ALLAN EDUARDO FEITOSA

# Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks

São Paulo
October, 2023

ALLAN EDUARDO FEITOSA

# Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks

**Corrected Version**

Doctoral Thesis presented to the Polytechnic School of the University of São Paulo to obtain the degree of Doctor of Science.

Concentration area: Electronic Systems

Supervisor:
Prof. Dr. Vítor Heloiz Nascimento

Co-supervisor:
Prof. Dr. Cássio Guimarães Lopes

São Paulo
October, 2023

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: ________________________

Catalogação-na-publicação

Feitosa, Allan Eduardo
Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks / A. E. Feitosa -- versão corr. -- São Paulo, 2023.
116 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Processamento de sinais 2.Teste de hipóteses 3.Algoritmos úteis e específicos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Nome: FEITOSA, Allan Eduardo
Título: Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks
Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Aprovado em: 30 de agosto de 2023

Banca Examinadora

Prof. Dr. Vítor Heloiz Nascimento
Instituição: Universidade de São Paulo, Escola Politécnica
Julgamento: APROVADO

Prof. Dr. Magno Teófilo Madeira da Silva
Instituição: Universidade de São Paulo, Escola Politécnica
Julgamento: APROVADO

Prof. Dr. Fernando Gonçalves de Almeida Neto
Instituição: Universidade Federal Rural de Pernambuco
Julgamento: APROVADO

Prof. Dr. Márcio Holsbach Costa
Instituição: Universidade Federal de Santa Catarina
Julgamento: APROVADO

Prof. Dr. Rodrigo Caiado de Lamare
Instituição: Pontifícia Universidade Católica do Rio de Janeiro
Julgamento: APROVADO

*Para minha mãe, Sônia, com irremediável saudade.*

# ACKNOWLEDGEMENTS

I would like to thank first my thesis advisor, Professor Vítor Heloiz Nascimento, without whom I could have not finished this work, who without doubt gave me brilliant solutions to the problems of my research, and most of all who was very kind and supportive during the hardest months for many of us.

I would also like to thank Professor Cássio Guimarães Lopes, for his technical advice during the development of this thesis, for his review of the texts and support.

I would also like to thank the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) for funding and supporting this research (project 2016/06529-1).

I have special thanks to Paula Morillas de Holanda, who without question made this thesis happen by putting some (more) sense in my head.

I cannot help but be deeply grateful to some special friends. I have to thank to Adriana Roberti, Amalia Moura, Blanda Yamaia, Bruno de Araújo Nascimento, Daniele de Paula, Emerson Feitosa Ferreira Junior, Guilherme Risnik Romeiro, Henrique Cristino Leal, Henrique da Costa de Oliveira, Henrique Reis Wisinewski, Julia Rodrigues Molinari, Laiane Taveira Feitosa Ferreira, Marianne Frida Hildegard Teixeira dos Santos Cerqueira Herman, Matheus C. Conti, Murilo Coelho, Raul Figueiredo de Oliveira and Valdenor dos Anjos Teixeira da Silva.

Lastly, I cannot help but thanking myself, the person I though I knew my whole life, but I could only really get to know very recently, during the time this thesis was being produced. I have discovery in 2021 that I am autistic, and knowing that made me understand that my personal struggles have sense and that I must learn to deal with my limitations the best way I can. For that, I am very proud of myself for succeeding in completing this thesis.

*festīnā lentē!* (σπεῦδε βραδέως!)

-Caesar Augustus-

# ABSTRACT

FEITOSA, A. E. **Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks**. 2023. Thesis (Doctor in Electric Engineering) – Polytechnic School of the University of São Paulo, São Paulo, 2023.

This doctoral thesis describes the results of a detailed research conducted between January 2019 and July 2023 on a new distributed detection algorithm. In general terms, this study deals with the statistical detection problem using smart *Wireless Sensor Networks* (WSNs). In this context a network of sensors is distributed over a site in order to monitor the environment and decide the current *state of nature* based on observations under Gaussian noise. The sensors use embedded computation capabilities to locally process data and communicate wirelessly with closest sensors, enabling the exploitation of cooperative algorithms. More specifically, this study focused on a situation where WSNs are deployed over sites under stringent power conditions; therefore, low computational complexity and low power consumption is highly desired, which led to the development of a detection algorithm suitable for real applications and with a performance that tends to optimum under such restrictions. Moreover, in a world increasingly connected through the *Internet of Things* (IoT) paradigm, algorithms that perform indispensable tasks such as detection and operate with minimum energy consumption are highly sought after. Not incidentally, the main contribution of this thesis is the description of a detector with low computational complexity that approximates the performance expected from an optimal detector in terms of the average probability of error, provided certain conditions are met. The most crucial is *maintaining a slow learning rate* of the distributed algorithm that drives the detection routine, specifically the diffusion LMS (Least Mean Square), a well-known adaptive estimation algorithm for distributed networks. The diffusion LMS is used in this context for data processing and sharing information among sensors throughout the network. Notably, like the Tortoise of the fable, the performance of the detector improves as the value of the LMS step size is reduced, without penalizing the convergence rate in terms of probability of error, and despite the slowing down of the estimation routine at the core of the developed detection algorithm. This counter-intuitive result is explained theoretically and confirmed by simulations. The detection problem presented herein is modeled as a multiple hypothesis test using the Bayesian formulation, and it extends the research conducted during my master degree to a more general situation. Additionally, this thesis includes interesting insights about the value of the initial estimate of the LMS algorithm, paving the way for promising future research.

**Keywords**: Detection in wireless sensor networks (WSN). IoT. Distributed diffusion algorithms. Learning rate. Bayesian hypothesis test.

# RESUMO

FEITOSA, A. E. **Favor the Tortoise over the Hare: a study on an efficient detection algorithm for wireless sensor networks**. 2023. Tese (Doutorado em Engenharia Elétrica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2023.

Esta tese de doutorado descreve os resultados de uma pesquisa detalhada realizada entre janeiro de 2019 e julho de 2023 em um novo algoritmo de detecção distribuída. Em termos gerais, este estudo trata do problema estatístico de detecção usando Redes de Sensores Sem Fio (RSSF) inteligentes. Nesse contexto, uma rede de sensores é distribuída por um local para monitorar o ambiente e decidir o "estado da natureza" (*state of nature*) atual com base em observações sujeitas a ruído de natureza gaussiana. Os sensores utilizam capacidade de computação embarcada para processar localmente os dados e se comunicam sem fio com os sensores mais próximos, permitindo a exploração de algoritmos cooperativos. Mais especificamente, este estudo concentrou-se em uma situação em que as RSSF são implantadas em locais com restrições rigorosas de energia; portanto, é altamente desejável uma baixa complexidade computacional e baixo consumo de energia. Isso levou ao desenvolvimento de um algoritmo de detecção adequado para aplicações reais e com um desempenho que tende ao ótimo sob tais restrições. Além disso, em um mundo cada vez mais conectado por meio do paradigma da Internet das Coisas (*Internet of Things*—IoT), algoritmos que realizam tarefas indispensáveis, como a detecção, e operam com consumo mínimo de energia são muito procurados. Não por acaso, a principal contribuição desta tese é a descrição de um detector com baixa complexidade computacional que se aproxima do desempenho esperado de um detector ótimo em termos da probabilidade média de erro, desde que certas condições sejam atendidas. A condição mais crucial é manter uma taxa de aprendizado lenta do algoritmo distribuído que conduz a rotina de detecção, especificamente o algoritmo *diffusion* LMS (*Least Mean Square*), um conhecido algoritmo adaptativo de estimação para redes distribuídas. O *diffusion* LMS é aqui utilizado para processamento de dados e compartilhamento de informações entre os sensores em toda a rede. De forma notável, assim como a Tartaruga da fábula, o desempenho do detector melhora à medida que o valor do tamanho do passo do LMS é reduzido, sem penalizar a taxa de convergência em termos de probabilidade de erro, apesar da desaceleração da rotina de estimação no cerne do algoritmo de detecção desenvolvido aqui. Esse resultado contraintuitivo é explicado teoricamente e confirmado por simulações. O problema de detecção apresentado aqui é modelado como um teste de múltiplas hipóteses usando a formulação Bayesiana e estende a pesquisa realizada durante meu mestrado para uma situação mais geral. Além disso, esta tese inclui discussões interessantes sobre o valor da estimativa inicial do algoritmo LMS, abrindo caminho para pesquisas futuras promissoras.

**Palavras-chave**: Detecção em redes de sensores sem-fio (RSSF). IoT. Algoritmos distribuídos de difusão. Taxa de Aprendizado. Teste de hipótese bayesiano.

# LIST OF FIGURES

# CONTENTS

# 1 INTRODUCTION

## 1.1 Wireless Sensor Networks: a brief history and motivation

Since their emergence at the very end of the second millennium, wireless sensor networks (WSNs) have garnered considerable interest in the fields of science and engineering [1, 2]. The extensive body of scientific contributions on this subject over the years is a direct consequence of advancements in wireless communications and electronics during the final years of the 1990s [3, 4]. These advancements enabled the development of low-cost, low-power, multi-functional tiny intelligent sensors. This rising interest was also driven by the wide range of potential applications [5–7], which traditionally encompasses, for instance, environmental monitoring such as water and air pollution, forest fire, gas leakage, and coal mining [8–12]. They are also utilized in condition-based maintenance [13], smart buildings and smart cities [14, 15], health care monitoring [16, 17], defense [18], precision agriculture [19–21], vehicle tracking [22, 23], transportation [24], factory instrumentation [25, 26], animal tracking [27] and industrial control [28].

The versatility of WSNs stems from their defining characteristics: an ensemble of tiny low-powered sensor devices, called *nodes*, endowed with an embedded local processing unit. These nodes are densely deployed and easily spread over a geographic area to closely monitor a phenomenon of interest (e.g., humidity, pressure, heat, or vibration) via their sensing interfaces. Moreover, each node is capable of locally processing data before being sent through wireless communication channels. Those features provide WSNs with high flexibility, robustness against environmental obstacles, and simplicity in design and operation [5]. Furthermore, the ability of nodes to communicate enables a high potential for the use of cooperative routines, which can be exploited by adequate algorithms.

Within this vast universe of WSNs, this thesis focuses on the problem of decision making and detection of events. This topic arises naturally in WSNs, as it is often the initial or even the main purpose of any deployed sensing system. The classical distributed layout for detection was first presented in [29], using both Neyman-Pearson and Bayesian formulations within the context of decentralized radar for military purposes. At that time and in the following years, “decentralized” simply meant that the nodes would process and compact data before sending it to a *fusion center*, responsible for making the ultimate decision. For instance, [30] describes a real example of a deployed WSN for volcanic eruption detection in 2004. Consequently, in such a network structure, the designer’s effort was directed towards determining what information each sensor should send to the fusion center. Indeed, there is a vast and rich literature concerning the detection problem using this structure, and the interested reader can explore this by referring to the by-no-means exhaustive list of key works [31–39]. Another feature of these original detectors was that they were *static* detectors, in the sense that the sensors perform a decision only after receiving the whole sequence of observations. This must be distinguished from dynamic *sequential* detectors, which process the observations sequentially in time once they become available [40].

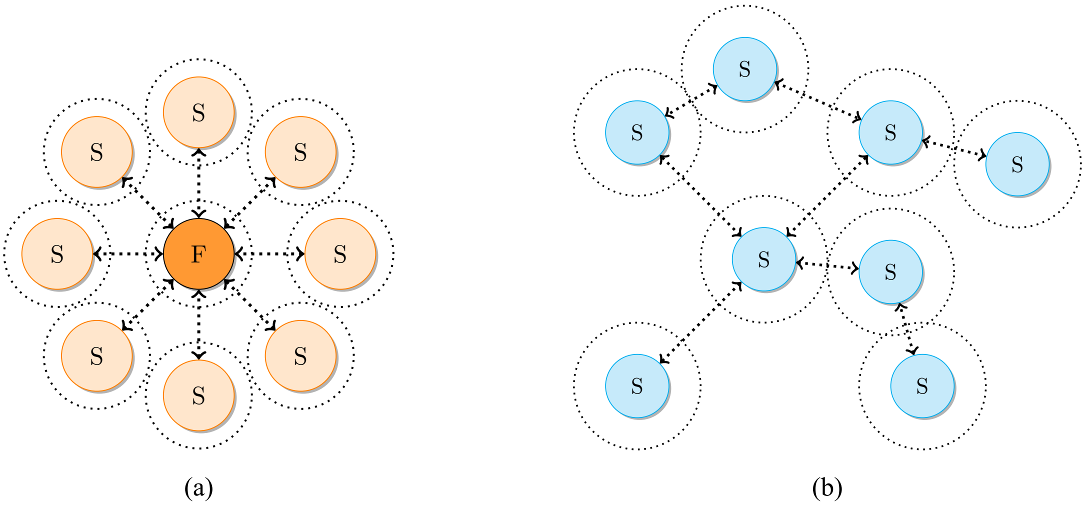


Figure 1 – Abstract representation of the two main setting of WSN.

