$\sigma_k(i) \approx \sigma_k(i) \Big|_{\mu=0} + \mu \frac{d}{d\mu} \sigma_k(i) \Big|_{\mu=0} = \mu \frac{d}{d\mu} \sigma_k(i) \Big|_{\mu=0}. \quad (\text{C.3})$$

Differentiating (C.2), we have

$$\begin{aligned} \frac{d}{d\mu} \sigma_k(i) &= \frac{d}{d\mu} \sqrt{\sigma_k^2(i)} = \frac{1}{2\sqrt{\sigma_k^2(i)}} \frac{d}{d\mu} \sigma_k^2(i) = \frac{\sum_{n=1}^{i+1} 2n \mu^{2n-1} \phi_{2n,k}(i)}{2\sqrt{\sum_{n=1}^{i+1} \mu^{2n} \phi_{2n,k}(i)}} \\ &= \frac{\mu \sum_{n=1}^{i+1} n \mu^{2n-2} \phi_{2n,k}(i)}{\mu \sqrt{\sum_{n=1}^{i+1} \mu^{2n-2} \phi_{2n,k}(i)}}, \end{aligned}$$

and assuming $\mu \neq 0$, we have

$$\frac{d}{d\mu} \sigma_k(i) = \frac{\sum_{n=1}^{i+1} n \mu^{2n-2} \phi_{2n,k}(i)}{\sqrt{\sum_{n=1}^{i+1} \mu^{2n-2} \phi_{2n,k}(i)}} = \frac{\phi_{2,k}(i) + \sum_{n=2}^{i+1} n \mu^{2n-2} \phi_{2n,k}(i)}{\sqrt{\phi_{2,k}(i) + \sum_{n=2}^{i+1} \mu^{2n-2} \phi_{2n,k}(i)}}. \quad (\text{C.4})$$

Using the approximation in (C.3) around $\mu = 0$ (but for $\mu \neq 0$) and the result in (C.4), we have

$$\sigma_k(i) \approx \mu \frac{\phi_{2,k}(i)}{\sqrt{\phi_{2,k}(i)}} = \mu \sqrt{\phi_{2,k}(i)}, \quad (\text{C.5})$$

and, therefore, the resulting first order approximation is

$$\sigma_k(i) \approx \mu \sqrt{(\mathbf{w}_2 - \mathbf{w}_1)^T P_{2,k}(i) (\mathbf{w}_2 - \mathbf{w}_1)}, \quad (\text{C.6})$$

which is the result presented in (3.43).

APPENDIX D

Network with 100 nodes

The network configuration we used to simulated our algorithm in Subsection 3.4.5 is presented here. Figure 23 shows the connections among the nodes (numbered from 1 to 100) in a matrix form, where rows and columns represent the nodes, and the colors in the positions indicate if a nodes in the corresponding row and columns are connected. Yellow represents connection, while blue indicates no connection. The purpose of this matrix is to display the connectivity of the nodes in the network.

Figure 23: Matrix showing the connections among nodes. Rows and columns represent the nodes (numenerated from 1 to 100), and a position colored in yellow indicates a connection between the corresponding nodes of the row and the column, while blue indicates no connection.

Tables 4, 5 and 6 display the values of α_k , $\sigma_{u,k}^2$ and $\sigma_{v,k}^2$. The rows of the tables indicate the tens of the node number (1 to 100), while the columns indicate the ones.

Table 4: Values of α_k for each node k .

	1	2	3	4	5	6	7	8	9	10
0	0.6632	0.4549	0.6831	0.7599	0.5006	0.8906	0.3377	0.7519	0.4655	0.2249
10	0.6362	0.7385	0.4318	0.4149	0.7973	0.8126	0.5285	0.7664	0.1135	0.5542
20	0.2737	0.7200	0.1192	0.3876	0.2236	0.2394	0.6871	0.1765	0.5836	0.1815
30	0.8584	0.2067	0.5683	0.6986	0.8214	0.3550	0.1611	0.4374	0.8110	0.3735
40	0.6320	0.2495	0.6079	0.6578	0.5973	0.3895	0.3002	0.3945	0.5337	0.1063
50	0.2339	0.3048	0.8010	0.5207	0.5150	0.1521	0.2113	0.7315	0.6669	0.3803
60	0.6374	0.5415	0.6123	0.6979	0.4966	0.4621	0.6021	0.1465	0.5291	0.3143
70	0.5218	0.5608	0.8386	0.6005	0.3317	0.6710	0.7283	0.5478	0.3635	0.6637
80	0.3658	0.4939	0.4219	0.7107	0.6407	0.1438	0.7349	0.2973	0.7495	0.8155
90	0.4432	0.5940	0.3349	0.1518	0.4886	0.8255	0.3209	0.1559	0.6060	0.5786

Table 5: Values of $\sigma_{u,k}^2$ for each node k .