*Note*: In the fusion-centered network in (a), the sensor nodes collect, process and send data to the fusion center F, which is responsible for the main detection routine using the received data from all nodes. On the other hand, in the *fully* distributed network in (b), sensor nodes cooperatively participate in the decision process by sharing information with their closest neighbors.

Early works on distributed detection assumed that sensor messages are received reliably at the fusion center, ignoring the link variability intrinsic to wireless communications [41]. Thereupon, this issue was addressed in works like [42, 43], which incorporated a fading channel model between sensor and fusion center. However, another way to mitigate the effect of path loss in the communication link is by means of a *fully* distributed network, based on node-to-node message passing (see Figure 1). In this setting, no fusion center is needed, as the nodes themselves cooperatively participate in the decision-making process. Also, communication is allowed only among closely positioned nodes, reducing the attenuation effects due to longer transmission paths. Moreover, a fully distributed WSN has a simpler communication infrastructure, higher scalability and robustness to sensor or link failures, and possibly a more efficient use of limited system resources, such as energy and spectral bandwidth. However, it is essential to note that fusion-centered distributed detection remains an active field of research anymore, as evident from recent works such as [44–46].

In this thesis, sequential and fully distributed cooperative detection in WSNs are considered, envisioning real-word applications. Thus, it is important to realize that to fully leverage the potential flexibility in WSN applications, the sensor nodes must be capable of working under very stringent power requirements, as power supply significantly impacts how WSNs can be deployed in a given environment. Such considerations become even more crucial with the influence of the

Internet of Thing (IoT) paradigm upon the WSN research in the recent years [47]. Not incidentally, WSNs are considered a key technology for materializing the IoT concept, since they perform the fundamental role of detecting events and measuring physical and environmental quantities of interest [48, 49]. Therefore, the development of energy-efficient WSN algorithms compatible with IoT is essential (see Figure 2 for an example of WSN as described in [21]). One major example in this direction is the research effort towards energy harvesting in WSNs [50–54], which involves harnessing from ambient power sources such as vibration, heat, and electromagnetic waves in the immediate surroundings of the network. Equally important is the study and development of specialized low-cost algorithms, a focus on which this thesis centers.

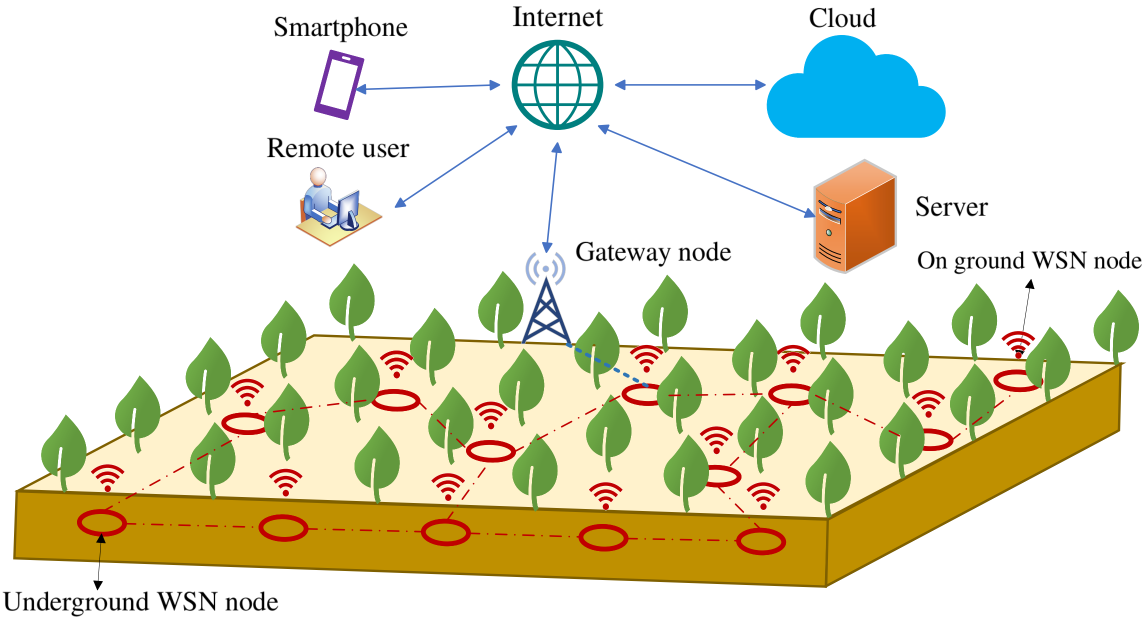


Figure 2 – Conception of a WSN deployed for agriculture applications.

Source: adapted from [21].

*Note*: This is a conception of a WSN deployed for agriculture applications as described in [21]: sensors nodes on the ground and underground monitor the environment of a farm and share information among them. Also, the network is connected via Internet to other services, such as a remote user, smartphone, a server or cloud, following the IoT paradigm.

The goal of a *fully*-distributed and cooperative WSN for detection purposes is for all its nodes to converge to a common decision. Each node updates its guesses from $(i)$ information from its neighbors, and $(ii)$ processed local data from up-to-date local observations. For instance, consider a cooperative underwater network for marine surveillance as described in [55]: small, low-powered and possibly mobile robots patrol an area of coverage, looking for predefined targets to be detected (see Figure 3). If a target is present in the network coverage, it causes some physical perturbation, strong enough to be sensed by the agents and distinguishable from other alternative targets through some statistical procedure. At each time instant, agents decide about the presence of the possible targets based on those measurements and send their locally processed data in the form of a *statistic* (i.e., a preprocessing of the data) to all their closest neighbors via wireless communication. Agents then update their decisions upon the data received, thereby improving the quality of the decision. Then, the process is repeated, with more iterations expected to result in better local decision quality. A similar case that operates along these lines can be found in cooperative unmanned aerial vehicles (UAV) applied, for instance, in fire detection [56, 57].

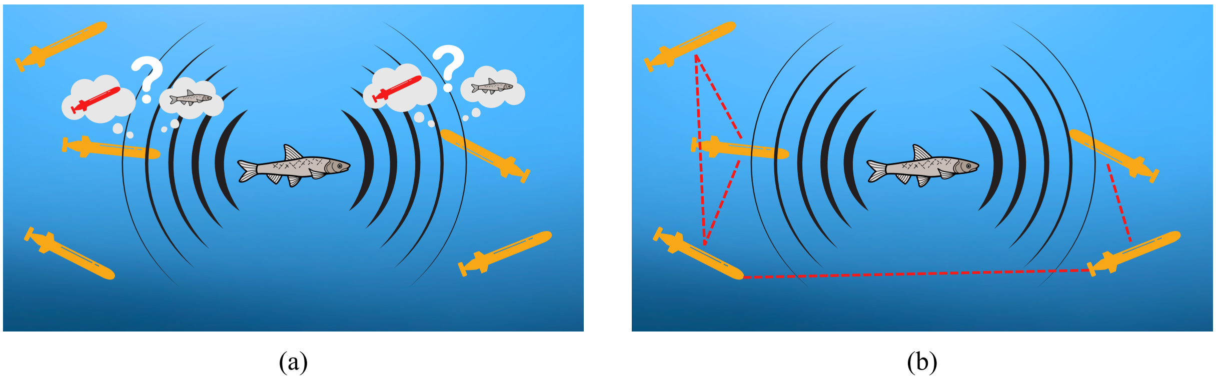

(a) (b)

Figure 3 – Illustration of a practical use of a mobile robotic underwater network.

*Note*: This is an illustration of a practical use of a mobile robotic underwater network for surveillance as described in [55]. In (a), a physical signal is generated, such as pressure variation, and the network patrolling the environment must decide which target generated the signal (in the example, an ordinary fish or a potential dangerous intrusive robot). In (b), the robots communicate with each other sending their collected and processed data to perform a better decision by incorporating more information from other robots.

Another envisioned application of WSN is for Structural Health Monitoring (SHM), which involves implementing a damage detection strategy for aerospace, civil or mechanical engineering infrastructures. SHM includes observing a structure or mechanical system over time using periodically spaced dynamic response measurements, the extraction of damage-sensitive features from these measurements and the statistical analysis of these features to determine the current state of system health [58]. An SHM system can encompass the installation of multiple wireless sensors along the structure under test, receiving spatio-temporal signals containing relevant information about the mechanical stress to which the structure is subjected. Notably, there is a large interest in developing SHM systems for aircraft structures, aiming to increase their reliability and lifetime while reducing maintenance costs.

In my master's thesis, completed in 2018, I participated in a collaborative project between Escola Politécnica da Universidade de São Paulo and Empresa Brasileira de Aeronáutica (EMBRAER), aimed at developing methods for detection of events during tests in aircraft structures to improve the SHM system used by EMBRAER. In these tests the structure is kept on the ground, subjected to load cycles from mechanical actuators that emulate the structural efforts that a plane experiences when airborne, and a collection of piezoelectric sensors is placed along the structure under inspection (see Figure 4). These sensors collect acoustic emission signals and send them to a processing center [59]. That research pursued two main approaches: one focused on developing classifiers using the installed SHM system for short-term implementation, while the other looked at the long-term solution using WSN equipped with a fully distributed detector. The results of the latter generated a paper that was published in the IEEE *Signal Processing Letters* [60].

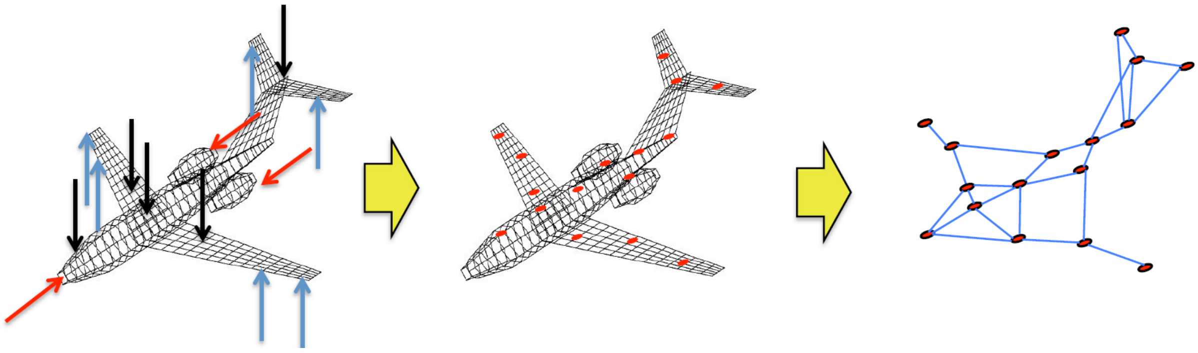


Figure 4 – Conception of a WSN employed in structural health monitoring on airships.

*Note*: This is a conception of a WSN employed in structural health monitoring on airships. (a) Mechanical strains in an aeronautical structure in ground tests. (b) Collection of sensors along the structure to capture acoustic emission signals. (c) Representation of the sensor network.

## 1.2 Most related literature

In recent years, we have witnessed the emergence of two major sequential cooperative strategies for both parameter estimation and detection for distributed settings: consensus algorithms [61–71] and diffusion algorithms [60, 72–82]. In the consensus strategy, nodes run a protocol that asymptotically leads to an agreement about a common value that represents the final detection statistic, shared by all the nodes. The classical consensus algorithm relied on the use of two separate time-scales for environment sensing and consensus [61–64], whereas the *running consensus* technique (also called *consensus+innovation*) unifies these two phases into a simultaneous sequential procedure [65–71].

Diffusion strategies for distributed detection, unlike consensus, offer more flexibility as agreement among nodes is not enforced. Each sensor, after exchanging data in its vicinity, updates its local statistic as a convex combination of local and received information data, aiming at converging to a desired solution with acceptable mean-square error bounds. This is so because diffusion algorithms were originally designed for distributed parameter estimation, relying on algorithms such as the diffusion Least Mean Square (dLMS) [72], the diffusion Recursive Least Square (dRLS) [83], and diffusion Kalman filtering [84]. It has been argued that diffusion strategies entail superior stability compared with consensus under constant *step sizes* [85]. The step size is a key parameter which, roughly speaking, controls the performance of these algorithms over time, affecting decay rate and steady-state value of the mean-square error on estimation. According to [85], consensus strategies generally depends on diminishing values of the step size to achieve asymptotic optimal performance, whereas constant values of it can lead to instability. However, as emphasized in [75], a constant step size is required if one wants the ability of tracking changes (drifts) in the current state of nature, though this also imposes a performance trade-off, as large values of the step size favor tracking speed whereas small values favor detection accuracy. As detailed later on, the step size also plays a decisive role regarding the performance of the proposed detector, since the dLMS is the core of the detection algorithm due to its known distributed qualities.

The dLMS used for distributed detection was first proposed in [77] and further developed in [78], in the form of a distributed two-hypotheses *Neyman-Pearson* (NP) detector [86]: given a *null* hypothesis $H_0$ and an *alternative* $H_1$, the detector seeks to maximize the probability of correct detection of the alternative event (detects $H_1$ when $H_1$ is true), given a desired probability of false alarm (detects $H_1$ when $H_0$ is true). This approach is appropriate, for instance, to detect the presence of a target in a radar. Alternatively, a Bayesian approach can be employed, where the detector aims to minimize the overall detection error, taking into account *prior probabilities* assigned to possible outcomes. When the prior probabilities are assumed to be equal for all possible alternatives, the Bayesian detector reduces to the *maximum likelihood* (ML) detector [86], which can be applied, for instance, in problems where the prior probabilities are uncertain—the ML detector, in this case, minimizes the worst-case error probability [87, thm. 1.2.1], since the maximum error probability in detection occurs when the prior probabilities are equally likely. The already mentioned master's thesis that preceded this research developed a two-hypotheses ML distributed detector [60], using the dLMS similarly as proposed in [78]. After a few months of research in this thesis, those results were extended to detection problems with different probabilities assigned to the two hypotheses [88], resulting in a binary Maximum a Posteriori (MAP) detector. The new detector presented in this work further extends these results to problems with more than two hypotheses, allowing for arbitrary values to be assigned to the prior probabilities.

Traditionally, the majority of the works in distributed detection have focused on asymptotic steady-state behavior, which is specially appealing for a continuous sequential detection, where the sensor network keeps monitoring its environment indefinitely. In this case, the decision accuracy is improved as its performance converges asymptotically to the steady-state. However, equally important is the capacity of making a decision over a limited time interval while ensuring bounds on the detection performance. As argued in [89], this is a crucial question, since in realistic scenarios, one often faces situations where a decision must be taken within a *finite time* to save limited resources or to prevent potential costly damages. As demonstrated in this thesis, the proposed detector aims at detecting events with a performance comparable to an optimal sequential detector within a finite time interval while keeping a simple, feasible detection routine, computationally speaking. Additionally, a designer may ask, prior to deployment, how long the algorithm must run to guarantee a performance benchmark, such as a maximum probability of error. For example, [90] proposes a distributed consensus-based detection algorithm that finds an appropriate stopping rule. In this thesis, an off-line algorithm that can estimate a stopping time is also presented.

The detection problem addressed in this thesis must be contrasted with an apparently similar but fundamentally different situation, which is the *quickest detection of changes in the state of nature*. In this case, the goal is to detect a transition between states as *quickly as possible* after its occurrence. Therefore, in this case both tracking ability and quick response are at a premium [91–93]. Although tracking changes is not a goal of the proposed detector, a simple periodic reset strategy can, in fact, enable the proposed algorithm to track this changes in state adequately.

Finally, to close this section on the most related literature, it is worth mentioning few more recent subareas on distributed detection. For instance, [94] addresses decision learning on distributed *multi-task* networks using a diffusion algorithm, where nodes are grouped into clusters of different addressing decision problems while keeping both a learning rate and tracking ability of the network. Works like [95] and more recently [96] explore distributed network decision problems over the *Social Learning* paradigm, where agents incorporate information from their own observations to form their prior beliefs over a set of hypotheses, and then combine their beliefs locally among neighbors. [97] develops an distributed consensus-based Chernoff test on a *active decision* setting, where the network agents must continuously choose the best action to take over a set of possible actions.

## 1.3 Objectives and contributions

Based on the discussions above, the research objectives can be summarized as follows:

1. Obtain a feasible detection algorithm for cooperative WSNs to determine the current state of the environment being monitored under Gaussian noise.
2. The algorithm in Objective 1 must have *low complexity*, having in mind operation under stringent power limitations.
3. The algorithm in Objective 1 should reach a required detection performance in a finite time interval which can be previously estimated.

The proposed detector satisfies all the objectives above. Now, let us delve into the new contributions of the proposed detector, built upon previous research presented in my master's thesis [59] and published in the *IEEE Signal Processing Letters* [60], where it was first proposed a binary ML distributed detector based on the dLMS algorithm and was shown that the convergence rate of the probability of detection error is *independent* of the value of the LMS step size *if it is chosen to be as small as possible*. In other words, by adopting a slower learning rate in estimation, similar to a “tortoise-like” pace. These results were further extended to a binary

MAP detector during the initial months of this thesis research, which was later published in the *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP) Proceedings [88]. Thereafter, these are the new contributions:

1. A new general detector that generalizes previous results already published [60, 88] to more than two hypotheses with arbitrary values for the prior probabilities.

2. The new detector deals with uncertainty in the parametric vector that represents the different hypotheses.

3. A more detailed theoretical treatment that encompasses the previous results as particular cases.

4. A detailed description of how to obtain an implementable low-complexity algorithm envisioning real detection problems.

5. A proof that as the dLMS estimation rate at the core of the detection algorithm becomes slower, the proposed detector tends to the optimal performance.

6. Additionally, this research includes a simple routine to estimate the stopping time and a strategy to track potential changes in the state of nature.

## 1.4 Notation

### 1.4.1 Basics

We use both normal lowercase "$x$" and uppercase "$X$" for scalars or functions; the uppercase is preferred to denote scalar constants or sets.

Lowercase boldface "$\boldsymbol{x}$" is used for vector quantities. Vectors are always column vectors; a row vector is thus denoted by $\boldsymbol{x}^\top$.

Uppercase boldface "$\boldsymbol{X}$" or calligraphic "$\mathcal{X}$" is used for matrices. A calligraphic font denotes a new matrix that groups together other smaller matrices of kindred origin that refer to each sensor node, using above all the operators "col" or "diag" (see 1.4.2 below for definitions on these functions).

$\boldsymbol{I}_D$ denotes a $D \times D$ identity matrix.

"**0**" denotes either a vector or a matrix filled with zeros, with adequate dimensions depending on the given context.

The operator $\| \boldsymbol{x} \|_2^2$ represents the squared $\ell_2$-norm $(\| \boldsymbol{x} \|_2)^2$ of vector $\boldsymbol{x}$.

### 1.4.2 Special Operators

The “col” operator stacks its arguments (scalars, vectors or matrices) in a column-wise fashion, producing new vectors or new “tall” matrices. For example, for $x_i \in \mathbb{R}, 1 \leq i \leq N$, the operation col$\{x_1, x_2, \dots, x_N\}$ produces a new vector $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_N]^\top \in \mathbb{R}^N$; similarly, for matrices $\boldsymbol{X}_i \in \mathbb{R}^{K\times L}, 1 \leq i \leq N$, the operation col$\{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_N\}$ produces a new matrix $\mathcal{X} = [\boldsymbol{X}_1^\top\ \boldsymbol{X}_2^\top\ \dots\ \boldsymbol{X}_N^\top]^\top \in \mathbb{R}^{NK\times L}$. The vector case follows from the matrix example for $L = 1$.

The “diag” operator arranges its arguments along the diagonal of a square matrix with appropriate dimensions, filling the remaining of the new resulting matrix with zeros. For example, for $x_i \in \mathbb{R}$ and $\boldsymbol{X}_i \in \mathbb{R}^{K\times L}$, $1 \leq i \leq N$, we have

$$\text{diag}\{x_1,\, x_2, \dots,\, x_N\} = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_N \end{bmatrix} \in \mathbb{R}^{N\times N}, \text{ and}$$

$$\text{diag}\{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_N\} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{X}_N \end{bmatrix} \in \mathbb{R}^{KN\times LN}.$$

$\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$; i.e., for $\boldsymbol{A} = (a_{k\ell}) \in \mathbb{R}^{K\times L}$ and $\boldsymbol{B} \in \mathbb{R}^{M\times N}$

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1L}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2L}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K1}\boldsymbol{B} & a_{K2}\boldsymbol{B} & \cdots & a_{KL}\boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{KM\times NL}.$$

### 1.4.3 Time indices

Discrete time-dependent quantities are denoted by indices $i, j, \iota$ inside brackets (e.g., $x[\,i\,], \boldsymbol{x}[\,j\,], \boldsymbol{X}[\,i, j\,], \mathcal{X}[\,i, j, \iota\,]$).

Vectors or matrices that are a collection of quantities from time $i$ down to time 0 are denoted using “$0\!:\!i$” as subscript; e.g.,

$$\begin{aligned} \boldsymbol{x}_{0:i} &= \text{col}\{x[\,i\,],\, x[\,i-1\,], \dots,\, x[\,0\,]\}, \\ \boldsymbol{X}_{0:i} &= \text{diag}\{\boldsymbol{X}[\,i\,], \boldsymbol{X}[\,i-1\,], \dots, \boldsymbol{X}[\,0\,]\}, \end{aligned}$$

### 1.4.4 Indices for states

If the definition of a quantity depends on a determined state of nature, this is denoted by indices $n, m, \nu$ in subscript (e.g., $x_n$, $\boldsymbol{X}_m$).

### 1.4.5 Random quantities and its operators

We represent random quantities by a tilde mark underneath its symbol, as in $\underset{\sim}{x}$ and $\underset{\sim}{\boldsymbol{x}}$; their respective realizations are generally denoted simply as $x$ and $\boldsymbol{x}$.

If $\underset{\sim}{\boldsymbol{x}}$ has a distribution *conditioned* on a given state of nature indexed by $n \in \mathbb{N}$, we denote such dependence as $\underset{\sim}{\boldsymbol{x}}_n$.

The operator $\mathrm{E}(\underset{\sim}{\boldsymbol{x}})$ denotes *unconditional* expectation, and $\mathrm{E}(\underset{\sim}{\boldsymbol{x}} \,|\, H_n)$ denotes an expectation *conditioned* on a given hypothesis $H_n$.

Similarly, the operator $\mathrm{Var}(\underset{\sim}{x})$ denotes the *unconditional* variance of the random variable $\underset{\sim}{x}$, and $\mathrm{Cov}(\underset{\sim}{x}\,, \underset{\sim}{w})$ denotes the *unconditional* covariance between the random variables $\underset{\sim}{x}$ and $\underset{\sim}{w}$. No *conditional* variance or covariance is used throughout this thesis.

We may see the abuse of notation $\mathrm{Var}(\,\underset{\sim}{\boldsymbol{x}}\,)$ or even $\mathrm{Var}(\underset{\sim}{\boldsymbol{x}}^\top)$ to refer to the corresponding covariance matrix of the random vector $\underset{\sim}{\boldsymbol{x}}$, defined as $\mathrm{E}(\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}^\top) - \mathrm{E}(\underset{\sim}{\boldsymbol{x}})\mathrm{E}(\underset{\sim}{\boldsymbol{x}}^\top)$.

### 1.4.6 Indices denoting sensor nodes

Subscripts $k$ and $\ell$ are used to label sensor nodes in the network. If a time varying quantity $\boldsymbol{x}$ depends on a given state $n$ and also on a specific sensor node labeled as $k$, the indices for the state and for the node are separated by a comma, with that of the state appearing first (e.g. $\boldsymbol{x}_{n,k}[\,i\,]$).

### 1.4.7 Avoiding too many subscript indices

Sometimes a quantity $x_{n,k}[\,i\,]$, which depends on state $n$ and is defined at node $k$, can be a realization of a random process that is also conditioned on a given state $m$. In order to avoid denoting this random process with too many subscript indices such as $\underset{\sim}{x}_{m,n,k}[\,i\,]$, instead no subscript for the conditioning is used, and it is denoted by other means such as

$$\begin{gathered}\underset{\sim}{x}_{n,k}[\,i\,] \text{ conditioned on state } m, \text{ or} \\ \underset{\sim}{x}_{n,k}[\,i\,] \mid \text{“state } m\text{”}, \text{ or} \\ \underset{\sim}{x}_{n,k}[\,i\,] \mid H_m,\end{gathered}$$

where $H_m$ refers to the *hypothesis* that “state $m$” is the actual state of nature.

Similarly, quantities such as $\boldsymbol{x}_{0:i}$ that are the realization of a random process conditioned on a given state $n$are are better denoted as “$\underset{\sim}{\boldsymbol{x}}_{0:i}$ conditioned on state $n$” or other variations.