	1	2	3	4	5	6	7	8	9	10
0	0.7794	0.9010	0.7356	0.8031	0.8876	0.9725	0.5319	0.9621	0.9652	0.7790
10	0.5762	0.7663	0.8148	0.5278	0.6908	0.9378	0.6620	0.5289	0.9873	0.5370
20	0.7967	0.7571	0.8337	0.6836	0.8255	0.6989	0.9921	0.5671	0.9399	0.6483
30	0.6013	0.8143	0.5142	0.7049	0.6786	0.7193	0.6590	0.5564	0.7940	0.7407
40	0.7967	0.7087	0.8369	0.6119	0.5741	0.9262	0.9146	0.5921	0.5115	0.9550
50	0.7632	0.6393	0.7622	0.9983	0.8900	0.8909	0.6599	0.6647	0.6863	0.6706
60	0.6048	0.7837	0.6257	0.9696	0.6575	0.5864	0.8653	0.6197	0.8896	0.5416
70	0.7607	0.5164	0.8557	0.6362	0.6484	0.9597	0.6106	0.9550	0.9753	0.7209
80	0.6354	0.7797	0.8128	0.6077	0.8276	0.6021	0.8310	0.9766	0.9648	0.5550
90	0.5089	0.6364	0.7165	0.9797	0.7936	0.5284	0.9880	0.5143	0.6464	0.5486

Table 6: Values of $\sigma_{v,k}^2$ for each node k .

	1	2	3	4	5	6	7	8	9	10
0	0.2546	0.2959	0.4467	0.4146	0.4601	0.2857	0.4446	0.3874	0.4759	0.2632
10	0.4850	0.3883	0.4521	0.4592	0.3004	0.4297	0.4481	0.4729	0.4074	0.2768
20	0.3623	0.4397	0.4997	0.2755	0.3903	0.4893	0.4434	0.3561	0.3408	0.4532
30	0.3422	0.2974	0.4609	0.3276	0.3032	0.4285	0.2944	0.3328	0.3257	0.4399
40	0.4843	0.3102	0.4571	0.3224	0.2639	0.4276	0.3245	0.3842	0.4919	0.2836
50	0.4394	0.4447	0.2753	0.2927	0.2673	0.4803	0.4565	0.2721	0.3580	0.2636
60	0.3992	0.3661	0.3208	0.4937	0.2940	0.4372	0.3244	0.3158	0.3113	0.4386
70	0.3971	0.3551	0.3053	0.4026	0.2904	0.2575	0.3569	0.3154	0.3910	0.2837
80	0.4380	0.4010	0.4573	0.3461	0.3891	0.4633	0.3214	0.3196	0.2542	0.3318
90	0.2604	0.4748	0.3055	0.2982	0.4233	0.3843	0.2862	0.2948	0.3422	0.4723

APPENDIX E

Definition of features used in the SHM system

rise time: time from the capture of an AE hit to its maximum value (amp).

counts: number of excursions of the AE signal over the AE threshold.

energy: integral of the rectified voltage signal over the duration of the AE hit.

asl: average signal level, measures the “averaged” amplitude of the AE signal.

duration: time from the first threshold crossing to the end of the last threshold crossing of the AE signal.

amp: maximum value of the AE hit.

rms: root mean square of the AE hit.

pcnts: number of AE counts between the AE hit start and the pea amplitude.

strength: similar to the feature energy, but calculated over the entire AE signal dynamic range and is independent of preamp gain.

absenergy: integral of the squared voltage signal over the duration of the AE hit, divided by a reference resistance.