## 1.5 Brief description of the next chapters

The following Chapter 2 formulates the decision problem discussed in this Introduction but restricted to a single isolated sensor node, in order to build on that the theoretical foundations and basic assumptions pertaining both the model of the observed data and the statistical hypothesis test. In this fashion the reader becomes familiar with the terminology and notations used throughout this thesis in the context of a simpler system. The chapter takes advantage of this simplicity to present the important connection between detection and *parameter estimation*, and discuss how such a connection can lead to a more suitable detection strategy.

Next, Chapter 3 expands the discussion of the preceding chapter to a distributed network of wireless sensors and formulates a cooperative detection algorithm based on the estimation algorithm dLMS, using the detection-estimation connection discussed previously. The main mathematical basis of the distributed detector described in this thesis is developed in this chapter, producing a theoretical and “no-approximations” detector which is the base for the analysis and developments in the following chapters.

The performance analysis of the dLMS detector is done in Chapter 4, where it will be proved that its performance is independent of the value of the step size of the LMS algorithm, provided it is sufficiently small, and that this performance tends to optimum.

In Chapter 5, a low-complexity distributed detector, suitable for applications under stringent power conditions, is derived from the theoretical one developed in Chapter 3 by means of a few justifiable approximations, but maintaining the same performance behavior shown in Chapter 4. Also, we discuss the effect of the initial value of the LMS estimate in the performance and in the algorithm complexity. Finally, Chapter 5 concludes presenting a technique to estimate *off-line* the stopping time of the proposed detector and how it can be used to devise a strategy to track eventual changes in the state of nature.

The simulations showing the performance of the low-complexity detector developed in Chapter 5 are presented in Chapter 6, where it is compared with the performances of the theoretical detector of Chapter 3 and with an optimal detector under the same conditions. Additional simulations regarding other topics and later findings are also presented.

Finally, Chapter 7 provides conclusive remarks regarding this thesis and provide some ideas for future work.

# 2 FUNDAMENTALS OF THE DECISION PROBLEM: THE SINGLE NODE CASE

We start addressing the decision problem of a single isolated sensor node, so that the theoretical foundations and basic assumptions can be built on top of a simpler system. We then derive and discuss an optimal test and its *statistic*, and from this examination an alternative test based upon a *recursive parameter estimator* is presented. This connection between detection and estimation produces a *sequential detector*, which is more suitable for the distributed problem being addressed and further developed in Chapter 3. The chapter then concludes with a brief discussion concerning the statistical sufficiency of the parameter estimator.

## 2.1 Framework

Suppose that an isolated node, endowed with sensing and computation capacities, senses and measures events in its surroundings via an observable scalar $d[\,i\,]$, available periodically at every discrete time instant $i$. From these observations, our sensor must infer the current state of its environment, which must be chosen from a set of $N$ known possible states, each indexed by $n = 1, 2, \dots, N$. To each state $n$ is associated a different $D$-dimensional random vector of parameters $\underset{\sim}{\boldsymbol{w}}_n$, called *state vector*, and their prior distributions for each possible state form a family of normally distributed vectors, given as

$$\underset{\sim}{\boldsymbol{w}}_n \sim \mathcal{N}(\boldsymbol{\theta}_n, \boldsymbol{\Sigma}_{\boldsymbol{w}}) \in \mathbb{R}^D, \quad n \in \{1, \dots, N\}, \tag{2.1}$$

where $\{\boldsymbol{\theta}_n\}_{n=1}^N$ are known vector means and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ is independent of any specific state and represents our uncertainty about the true value of any *realization* $\boldsymbol{w}$ of the random state vector $\underset{\sim}{\boldsymbol{w}}_n$. Our observable scalar $d[\,i\,]$ is related to $\boldsymbol{w}$ through the linear model

$$d[\,i\,] = \boldsymbol{u}^{\top}[\,i\,]\boldsymbol{w} + v[\,i\,], \tag{2.2}$$

where $\boldsymbol{u}[\,i\,] \in \mathbb{R}^D$ is a known "input" vector, available at every time $i$, and the scalar $v[\,i\,]$ is a measurement noise, modeled as a realization of a white noise process with $\underset{\sim}{v}[\,i\,] \sim \mathcal{N}(0, \sigma_v^2)$ and assumed to be independent of $\underset{\sim}{\boldsymbol{w}}_n$. In summary, our node must decide from a noisy observation $d[\,i\,]$ and a known vector $\boldsymbol{u}[\,i\,]$ which state is at place at time $i$. In addition, assume that there is no change of states during the period of time under consideration, unless it is expressly stated.

The interpretation of the quantities $d_k[\,i\,]$, $\boldsymbol{u}_k[\,i\,]$ and $\boldsymbol{w}$ can vary substantially depending on the specific application where the linear model in (2.2) is valid. This is particularly the case in the field of Adaptive Filtering, which the algorithm LMS—the "core" of the proposed distributed detector described in Chapter 3—is mostly related with. We refer to [98, pp. 168–174] and [99] for some examples of applications of the model. As it is customary in Adaptive Filtering, assume that the input vector $\boldsymbol{u}[\,i\,]$ in Equation (2.2) has a shift delay structure; i.e., given a signal $u[\,j\,] \in \mathbb{R}$, $j = 0, \dots, i$, we have

$$\boldsymbol{u}[\,i\,] = \mathrm{col}\{u[\,i\,], u[\,i-1\,], \dots, u[\,i-D+1\,]\}.$$

For this reason, the model in Equation (2.2) is known as *tapped delay line*, commonly used to model a multi-path channel signal transmission [86, p. 169], and in this context the vector $\boldsymbol{u}[\,i\,]$ is called *regressor vector*, a terminology adopted in Adaptive Filtering and in this thesis.

For most scenarios where the model in Equation (2.2) is applied, the regressor vector is the observation of a random process $\utilde{\boldsymbol{u}}$. The sequence of observed regressors $\{\boldsymbol{u}[\,j\,]\}_{j=0}^{i}$ should be considered *deterministic* at time $i$, in the sense that they have already been observed. Therefore, the solution to the decision problem herein can be seen as designed for this particular sequence of realizations of the random process $\utilde{\boldsymbol{u}}$, and such a solution is in fact optimal and has a closed-form, as we see later on.

Finally, in order to arrange and simplify the description of the decision problem, let us collect all data samples up to instant $i$:

$$\begin{aligned} \boldsymbol{d}_{0:i} &= \mathrm{col}\{d[\,i\,], d[\,i-1\,], \dots, d[\,0\,]\} \in \mathbb{R}^{i+1}, \\ \boldsymbol{U}_{0:i} &= \mathrm{col}\{\boldsymbol{u}^{\top}[\,i\,], \boldsymbol{u}^{\top}[\,i-1\,], \dots, \boldsymbol{u}^{\top}[\,0\,]\} \in \mathbb{R}^{(i+1)\times D}. \end{aligned} \tag{2.3}$$

Thus, given state $n$, the data model in Equation (2.2) can be grouped as

$$\boldsymbol{d}_{0:i} = \boldsymbol{U}_{0:i}\boldsymbol{w} + \boldsymbol{v}_{0:i}\,, \tag{2.4}$$

where $\boldsymbol{v}_{0:i} = \mathrm{col}\{v[\,i\,], \dots, v[\,0\,]\}$ is the realization of a white noise process $\utilde{\boldsymbol{v}}_{0:i} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\big)$ which is independent of $\utilde{\boldsymbol{w}}_n$, and whose covariance matrix is

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}} = \mathrm{diag}\{\sigma_v^2, \dots, \sigma_v^2\} \in \mathbb{R}^{(i+1)\times(i+1)}. \tag{2.5}$$

## 2.2 Formulating a Hypothesis Test

The solution of a decision problem is a rule that associates the observed data with a unique true hypothesis, chosen from a sequence $\{H_n\}_{n=1}^{N}$ of possible alternatives, where each $H_n$ represents the event {“state $n$ is the current state”}. We start defining a function $\mathit{\Phi} : \mathbb{R}^{i+1} \to \mathbb{N}^*$ of the observed data $\boldsymbol{d}_{0:i}$, and the event

$$\{\mathit{\Phi}(\boldsymbol{d}_{0:i}) = n \mid \boldsymbol{U}_{0:i}, H_m\}, \quad n \in \{1, \dots, N\}, \tag{2.6}$$

represents a decision in favor of hypothesis $H_n$ based on the observed data $\boldsymbol{d}_{0:i}$, given a particular matrix $\boldsymbol{U}_{0:i}$ and $H_m$, $m \neq n$, is the true hypothesis. In a Bayesian framework, we assign a cost $(H_n, H_m) \mapsto C_{nm}$ whenever we decide for $H_n$ when $H_m$ is true, and we measure the performance of the decision rule $\varPhi(\boldsymbol{d}_{0:i})$ through the *average decision cost* (also called *Bayes risk function*), given as [86, p. 80]

$$R[\,i\,] = \sum_{n=1}^{N}\sum_{m=1}^{N} C_{nm}\,\mathbb{P}(\{\varPhi(\boldsymbol{d}_{0:i}) = n \mid \boldsymbol{U}_{0:i}, H_m\})\mathbb{P}(H_m).$$

The probabilities $\{\mathbb{P}(H_m)\}_{n=1}^{N}$ above are called the *prior probabilities* and represent our knowledge or belief about the "odds" of the hypotheses *before* any data is observed. Evidently, we have $\mathbb{P}(H_m) = \mathbb{P}(\{\text{"state } n \text{ is the current state"}\})$. In particular, setting $C_{nm} = 1$ if $n \neq m$ and $C_{nm} = 0$ otherwise, the Bayes risk reduces to the *average probability of error* $\xi[\,i\,]$, given as

$$\xi[\,i\,] = \sum_{n=1}^{N}\sum_{\substack{m=1\\ m\neq n}}^{N} \mathbb{P}(\{\varPhi(\boldsymbol{d}_{0:i}) = n \mid \boldsymbol{U}_{0:i}, H_m\})\mathbb{P}(H_m), \tag{2.7}$$

which can be minimized by setting $\varPhi(\boldsymbol{d}_{0:i}) = n$ whenever the probability $\mathbb{P}(H_n \mid \boldsymbol{d}_{0:i}, \boldsymbol{U}_{0:i})$ is maximum. In other words, we choose $H_n$ if

$$\mathbb{P}(H_n \mid \boldsymbol{d}_{0:i}, \boldsymbol{U}_{0:i}) > \mathbb{P}(H_m \mid \boldsymbol{d}_{0:i}, \boldsymbol{U}_{0:i}) \quad \forall m \neq n\,. \tag{2.8}$$

The probabilities above are called *posterior probabilities*, since they represent the chances of each hypothesis to be correct *after* data is observed. Consequently, this test is called *maximum a posteriori* (MAP). Note that the MAP test in Equation (2.8) implies the evaluation of $N$ posterior probabilities and $N-1$ comparisons between pairs in order to find the maximum.

Using Bayes' theorem, we can compare a given pair of hypotheses $\{H_n, H_m\}$ by

$$f(\boldsymbol{d}_{0:i} \mid \boldsymbol{U}_{0:i}, H_n)\mathbb{P}(H_n) \underset{\varPhi\neq n}{\overset{\varPhi\neq m}{\gtrless}} f(\boldsymbol{d}_{0:i} \mid \boldsymbol{U}_{0:i}, H_m)\mathbb{P}(H_m)\,, \tag{2.9}$$

where $f(\boldsymbol{d}_{0:i} \mid \boldsymbol{U}_{0:i}, H_n)$ is the *probability density function* (p.d.f.) of the random vector $\underset{\sim}{\boldsymbol{d}}_{0:i}$ conditioned on $\{\boldsymbol{U}_{0:i}, H_n\}$, which is, from Equation (2.4), given as

$$\underset{\sim}{\boldsymbol{d}}_{0:i} \mid \{\boldsymbol{U}_{0:i}, H_n\} = \boldsymbol{U}_{0:i}\underset{\sim}{\boldsymbol{w}}_n + \underset{\sim}{\boldsymbol{v}}_{0:i}; \tag{2.10}$$

therefore,

$$\underset{\sim}{\boldsymbol{d}}_{0:i} \mid \{\boldsymbol{U}_{0:i}, H_n\} \sim \mathcal{N}\big(\boldsymbol{U}_{0:i}\boldsymbol{\theta}_n\,, \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}\big),$$

where $\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}} = \boldsymbol{U}_{0:i}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{U}_{0:i}^{\top} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}$ is its covariance matrix. Since it is a normal random vector, its p.d.f. is given as

$$f(\boldsymbol{d}_{0:i} \mid \boldsymbol{U}_{0:i}, H_n) = \Big((2\pi)^{\frac{i+1}{2}}\det{}^{\frac{1}{2}}\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}\Big)^{-1} \mathrm{e}^{\tau_n[\,i\,]}, \tag{2.11}$$

where $\tau_n[\,i\,]$ is a function of the data given as

$$\tau_n[\,i\,] = -\frac{1}{2}(\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n)^{\top}\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}(\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n)\,. \tag{2.12}$$

Thus, taking the logarithm on both sides of the inequality in Equation (2.9) and canceling constants out, the test can be reduced to the following equivalent:

$$\tau_n[\,i\,] \underset{\Phi \neq n}{\overset{\Phi \neq m}{\gtrless}} \tau_m[\,i\,] + \gamma_{mn}, \tag{2.13}$$

where $\gamma_{mn} = \ln(\mathbb{P}(H_m)/\mathbb{P}(H_n))$. The functions of data $\{\tau_n[\,i\,]\}_{n=1}^{N}$ used to compare pairs of hypotheses are called *test statistics*. Let us summarize these results in the following proposition:

**Proposition 1.** *A test statistic $\tau_n[\,i\,]$ and the hypothesis test to decide in favor of $H_n$ from $\boldsymbol{d}_{0:i} = \boldsymbol{U}_{0:i}\boldsymbol{w} + \boldsymbol{v}_{0:i}$ are given as*

$$\tau_n[\,i\,] = -\frac{1}{2}(\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}(\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n),$$

$$\tau_n[\,i\,] \underset{\Phi \neq n}{\overset{\Phi \neq m}{\gtrless}} \tau_m[\,i\,] + \gamma_{mn},$$

*where $\gamma_{mn} = \ln(\mathbb{P}(H_m)/\mathbb{P}(H_n))$.*

*Proof.* See equations (2.9) to (2.13) above. □

The test statistic just presented in Equation (2.12) consists in a *batch* process; i.e, in order to evaluate it and make a decision at time $i$, we have to process all the previously collected data at once. Therefore, in such a scenario it is reasonable to evaluate the test statistic only at the end of our observation, when all data are collected. In this thesis, the focus in on a solution that produces intermediate statistics for every time $i$ without the need for data accumulation, since the sensor nodes could benefit from this in-between processing to share their intermediate results with its neighboring nodes, which in turn can combine this new information with their own local statistic, thus improving their performance. This could be achieved, for instance, by using a recursive algorithm, in such a way that the evaluation of these new intermediate statistics depends only on its past values and on the most recently observed data. In the next section, we will discuss the case where these intermediate statistics are estimates of the state vector $\boldsymbol{w}$ and how we can employ it as the core of a detection algorithm.

## 2.3 Detection & Estimation

From our previous discussion, we are interested in producing a test with the following structure:

$$g(\breve{\boldsymbol{w}}[\,i\,] \mid H_n) \underset{\Phi \neq n}{\overset{\Phi \neq m}{\gtrless}} g(\breve{\boldsymbol{w}}[\,i\,] \mid H_m) + \gamma'_{mn}, \tag{2.14}$$

where $\breve{\boldsymbol{w}}[\,i\,] \in \mathbb{R}^D$ is any estimate of $\underset{\sim}{\boldsymbol{w}}_n$, $\gamma'_{mn}$ is an appropriate decision threshold and $g : \mathbb{R}^D \to \mathbb{R}$ is a function which depends on the observable data $\boldsymbol{d}_{0:i}$ only through $\breve{\boldsymbol{w}}[\,i\,]$. To get rid of the *direct* dependence on previously observed data, assume that $\breve{\boldsymbol{w}}[\,i\,]$ can be evaluated recursively in time as follows:

$$\breve{\boldsymbol{w}}[\,i\,] = G(\breve{\boldsymbol{w}}[\,i-1\,],\, d[\,i\,], \boldsymbol{u}[\,i\,]),$$

where $G(\cdot)$ is a function which depends on previous information only through $\breve{\boldsymbol{w}}[\,i-1\,]$. As mentioned, nodes in a cooperative wireless network can take advantage of these local estimates to share information among them, which we should expect will improve the detection performance. Not incidentally, there are efficient estimators whose use was devised for detection in distributed networks, such as the already mentioned dLMS algorithm (see the most related literature in Section 1.2). We will discuss in Chapter 3 how we can formulate a distributed detector based on the dLMS. But first it must be shown that a hypothesis test can be produced from an estimator, as shown in Equation (2.14).

According to the Bayesian philosophy, the criterion typically employed to choose an *estimator*—i.e., a random quantity whose realizations are *estimates*—is to select the one that minimize the *Bayesian mean squared error* (BMSE): given a hypothesis $H_n$, we choose the estimator such that

$$\utilde{\breve{\boldsymbol{w}}}{}^*_n[\,i\,] = \underset{\utilde{\breve{\boldsymbol{w}}}[\,i\,]}{\arg\min}\; \mathrm{E}\Big(\|\, \utilde{\boldsymbol{w}}_n - \utilde{\breve{\boldsymbol{w}}}[\,i\,]\,\|_2^2 \mid \boldsymbol{d}_{0:i}, \boldsymbol{U}_{0:i}\Big).$$

The solution to such problem is the Bayesian *minimum mean square error estimator* (MMSE) [100, p. 316], whose estimates at time $i$ are set as the mean value of the random variable $\utilde{\boldsymbol{w}}_n$ after (conditioned on) the observation of data; put mathematically,

$$\breve{\boldsymbol{w}}^*[\,i\,] = \mathrm{E}(\utilde{\boldsymbol{w}}_n \mid \boldsymbol{d}_{0:i}, \boldsymbol{U}_{0:i}). \tag{2.15}$$

Therefore, assuming the linear model in Equation (2.4), the MMSE can be expressed as [100, p. 326]

$$\breve{\boldsymbol{w}}^*[\,i\,] = \boldsymbol{\theta}_n + \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{U}_{0:i}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\, \boldsymbol{\theta}_n), \tag{2.16}$$

where $\boldsymbol{\theta}_n = \mathrm{E}(\utilde{\boldsymbol{w}}_n)$ (see Equation (2.1)). The MMSE is noteworthy because its estimates can be calculated recursively. Such recursive routine is given by the following equation [100, p. 398]:

$$\breve{\boldsymbol{w}}^*[\,i\,] = \boldsymbol{\omega}[\,i\,]d[\,i\,] + (\boldsymbol{I}_D - \boldsymbol{\omega}[\,i\,]\boldsymbol{u}^{\top}[\,i\,])\breve{\boldsymbol{w}}^*[\,i-1\,], \tag{2.17}$$

where

$$\begin{aligned}
\boldsymbol{\omega}[\,i\,] &= \frac{\boldsymbol{\Omega}[\,i-1\,]\boldsymbol{u}[\,i\,]}{\sigma_v^2 + \boldsymbol{u}^{\top}[\,i\,]\boldsymbol{\Omega}[\,i-1\,]\boldsymbol{u}[\,i\,]} \in \mathbb{R}^D, \text{ and} \\
\boldsymbol{\Omega}[\,i\,] &= (\boldsymbol{I}_D - \boldsymbol{\omega}[\,i\,]\boldsymbol{u}^{\top}[\,i\,])\, \boldsymbol{\Omega}[\,i-1\,] \in \mathbb{R}^{D\times D},
\end{aligned}$$

which must be initialized as $\breve{\boldsymbol{w}}^*[-1] = \boldsymbol{\theta}_n$ and $\boldsymbol{\Omega}[-1] = \boldsymbol{\Sigma}_{\boldsymbol{w}}$. However, if we want to make use of the MMSE, we would need to remove the direct dependence on a specific $\boldsymbol{\theta}_n$, because we do not know the current state $n$ in order to set $\breve{\boldsymbol{w}}^*[-1] = \boldsymbol{\theta}_n$. Furthermore, this would be preposterous, since if we knew the correct $\boldsymbol{\theta}_n$ to choose, the detection problem would be solved already. In order to fix this inconsistency, let us notice that the MMSE is unbiased, since from Equation (2.16)

$$\mathrm{E}(\underset{\sim}{\boldsymbol{w}}_n - \mathrm{E}(\check{\underset{\sim}{\boldsymbol{w}}}{}^*_n)) = \mathrm{E}(\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n) = \mathbf{0};$$

in fact, we can remove the direct dependence on a specific $\boldsymbol{\theta}_n$ by discarding the unbiased condition. By substituting $\boldsymbol{\theta}_n$ in Equation (2.16) by an arbitrary constant vector $\boldsymbol{b} \in \mathbb{R}^D$, we produce a new *biased* estimator $\underset{\sim}{\widehat{\boldsymbol{w}}}$ based on the MMSE, whose estimates are given as

$$\widehat{\boldsymbol{w}}[\,i\,] = \boldsymbol{b} + \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{U}_{0:i}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}). \tag{2.18}$$

which is a biased estimate for all $\boldsymbol{b} \neq \boldsymbol{\theta}_n$. Although $\underset{\sim}{\widehat{\boldsymbol{w}}}$ is not optimal according to the BMSE criterion due to the insertion of a bias, we will see ahead in Section 2.4 that, from a detection perspective, this bias does not change the detector performance because it is a *sufficient statistic* for $\boldsymbol{\theta}_n$. Furthermore, the new estimator $\underset{\sim}{\widehat{\boldsymbol{w}}}$ can also be updated by the same recursive algorithm given in Equation (2.17) for the MMSE, setting the initial estimate to $\widehat{\boldsymbol{w}}[-1] = \boldsymbol{b}$.

Next, we define a new test statistic $\tau_n'[\,i\,]$ from $\tau_n[\,i\,]$ in Equation (2.12) that depends on $\boldsymbol{d}_{0:i}$ only through the estimate $\widehat{\boldsymbol{w}}[\,i\,]$ and is equivalent to $\tau_n[\,i\,]$ with regard to detection. Since this is an important result, let us state it in the following proposition:

**Proposition 2.** *The following hypothesis test with its corresponding statistic test, namely*

$$\tau_n'[\,i\,] \underset{\Phi \neq n}{\overset{\Phi \neq m}{\gtrless}} \tau_m'[\,i\,] + \gamma_{mn},$$
$$\tau_n'[\,i\,] = (\widehat{\boldsymbol{w}}[\,i\,] - \boldsymbol{b})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}) - \frac{1}{2} (\boldsymbol{\theta}_n - \boldsymbol{b})^{\top} \boldsymbol{U}_{0:i}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} \boldsymbol{U}_{0:i} (\boldsymbol{\theta}_n - \boldsymbol{b}), \tag{2.19}$$

*is completely equivalent to the original test in Equation* (2.13) *in a statistical hypothesis test point of view.*

*Proof.* The new test statistic $\tau_n'[\,i\,]$ is obtained from $\tau_n[\,i\,]$. First, by using the following algebraic manipulation:

$$\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n = (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}) - (\boldsymbol{U}_{0:i}\boldsymbol{\theta}_n - \boldsymbol{U}_{0:i}\boldsymbol{b}),$$

we can expand Equation (2.12) to obtain

$$\begin{aligned} \tau_n[\,i\,] = &-\frac{1}{2} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}) + (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} \boldsymbol{U}_{0:i} (\boldsymbol{\theta}_n - \boldsymbol{b}) \\ &- \frac{1}{2} (\boldsymbol{\theta}_n - \boldsymbol{b})^{\top} \boldsymbol{U}_{0:i}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} \boldsymbol{U}_{0:i} (\boldsymbol{\theta}_n - \boldsymbol{b}). \end{aligned} \tag{2.20}$$

Let $h(\boldsymbol{d}_{0:i})$ be the first right-hand side term above; i.e.,

$$h(\boldsymbol{d}_{0:i}) = -\frac{1}{2} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}).$$

Note that $h(\boldsymbol{d}_{0:i})$ does not change when $H_n$ changes; thus, $h(\boldsymbol{d}_{0:i})$ cannot influence the test result and it is always canceled out when we compare a pair $\{\tau_n[\,i\,], \tau_m[\,i\,]\}$ of statistics. Hence, we can remove it by defining $\tau'_n[\,i\,] = \tau_n[\,i\,] - h(\boldsymbol{d}_{0:i})$. Next, substitute in Equation (2.20) the following identity:

$$(\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{d}_{0:i}} \boldsymbol{U}_{0:i} = (\widehat{\boldsymbol{w}}[\,i\,] - \boldsymbol{b})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\,,$$

which can be obtained by manipulating Equation (2.18) accordingly:

$$\begin{aligned} \widehat{\boldsymbol{w}}[\,i\,] - \boldsymbol{b} = \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{U}^\top_{0:i} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{d}_{0:i}} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}) \Rightarrow \\ \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} (\widehat{\boldsymbol{w}}[\,i\,] - \boldsymbol{b}) = \boldsymbol{U}^\top_{0:i} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{d}_{0:i}} (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{b}), \end{aligned}$$

and then transposing both sides of the equation. Thus, we have

$$\tau'_n[\,i\,] = (\widehat{\boldsymbol{w}}[\,i\,] - \boldsymbol{b})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} (\boldsymbol{\theta}_n - \boldsymbol{b}) - \frac{1}{2} (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{U}^\top_{0:i} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{d}_{0:i}} \boldsymbol{U}_{0:i} (\boldsymbol{\theta}_n - \boldsymbol{b}), \tag{2.21}$$

which is, as in Equation (2.14), a function that depends on the observable data $\boldsymbol{d}_{0:i}$ only through the estimate $\widehat{\boldsymbol{w}}[\,i\,]$. From the test in Equation (2.13), we also conclude that $\gamma'_{mn} = \gamma_{mn}$. Therefore, the hypothesis test presented in Proposition 2 is equivalent to the test in Equation (2.13). □

Note that, because the statistic $\tau_n[\,i\,]$ in Equation (2.12) does not depend on the particular value of $\boldsymbol{b}$, and the test in Equation (2.19) is equivalent to that in Equation (2.13), we conclude that the value chosen for $\boldsymbol{b}$ can not, in principle, influence the detection performance. This result justifies replacing the MMSE in Equation (2.17) with a more consistent biased version without losing anything by doing this replacement from a detection perspective. However, although we can choose any value for $\boldsymbol{b}$, there are in fact more adequate choices when we take approximations for the test statistics as we develop the implementable distributed detector later in Chapter 5.