REFERENCES

- [1] C. R. Farrar and K. Worden, *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.
- [2] A. E. Feitosa, V. H. Nascimento, and C. G. Lopes, “Adaptive detection in distributed networks using maximum likelihood detector,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 974–978, Jul. 2018.
- [3] K. Diamanti and C. Soutis, “Structural health monitoring techniques for aircraft composite structures,” *Progress in Aerospace Sciences*, vol. 46, no. 8, pp. 342–352, 2010.
- [4] R. Gutkin, C. Green, S. Vangrattanachai, S. Pinho, P. Robinson, and P. Curtis, “On acoustic emission for failure investigation in cfrp: Pattern recognition and peak frequency analyses,” *Mechanical Systems and Signal Processing*, vol. 25, no. 4, pp. 1393–1407, 2011.
- [5] J. P. McCrory, S. K. Al-Jumaili, D. Crivelli, M. R. Pearson, M. J. Eaton, C. A. Featherston, M. Guagliano, K. M. Holford, and R. Pullin, “Damage classification in carbon fibre composites using acoustic emission: A comparison of three techniques,” *Composites Part B: Engineering*, vol. 68, pp. 424–430, 2015.
- [6] R. Pullin, K. M. Holford, S. L. Evans, M. Baxter, and J. Hensman, “Advanced location and characterisation of damage in complex metallic structures using acoustic emission,” in *Proceedings of the 13th International Conference on Experimental Mechanics ICEM*, vol. 13, 2007, pp. 925–926.
- [7] M. Rabiei and M. Modarres, “Quantitative methods for structural health management using in situ acoustic emission monitoring,” *International Journal of Fatigue*, vol. 49, pp. 81–89, 2013.
- [8] S. Rippengill, K. Worden, K. M. Holford, and R. Pullin, “Automatic classification of acoustic emission patterns,” *Strain*, vol. 39, no. 1, pp. 31–41, 2003.
- [9] D.-A. Tibaduiza, M.-A. Torres-Arredondo, L. Mujica, J. Rodellar, and C.-P. Fritzen, “A study of two unsupervised data driven statistical methodologies for detecting and classifying damages in structural health monitoring,” *Mechanical Systems and Signal Processing*, vol. 41, no. 1, pp. 467–484, 2013.
- [10] P. Vanniamparambil, U. Guclu, and A. Kontsos, “Identification of crack initiation in aluminum alloys using acoustic emission,” *Experimental Mechanics*, vol. 55, no. 5, pp. 837–850, 2015.

- [11] X. Wang, H.-P. Zhang, and X. Yan, "Classification and identification of damage mechanisms in polyethylene self-reinforced laminates by acoustic emission technique," *Polymer Composites*, vol. 32, no. 6, pp. 945–959, 2011.
- [12] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.
- [13] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.
- [14] M. D. Prieto, D. Z. Millan, W. Wang, A. M. Ortiz, J. A. O. Redondo, and L. R. Martinez, "Self-powered wireless sensor applied to gear diagnosis based on acoustic emission," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 15–24, 2016.
- [15] W. C. Wilson and G. M. Atkinson, "Passive wireless sensor applications for NASA's extreme aeronautical environments," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3745–3753, 2014.
- [16] C. Mo and J. Davidson, "Energy harvesting technologies for structural health monitoring applications," in *2013 1st IEEE Conference on Technologies for Sustainability (SusTech)*, 2013, pp. 192–198.
- [17] D. Samson, M. Kluge, T. Becker, T. Otterpohl, and U. Schmid, "Energieautonomer Funknetzwerkknoten für Luftfahrzeuge," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 6, pp. 176–180, 2010.
- [18] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
- [19] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [20] Y. P. Bergamo and C. G. Lopes, "Scalar field estimation using adaptive networks," in *2012 IEEE Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3565–3568.
- [21] P. Binder, M. Muma, and A. M. Zoubir, "Robust and computationally efficient diffusion-based classification in distributed networks," in *2015 IEEE Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 3432–3436.
- [22] S. P. Chepuri and G. Leus, "Sparse sensing for distributed detection," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1446–1460, 2016.
- [23] F. S. Cattivelli and A. H. Sayed, "Distributed detection over adaptive networks using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1917–1932, 2011.
- [24] J.-W. Lee, S.-E. Kim, W.-J. Song, and A. H. Sayed, "Spatio-temporal diffusion strategies for estimation and detection over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4017–4034, 2012.

- [25] S. Al-Sayed, A. M. Zoubir, and A. H. Sayed, "Robust distributed detection over adaptive diffusion networks," in *2014 IEEE Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7233–7237.
- [26] S. Al-Sayed, J. Plata-Chaves, M. Muma, M. Moonen, and A. M. Zoubir, "Node-specific diffusion LMS-based distributed detection over adaptive networks," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 682–697, Feb. 2018.
- [27] P. Braca, S. Marano, V. Matta, and A. H. Sayed, "Large deviations analysis of adaptive distributed detection," in *2014 IEEE Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6112–6116.
- [28] V. Matta, P. Braca, S. Marano, and A. H. Sayed, "Exact asymptotics of distributed detection over adaptive networks," in *2015 IEEE Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 3377–3381.
- [29] —, "Detection over diffusion networks: Asymptotic tools for performance prediction and simulation," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1503–1507.
- [30] —, "Diffusion-based adaptive distributed detection: Steady-state performance in the slow adaptation regime," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4710–4732, 2016.
- [31] —, "Distributed detection over adaptive networks: Refined asymptotics and the role of connectivity," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 442–460, 2016.
- [32] D. Jakovetic, J. M. Moura, and J. Xavier, "Distributed detection over noisy networks: Large deviations analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4306–4320, 2012.
- [33] D. Bajovic, D. Jakovetic, J. M. Moura, J. Xavier, and B. Sinopoli, "Large deviations performance of consensus+ innovations distributed detection with non-gaussian observations," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5987–6002, 2012.
- [34] A. K. Sahu and S. Kar, "Recursive distributed detection for composite hypothesis testing: Nonlinear observation models in additive gaussian noise," *IEEE Trans. Inf. Theory*, 2017.
- [35] C. A. M. Sotomayor, R. P. David, and R. Sampaio-Neto, "Adaptive nonassisted distributed detection in sensor networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 6, pp. 3165–3174, Dec. 2017.
- [36] S. M. Kay, *Fundamentals of statistical signal processing: Detection theory*, vol. 2. Prentice Hall Upper Saddle River, NJ, USA:, 1998.
- [37] T. Derham, S. Doughty, K. Woodbridge, and C. Baker, "Design and evaluation of a low-cost multistatic netted radar system," *IET Radar, Sonar & Navigation*, vol. 1, no. 5, pp. 362–368, 2007.

- [38] K. Tu, T. M. Duman, J. G. Proakis, and M. Stojanovic, "Cooperative MIMO-OFDM communications: Receiver design for doppler-distorted underwater acoustic channels," in *2010 Conference Record of the Forty-Fourth Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 1335–1339.
- [39] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, 2003.
- [40] T. Hesketh, R. C. D. Lamare, and S. Wales, "Joint maximum likelihood detection and link selection for cooperative MIMO relay systems," *IET Communications*, vol. 8, no. 14, pp. 2489–2499, Sep. 2014.
- [41] T. Wang, A. Cano, G. B. Giannakis, and J. N. Laneman, "High-performance cooperative demodulation with decode-and-forward relays," *IEEE Trans. Commun.*, vol. 55, no. 7, pp. 1427–1438, 2007.
- [42] X. Zhu and R. D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Trans. Comput.*, vol. 50, no. 2, pp. 187–191, 2002.
- [43] J. M. Cioffi, "Signal processing and detection," class notes, page 18. [Online]. Available: <http://web.stanford.edu/group/cioffi/doc/book/chap1.pdf>
- [44] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 2007, pp. III–917.
- [45] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.
- [46] N. Bogdanović, D. Ampeliotis, and K. Berberidis, "A coalitional game theoretic outlook on distributed adaptive parameter estimation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 2, pp. 416–429, Jun. 2017.
- [47] *R15I-AST - 150 KHZ INTEGRAL PREAMP
AE SENSOR DATASHEET.* [Online]. Available: http://www.physicalacoustics.com/content/literature/sensors/Model_R15I-AST.pdf
- [48] C. Prete, "Estudo de ensaios de emissao acustica aplicados a testes de aeronaves," Escola Politecnica da USP / EMBRAER, Tech. Rep., 2015.
- [49] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [50] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*, vol. 1. Prentice Hall PTR, 1993.

- [51] A. H. Sayed, *Adaptive filters*. John Wiley & Sons, 2011.
- [52] S. O. Haykin, *Adaptive filter theory*. Pearson Higher Ed, 2013.
- [53] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. IPSN 2005. Fourth Int. Symp. Information Processing in Sensor Networks*, Apr. 2005, pp. 63–70.
- [54] C. G. Lopes, L. F. Chamon, and V. H. Nascimento, “Towards spatially universal adaptive diffusion networks,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 803–807.