## 2.4 Sufficiency

The detection equivalence between the tests Equation (2.13) and Equation (2.19) can be addressed by showing that the estimator $\widehat{\boldsymbol{w}}[\,i\,]$ as defined in Equation (2.18) is a *sufficient statistic* for $\boldsymbol{\theta}_n$, given that $H_n$ is true (notice that, when $H_n$ is true, $\boldsymbol{\theta}_n$ is a constant deterministic parameter). We can verify this fact by examining the p.d.f. of $\underset{\sim}{\boldsymbol{d}}_{0:i}$ in Equation (2.11), which is

$$f(\boldsymbol{d}_{0:i} \mid \boldsymbol{U}_{0:i}, H_n) = \left( (2\pi)^{\frac{i+1}{2}} \det^{\frac{1}{2}} \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}} \right)^{-1} \mathrm{e}^{h(\boldsymbol{d}_{0:i}|\boldsymbol{U}_{0:i})}\, \mathrm{e}^{\tau'_n[\,i\,]}. \tag{2.22}$$

As we can see, the p.d.f. in Equation (2.22) is a product of a function $\mathrm{e}^{h(\boldsymbol{d}_{0:i})}$ depending only on the data and function $\mathrm{e}^{\tau'_n[i]}$. Also, remember that $\tau'_n[\,i\,] = g(\widehat{\boldsymbol{w}}[\,i\,] \mid H_n)$ is a function that depends on the data $\boldsymbol{d}_{0:i}$ only through the estimate $\widehat{\boldsymbol{w}}[\,i\,]$. According to the *Neyman-Fisher Factorization Theorem* [100, p. 104], $\widehat{\boldsymbol{w}}[\,i\,]$ is, therefore, a sufficient statistic for $\boldsymbol{\theta}_n$. This fact justifies considering the estimates $\widehat{\boldsymbol{w}}[\,i\,]$, instead of $\boldsymbol{d}_{0:i}$, as our *de facto* data sample, since all the information needed to perform a detection about $\underset{\sim}{\boldsymbol{w}}_n$ (or, equivalently, about $\boldsymbol{\theta}_n$) is embedded in $\widehat{\boldsymbol{w}}[\,i\,]$. Framing the detection problem directly from an estimator expands the set of possible algorithms we can use, since there are many works on distributed estimation in WSNs that propose efficient cooperative algorithms for this task, such as the already mentioned dLMS. This is, perhaps, the motif for many works on detection in distributed WSNs that rely on solutions for the estimation problem, such as [78–81]. Moreover, finding a truly distributed sufficient statistic for WSNs in real world applications is likely an unattainable task. Therefore, there is a practical interest in finding a distributed estimator that can be used in lieu of an unattainable sufficient statistic and that can approximate, under certain conditions, an optimal performance. In the next chapters, we head towards a distributed detection solution based on the dLMS estimator, as well describe its main features (Chapter 3), and its interesting, useful and somewhat counter-intuitive performance behavior (Chapter 4), notably a decay rate of the probability of error that is independent of the learning rate of the LMS algorithm.

# 3 THE DISTRIBUTED DETECTOR USING THE DIFFUSION–LMS

In this chapter, we start the discussion about the proposed cooperative detector based on the dLMS, aiming at exploiting the diffusion of information within the wireless network. From the connection between detection and estimation discussed in the previous chapter, we develop the necessary mathematical tools to show that we can produce a cooperative $N$-ary hypothesis test whose "core" is the dLMS. The detector developed in this chapter is ideal in the sense that no approximations are taken during its conception, and is the base for the performance analysis in Chapter 4 and for the construction of a feasible low-complexity detector in Chapter 5.

## 3.1 Cooperative estimation with $K$ sensor nodes

Consider now a wireless network composed of $K$ different sensor nodes, where each node $k$ has its local inputs $d_k[\,i\,]$ and a regressor vector $\boldsymbol{u}_k[\,i\,]$ (seen as a realization of a random process $\underset{\sim}{\boldsymbol{u}}_k[\,i\,]$). All nodes are inserted in the same environment and therefore undergo the same ongoing state, and must decide, as the single node case, which is the current state. Likewise, each node is capable of processing data locally. They also can communicate bi-directionally with other nodes in their vicinity in order to share information in a fully distributed setting (i.e., there is no fusion center), and considering resource limitations such as power. Likewise, to each state $n$ is associated a random and unknown state vector $\underset{\sim}{\boldsymbol{w}}_n$ whose prior distribution depends on the current state $n$ according to Equation (2.1). We make the same assumptions taken in Equation (2.2) for the single node case: the inputs $d_k[\,i\,], \boldsymbol{u}_k[\,i\,]$ are related to a realization $\boldsymbol{w}$ of $\underset{\sim}{\boldsymbol{w}}_n$ by

$$d_k[\,i\,] = \boldsymbol{u}_k^{\top}[\,i\,]\boldsymbol{w} + v_k[\,i\,], \tag{3.1}$$

where $\underset{\sim}{v}_k[\,i\,] \sim \mathcal{N}\big(0, \sigma^2_{v_k}\big)$ is the noise at node $k$. As for the isolated node in Chapter 2, assume that the regressor vectors $\boldsymbol{u}_k[\,i\,]$ have a shift delay structure; i.e.,

$$\boldsymbol{u}_k[\,i\,] = \text{col}\{u_k[\,i\,],\, u_k[\,i-1\,],\, \dots\,,\, u_k[\,i-D+1\,]\}.$$

where $u[\,j\,] \in \mathbb{R}$, $j = 0, \dots\,, i$ is an input signal.[1]

Note that a general optimal solution for this decision problem would be unfeasible in practice, since a sufficient statistic would require that all data be collected at the same place (in other words, at a fusion center). Notwithstanding, the focus herein is not on global solutions, but in formulating and describing practical algorithms that exploit the sharing of information among nodes across the network, in such a way that we can achieve a detection performance that we expect to be superior in general to what sensor nodes would attain if they acted with no cooperation.

[1] For an example, see [78] for a application of distributed detection using the linear model in the context of a cognitive radio problem.

Based on our previous discussion, we are employing a preprocessing of data via a cooperative estimation algorithm that can satisfactorily exploit information sharing in the network, instead of an impractical sufficient statistic. Therefore, assume that each node $k$ can calculate recursively from $d_k[\,i\,]$ and $\boldsymbol{u}_k[\,i\,]$ a local estimate $\hat{\boldsymbol{w}}_k[\,i\,]$ of the state vector $\boldsymbol{w}$ and send it to its neighbors at every time $i$. For this task the diffusion–LMS (dLMS) was chosen as the cooperative estimation algorithm due to its well known good performance and simplicity for adaptive estimation on WSNs [72]. The dLMS estimates at nodes are denoted as $\hat{\boldsymbol{w}}_k[\,i\,]$, whose recursive algorithm can be outlined by the following equations:

$$\boldsymbol{\psi}_k[\,i\,] = \hat{\boldsymbol{w}}_k[\,i-1\,] + \mu_k \boldsymbol{u}_k[\,i\,](d_k[\,i\,] - \boldsymbol{u}_k^{\top}[\,i\,]\hat{\boldsymbol{w}}_k[\,i-1\,]), \tag{3.2}$$

$$\hat{\boldsymbol{w}}_k[\,i\,] = \sum_{\ell\,\in\,\vartheta_k} a_{\ell k}\boldsymbol{\psi}_\ell[\,i\,]\,, \tag{3.3}$$

where $a_{\ell k} \geq 0$ and $\sum_{\ell=1}^{K} a_{\ell k} = 1 \;\forall k, \ell \in \{1, 2, \ldots, K\}$.

Equation (3.2) shows that the intermediate estimates $\boldsymbol{\psi}_k[\,i\,]$ are the output of a local LMS update from the previous estimate $\hat{\boldsymbol{w}}_k[\,i-1\,]$ using local up-to-date inputs $d_k[\,i\,]$ and $\boldsymbol{u}_k[\,i\,]$. Next, the estimates $\hat{\boldsymbol{w}}_k[\,i\,]$ are obtained via a convex combination of the intermediate estimates $\boldsymbol{\psi}_\ell[\,i\,]$ from neighbors connected to node $k$, which constitute the *vicinity* of node $k$, represented by the set

$$\vartheta_k = \{\ell : \text{node } \ell \text{ is connected with node } k\}$$

(also consider that every node $k$ is connected to itself). The weights of this combination are given by the coefficients $a_{\ell k}$. Figure 5 shows an illustration of the diffusion-LMS. Finally, $\mu_k > 0$ is the constant step size of the LMS algorithm[2].

From the structure of the dLMS algorithm in Equation (3.2) and Equation (3.3), notice that each estimate $\hat{\boldsymbol{w}}_k[\,i\,]$ condenses information from a set of measurable quantities

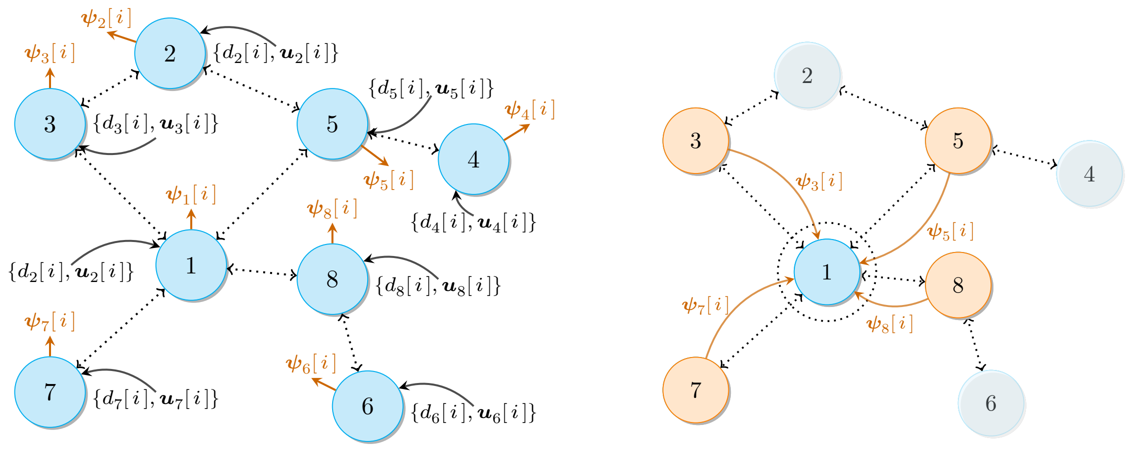
$$\varUpsilon_k[\,i\,] = \{d_k[\,i\,], \boldsymbol{u}_k[\,i\,]\} \cup \left\{ \bigcup_{\ell\in\vartheta_k} \{\hat{\boldsymbol{w}}_\ell[\,i-1\,]\} \right\}. \tag{3.4}$$

Therefore, it is reasonable to consider $\hat{\boldsymbol{w}}_k[\,i\,]$ as a sample of a process which depends on those quantities in the set $\varUpsilon_k[\,i\,]$; in other words, we could build a hypothesis test as in Equation (2.9) using corresponding p.d.f. of the estimator $\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}[\,i\,]$ as follows:

$$f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,],\, H_n) \underset{\varPhi_k \neq n}{\overset{\varPhi_k \neq m}{\gtrless}} f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,],\, H_m),$$

where $\varPhi_k : \mathbb{R}^D \to \mathbb{N}^*$ is the decision rule at node $k$, which is a function of the estimate $\hat{\boldsymbol{w}}_k[\,i\,]$. However, in order to do so, first we need to find its distribution of $\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}$ for each node $k$ and hypothesis $H_n$ , which it is done in the next section.

[2] The step size regulates the dynamics of the learning ability of the LMS algorithm—in a general sense, a greater value of $\mu$ yields a faster convergence rate, whereas smaller values reduce the steady–state quadratic error. These two performance metrics are, in general, conflicting; thus, one should set a value of the step size that balances both metrics. However, as shown later, the proposed detector does not show this conflict between transient state and steady state, which means we do not need to set any compromise value for the step size.



(a) (b)

Figure 5 – Diffusion LMS algorithm

*Note*: Example of a dLMS algorithm running in a small wireless network. In (a) each node $k$ in the network individually collects a pair $\{d_k[i], \boldsymbol{u}_k[i]\}$ and processes it to produce $\boldsymbol{\psi}_k[i]$ according to Equation (3.2). In (b) we observe the information sharing with focus on node $k = 1$: neighboring nodes send their local estimates produced in the last step to node 1, which in turn sends its own local estimate. The final estimate is produced by taking a convex combination of the received and local estimates, as described in Equation (3.3).

## 3.2 The distribution of the local dLMS estimator

As we were discussing, before building a hypothesis test based on the dLMS algorithm, first we need to find the distribution of random vector $\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}[i]$ from which the dLMS estimate $\hat{\boldsymbol{w}}_k[i]$ in Equation (3.3) is the corresponding realization. We start by rearranging the dLMS equations in Equation (3.2) and Equation (3.3) to find an expression for the p.d.f. of the local estimates. First, let us define the matrix

$$\boldsymbol{Y}_k[i] = \boldsymbol{I}_D - \mu_k \boldsymbol{u}_k[i]\boldsymbol{u}_k^\top[i], \tag{3.5}$$

and rewrite Equation (3.2) and Equation (3.3) more compactly as

$$\hat{\boldsymbol{w}}_k[i] = \sum_{\ell \in \vartheta_k} a_{\ell k}(\mu_\ell \boldsymbol{u}_\ell[i] d_\ell[i] + \boldsymbol{Y}_\ell[i]\hat{\boldsymbol{w}}_\ell[i-1]). \tag{3.6}$$

We define a *combination matrix* $\boldsymbol{A} = (a_{\ell k})$ that, by the definition of the coefficients $a_{\ell k}$ in Equation (3.3), displays the property $\boldsymbol{A}^\top \mathbb{1}_K = \mathbb{1}_K$, where $\mathbb{1}_K \in \mathbb{R}^K$ and

$$\mathbb{1}_K = \text{col}\{1, 1, \dots, 1\}.$$

In other words, $\boldsymbol{A}$ is a *left stochastic* matrix.

Let us also define the following matrices:

$$\begin{aligned}
\mathcal{A} &= \boldsymbol{A}^{\top} \otimes \boldsymbol{I}_{D} \in \mathbb{R}^{DK \times DK},^{3}\\
\mathcal{M} &= \operatorname{diag}\{\mu_1 \boldsymbol{I}_D, \, \dots \, , \, \mu_k \boldsymbol{I}_D, \, \dots \, , \, \mu_K \boldsymbol{I}_D\} \in \mathbb{R}^{DK \times DK},\\
\boldsymbol{d}[\,i\,] &= \operatorname{col}\{d_1[\,i\,], \, \dots \, , \, d_k[\,i\,], \, \dots \, , \, d_K[\,i\,]\} \in \mathbb{R}^{K},\\
\mathcal{U}[\,i\,] &= \operatorname{diag}\{\boldsymbol{u}_1[\,i\,], \, \dots \, , \, \boldsymbol{u}_k[\,i\,], \, \dots \, , \, \boldsymbol{u}_K[\,i\,]\} \in \mathbb{R}^{DK \times K},\\
\mathcal{Y}[\,i\,] &= \operatorname{diag}\{\boldsymbol{Y}_1[\,i\,], \, \dots \, , \boldsymbol{Y}_k[\,i\,], \, \dots \, , \boldsymbol{Y}_K[\,i\,]\} \; \in \; \mathbb{R}^{DK \times DK}. \qquad (3.7)
\end{aligned}$$

We will use these new matrices to wrap the dLMS equations in Equation (3.6) for all nodes as a single expression, which can be done by defining the collection of estimates[4]

$$\hat{\mathrm{w}}[\,i\,] = \operatorname{col}\{\hat{\boldsymbol{w}}_1[\,i\,], \, \dots \, , \, \hat{\boldsymbol{w}}_k[\,i\,], \, \dots \, , \, \hat{\boldsymbol{w}}_K[\,i\,]\} \in \mathbb{R}^{DK}. \qquad (3.8)$$

Thus, we can rewrite $\hat{\mathrm{w}}[\,i\,]$ above as

$$\begin{aligned}
\hat{\mathrm{w}}[\,i\,] &= \begin{bmatrix} \sum_{\ell \in \vartheta_1} a_{\ell 1}(\mu_\ell \boldsymbol{u}_\ell[\,i\,] d_\ell[\,i\,] + \boldsymbol{Y}_\ell[\,i\,] \hat{\boldsymbol{w}}_\ell[\,i-1\,]) \\ \sum_{\ell \in \vartheta_2} a_{\ell 2}(\mu_\ell \boldsymbol{u}_\ell[\,i\,] d_\ell[\,i\,] + \boldsymbol{Y}_\ell[\,i\,] \hat{\boldsymbol{w}}_\ell[\,i-1\,]) \\ \vdots \\ \sum_{\ell \in \vartheta_k} a_{\ell K}(\mu_\ell \boldsymbol{u}_\ell[\,i\,] d_\ell[\,i\,] + \boldsymbol{Y}_\ell[\,i\,] \hat{\boldsymbol{w}}_\ell[\,i-1\,]) \end{bmatrix} \\
&= \begin{bmatrix} a_{11}\boldsymbol{I}_D & a_{21}\boldsymbol{I}_D & \cdots & a_{K1}\boldsymbol{I}_D \\ a_{12}\boldsymbol{I}_D & a_{22}\boldsymbol{I}_D & \cdots & a_{K2}\boldsymbol{I}_D \\ \vdots & \vdots & \ddots & \vdots \\ a_{1K}\boldsymbol{I}_D & a_{2K}\boldsymbol{I}_D & \cdots & a_{KK}\boldsymbol{I}_D \end{bmatrix} \cdot \left( \begin{bmatrix} \mu_1 \boldsymbol{u}_1[\,i\,] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_2 \boldsymbol{u}_2[\,i\,] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_K \boldsymbol{u}_K[\,i\,] \end{bmatrix} \begin{bmatrix} d_1[\,i\,] \\ d_2[\,i\,] \\ \vdots \\ d_K[\,i\,] \end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix} \boldsymbol{Y}_1[\,i\,] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_2[\,i\,] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{Y}_K[\,i\,] \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{w}}_1[\,i-1\,] \\ \hat{\boldsymbol{w}}_2[\,i-1\,] \\ \vdots \\ \hat{\boldsymbol{w}}_K[\,i-1\,] \end{bmatrix} \right) \\
&= \mathcal{A}(\mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{d}[\,i\,] + \mathcal{Y}[\,i\,]\hat{\mathrm{w}}[\,i-1\,]). \qquad (3.9)
\end{aligned}$$

Next, for analysis purposes, it is convenient to obtain a non-recursive version of Equation (3.9); but first, define the auxiliary matrix $\mathcal{F}[\,i,j\,] \in \mathbb{R}^{DK \times DK}$ as

$$\mathcal{F}[\,i,j\,] = \begin{cases} \prod_{\iota=j}^{i} \mathcal{A}\,\mathcal{Y}[\,i+j-\iota\,], & \text{if } j \leq i, \\ \boldsymbol{I}_{DK}\,, & \text{if } j > i. \end{cases} \qquad (3.10)$$

**Proposition 3.** *A non-recursive expression for* $\hat{\mathrm{w}}[\,i\,]$ *in Equation* (3.9) *is given as*

$$\hat{\mathrm{w}}[\,i\,] = \sum_{j=0}^{i} \mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\boldsymbol{d}[\,j\,] + \mathcal{F}[\,i,0\,]\hat{\mathrm{w}}[-1], \; i \geq 0, \qquad (3.11)$$

*where* $\hat{\mathrm{w}}[-1] = \operatorname{col}\{\hat{\boldsymbol{w}}_1[-1], \, \dots \, , \, \hat{\boldsymbol{w}}_k[-1], \, \dots \, , \, \hat{\boldsymbol{w}}_K[-1]\}$ *is the collection of initial estimates.*

[3] Matrix $\mathcal{A}$ is defined in terms of $\boldsymbol{A}^{\top}$ instead of $\boldsymbol{A}$ to maintain a cleaner notation.

[4] As this thesis deals with fully distributed solutions for the detection problem in WSN, the collection of all estimates as defined herein only exists abstractly and for the sake of analysis only.

*Proof.* We can prove Proposition 3 by a mathematical inductive process. Evaluating Equation (3.9) for $i = 0$, we have

$$\hat{\mathbf{w}}[0] = \mathcal{AMU}[0]\boldsymbol{d}[0] + \mathcal{AY}[0]\hat{\mathbf{w}}[-1] \tag{3.12}$$

According to the definition of $\mathcal{F}[i,j]$ in Equation (3.10) , we have

$$\mathcal{F}[i,i] = \mathcal{AY}[i] \text{ and } \mathcal{F}[i,i+1] = \boldsymbol{I}_{DK};$$

thus, we can rewrite Equation (3.12) as

$$\begin{aligned}\hat{\mathbf{w}}[0] &= \mathcal{F}[0,1]\mathcal{AMU}[0]\boldsymbol{d}[0] + \mathcal{F}[0,0]\hat{\mathbf{w}}[-1]\\ &= \sum_{j=0}^{0} \mathcal{F}[0,j+1]\mathcal{AMU}[j]\boldsymbol{d}[j] + \mathcal{F}[0,0]\hat{\mathbf{w}}[-1],\end{aligned}$$

which shows that Equation (3.11) is valid for $i = 0$. Next, suppose it is valid for a given $i = i_0 \geq 0$; let us verify if it is also valid for $i_0 + 1$; again, from Equation (3.9), we have

$$\begin{aligned}\hat{\mathbf{w}}[i_0+1] &= \mathcal{A}(\mathcal{MU}[i_0+1]\boldsymbol{d}[i_0+1] + \mathcal{Y}[i_0+1]\hat{\mathbf{w}}[i_0])\\ &= \mathcal{AMU}[i_0+1]\boldsymbol{d}[i_0+1]\\ &\quad + \mathcal{AY}[i_0+1]\left(\sum_{j=0}^{i_0} \mathcal{F}[i_0,j+1]\mathcal{AMU}[j]\boldsymbol{d}[j] + \mathcal{F}[i_0,0]\hat{\mathbf{w}}[-1]\right)\\ &= \mathcal{F}[i_0+1,i_0+2]\mathcal{AMU}[i_0+1]\boldsymbol{d}[i_0+1]\\ &\quad + \sum_{j=0}^{i_0} \mathcal{AY}[i_0+1]\mathcal{F}[i_0,j+1]\mathcal{AMU}[j]\boldsymbol{d}[j] + \mathcal{AY}[i_0+1]\mathcal{F}[i_0,0]\hat{\mathbf{w}}[-1].\end{aligned}$$

Notice that, from the definition of $\mathcal{F}[i,j]$ in Equation (3.10), it holds that

$$\begin{aligned}\mathcal{F}[i,j] &= (\mathcal{AY}[i]) \cdot (\mathcal{AY}[i-1]) \cdot (...) \cdot (\mathcal{AY}[j])\\ &= \mathcal{AY}[i]\Big((\mathcal{AY}[i-1]) \cdot (...) \cdot (\mathcal{AY}[j])\Big)\\ &= \mathcal{AY}[i]\prod_{\iota=j}^{i-1} \mathcal{AY}[i+j-\iota]\\ &= \mathcal{AY}[i]\mathcal{F}[i-1,j];\end{aligned}$$

hence, we can rewrite $\hat{\mathbf{w}}[i_0+1]$ above as

$$\begin{aligned}\hat{\mathbf{w}}[i_0+1] &= \mathcal{F}[i_0+1,i_0+2]\mathcal{AMU}[i_0+1]\boldsymbol{d}[i_0+1]\\ &\quad + \sum_{j=0}^{i} \mathcal{F}[i_0+1,j+1]\mathcal{AMU}[j]\boldsymbol{d}[j] + \mathcal{F}[i_0+1,0]\hat{\mathbf{w}}[-1]\\ &= \sum_{j=0}^{i+1} \mathcal{F}[i_0+1,j+1]\mathcal{AMU}[j]\boldsymbol{d}[j] + \mathcal{F}[i_0+1,0]\hat{\mathbf{w}}[-1].\end{aligned}$$

Therefore, we can see that expression Equation (3.11) holds for $i_0 + 1$ supposing that it holds for $i_0$. Since it was shown that it is valid for $i = 0$, by a mathematical inductive process we must conclude that it holds for any $i \geq 0$. □

It will be proven beneficial for our analysis to rewrite the summation of the non-recursive expression of $\hat{\mathbf{w}}[\,i\,]$ in Equation (3.11) as a product of matrices; keeping that in mind, we define a few more collections of data in the network:

$$\begin{aligned} \boldsymbol{U}[\,i\,] &= \mathrm{col}\{\boldsymbol{u}_1^\top[\,i\,],\, \dots\,, \boldsymbol{u}_k^\top[\,i\,],\, \dots\,, \boldsymbol{u}_K^\top[\,i\,]\} \,\in\, \mathbb{R}^{K\times D}, \\ \boldsymbol{v}[\,i\,] &= \mathrm{col}\{v_1[\,i\,],\, \dots\,, v_k[\,i\,],\, \dots\,, v_K[\,i\,]\} \,\in\, \mathbb{R}^{K}. \end{aligned} \tag{3.13}$$

Next, let us extend the definitions of the quantities in equations (2.3) and (2.4) to encompass all data of every node $k$ in the network:

$$\begin{aligned} \boldsymbol{d}_{0:i} &= \mathrm{col}\{\boldsymbol{d}[\,i\,],\, \dots\,,\, \boldsymbol{d}[\,j\,],\, \dots\,,\, \boldsymbol{d}[\,0\,]\} \,\in\, \mathbb{R}^{(i+1)K}, \\ \boldsymbol{U}_{0:i} &= \mathrm{col}\{\boldsymbol{U}[\,i\,],\, \dots\,,\, \boldsymbol{U}[\,j\,],\, \dots\,,\, \boldsymbol{U}[\,0\,]\} \,\in \mathbb{R}^{(i+1)K\times D}, \\ \boldsymbol{v}_{0:i} &= \mathrm{col}\{\boldsymbol{v}[\,i\,],\, \dots\,,\, \boldsymbol{v}[\,j\,],\, \dots,\, \boldsymbol{v}[\,0\,]\} \,\in\, \mathbb{R}^{(i+1)K}, \end{aligned}$$

where $\boldsymbol{d}[\,i\,]$ was defined in Equation (3.7) and $\boldsymbol{U}[\,i\,]$, $\boldsymbol{v}[\,i\,]$ in Equation (3.13). Please note that the definitions made for the single node case in Equation (2.3) are just a special case of this extended definition and correspond to $K = 1$. Also, the linear model in Equation (2.4), which we must recall:

$$\boldsymbol{d}_{0:i} = \boldsymbol{U}_{0:i}\boldsymbol{w} + \boldsymbol{v}_{0:i}\,, \tag{3.14}$$

evidently holds for the whole network of $K \geq 1$ nodes now. Next, we define a matrix $\mathcal{L}[\,i\,]$ as

$$\mathcal{L}[\,i\,] = \Big[\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,] \quad \dots \quad \mathcal{F}[\,i,j\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j-1\,] \quad \dots \quad \mathcal{F}[\,i,1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,0\,]\Big], \tag{3.15}$$

which can be combined with the time increasing vector $\boldsymbol{d}_{0:i}$ in order to rewrite the summation in Equation (3.11) as follows:

$$\sum_{j=0}^{i} \mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\boldsymbol{d}[\,j\,] = \mathcal{L}[\,i\,]\boldsymbol{d}_{0:i}.$$

Thereby, we can rewrite $\hat{\mathbf{w}}[\,i\,]$ in Equation (3.11) using Equation (3.14) as

$$\begin{aligned} \hat{\mathbf{w}}[\,i\,] &= \mathcal{L}[\,i\,]\boldsymbol{d}_{0:i} + \mathcal{F}[\,i,0\,]\hat{\mathbf{w}}[-1] \\ &= \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i}\boldsymbol{w} + \mathcal{L}[\,i\,]\boldsymbol{v}_{0:i} + \mathcal{F}[\,i,0\,]\hat{\mathbf{w}}[-1]. \end{aligned} \tag{3.16}$$

Define $\mathcal{P}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i} \in \mathbb{R}^{DK\times D}$; the following proposition will be useful to rewrite Equation (3.16) more appropriately.

**Proposition 4.** *For the matrices $\mathcal{F}[\,i,j\,]$ defined in Equation (3.10), $\mathcal{P}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i}$, and $\mathbb{I} = \mathrm{col}\{\boldsymbol{I}_D, \dots, \boldsymbol{I}_D\}$, it holds that*

$$\mathcal{F}[\,i,0\,]\,\mathbb{I} = \mathbb{I} - \mathcal{P}[\,i\,]\,. \tag{3.17}$$

*Proof.* From the definition of $\mathcal{F}[\,i,j\,]$ in Equation (3.10), we can rewrite it as

$$\mathcal{F}[\,i,j\,] = \mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{Y}[\,j\,].$$

From the definitions of $\mathcal{Y}[\,i\,]$ in Equation (3.7) and $\mathbf{Y}_k[\,i\,]$ in Equation (3.5), we can rewrite $\mathcal{Y}[\,i\,]$ as

$$\mathcal{Y}[\,i\,] = \boldsymbol{I}_{DK} - \mathcal{M}\,\mathcal{U}[\,i\,]\,\mathcal{U}^{\top}[\,i\,]; \tag{3.18}$$

thus, we can write

$$\mathcal{F}[\,i,j\,] = \mathcal{F}[\,i,j+1\,]\,\mathcal{A} - \mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,].$$

Thus, repeatedly substituting $\mathcal{F}[\,i,j+1\,]$, in the first term above from $j=0$ up to $i$, we obtain

$$\begin{aligned}\mathcal{F}[\,i,0\,] &= \mathcal{F}[\,i,1\,]\mathcal{A} - \mathcal{F}[\,i,1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,]\\ &= \mathcal{F}[\,i,2\,]\mathcal{A}^2 - \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,]\mathcal{A}^j\\ &\;\vdots\\ &= \mathcal{A}^{i+1} - \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,]\mathcal{A}^j \end{aligned} \tag{3.19}$$

From $\mathcal{P}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i}$ and the definition of $\mathcal{L}[\,i\,]$ given in Equation (3.15), we can write $\mathcal{P}[\,i\,]$ as follows:

$$\mathcal{P}[\,i\,] = \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\boldsymbol{U}[\,j\,]. \tag{3.20}$$

Note that we can rewrite the matrix $\boldsymbol{U}[\,i\,]$ defined in Equation (3.13) as $\boldsymbol{U}[\,i\,] = \mathcal{U}^{\top}[\,i\,]\,\mathbb{I}$, where $\mathbb{I} = \mathrm{col}\{\boldsymbol{I}_D,\ \dots,\ \boldsymbol{I}_D\}$. Furthermore, note that, by the definition of the combination matrix (recall that $\boldsymbol{A}^{\top}\mathbb{1}_K = \mathbb{1}_K$), it is true that $\mathcal{A}\,\mathbb{I} = \mathbb{I}$. Consequently, it is also true that $\mathcal{A}^i\mathbb{I} = \mathbb{I},\ \forall i \in \mathbb{N}$. Thereby, let us rewrite $\boldsymbol{U}[\,i\,] = \mathcal{U}[\,i\,]\mathcal{A}^i\mathbb{I}$ and then substitute it in Equation (3.20), in order to obtain

$$\mathcal{P}[\,i\,] = \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,]\mathcal{A}^j\mathbb{I}$$

Now, multiply both sides of Equation (3.19) by $\mathbb{I}$ to obtain

$$\begin{aligned}\mathcal{F}[\,i,0\,]\,\mathbb{I} &= \mathcal{A}^{i+1}\mathbb{I} \;\; - \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\,\mathcal{U}^{\top}[\,j\,]\mathcal{A}^j\mathbb{I}\\ &= \mathbb{I} - \mathcal{P}[\,i\,],\end{aligned}$$

which completes the proof. □

From the proposition above, if we set $\hat{\boldsymbol{w}}_k[-1] = \boldsymbol{b}$ for every node $k$, we can substitute Equation (3.17) in Equation (3.16), resulting in

$$\hat{\mathrm{w}}[\,i\,] = \mathcal{P}[\,i\,]\boldsymbol{w} + \mathcal{L}[\,i\,]\boldsymbol{v}_{0:i} + (\mathbb{I} - \mathcal{P}[\,i\,])\boldsymbol{b}.$$

Therefore, we conclude that the vector $\hat{\mathrm{w}}[\,i\,]$ is a realization of the random vector

$$\underset{\sim}{\hat{\mathrm{w}}}_n[\,i\,] = \mathcal{P}[\,i\,]\underset{\sim}{\boldsymbol{w}}_n + (\mathbb{I} - \mathcal{P}[\,i\,])\boldsymbol{b} + \mathcal{L}[\,i\,]\underset{\sim}{\boldsymbol{v}}_{0:i}. \tag{3.21}$$

Hence, since the state vector process $\underset{\sim}{\boldsymbol{w}}_n$ and the noise $\underset{\sim}{\boldsymbol{v}}_{0:i}$ are independent, we have, if $H_n$ is true, that

$$\hat{\underset{\sim}{\mathrm{w}}}_n[\,i\,] \sim \mathcal{N}(\mathcal{P}[\,i\,]\boldsymbol{\theta}_n + (\mathbb{I} - \mathcal{P}[\,i\,])\boldsymbol{b},\, \mathcal{S}[\,i\,]), \tag{3.22}$$

where $\mathcal{S}[\,i\,] \in \mathbb{R}^{DK\times DK}$ is defined as the covariance matrix of $\hat{\underset{\sim}{\mathrm{w}}}_n[\,i\,]$. It is shown in Appendix A that $\mathcal{S}[\,i\,]$ can be written from Equation (3.21) as

$$\mathcal{S}[\,i\,] = \mathcal{P}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{P}^{\top}[\,i\,] + \mathcal{L}^{\top}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}[\,i\,], \tag{3.23}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}$ is the extended covariance matrix of $\underset{\sim}{\boldsymbol{v}}_{0:i}$ :

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}} = \mathrm{E}(\underset{\sim}{\boldsymbol{v}}_{0:i}\underset{\sim}{\boldsymbol{v}}_{0:i}^{\top}) = \boldsymbol{I}_{i+1} \otimes \boldsymbol{\Sigma}_{\boldsymbol{v}},\ \text{where } \boldsymbol{\Sigma}_{\boldsymbol{v}} = \mathrm{diag}\{\sigma^2_{v_1},\, \dots\, ,\, \sigma^2_{v_K}\}. \tag{3.24}$$

Also, it is convenient for our analysis to define $\mathcal{Z}[\,i\,]$ as the second term in the right-hand side of Equation (3.23):

$$\mathcal{Z}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}^{\top}[\,i\,]. \tag{3.25}$$

Having the distribution of $\hat{\underset{\sim}{\mathrm{w}}}_n[\,i\,]$ in Equation (3.22), our next step is to describe the distributions of the various local estimators $\hat{\underset{\sim}{\boldsymbol{w}}}_{n,k}[\,i\,]$. Keeping that in mind, we divide matrix $\mathcal{P}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i}$ into $K$ matrix blocks:

$$\mathcal{P}[\,i\,] = \begin{bmatrix} \boldsymbol{P}_1[\,i\,] \\ \vdots \\ \boldsymbol{P}_k[\,i\,] \\ \vdots \\ \boldsymbol{P}_K[\,i\,]. \end{bmatrix},\ \text{with } \boldsymbol{P}_k[\,i\,] \in \mathbb{R}^{D\times D}. \tag{3.26}$$

Next, let us call $\boldsymbol{S}_k[\,i\,]$ the covariance matrix of the local estimator $\hat{\underset{\sim}{\boldsymbol{w}}}_k[\,i\,]$. Since $\mathcal{S}[\,i\,]$ is the covariance matrix of $\hat{\underset{\sim}{\mathrm{w}}}[\,i\,]$, and $\hat{\underset{\sim}{\mathrm{w}}}[\,i\,]$ is the collection of estimators as defined in (3.8), it follows that $\boldsymbol{S}_k[\,i\,]$ can be found by taking the $k$th diagonal block of $\mathcal{S}[\,i\,]$. It is shown in Appendix A that

$$\boldsymbol{S}_k[\,i\,] = \boldsymbol{P}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,] + \boldsymbol{Z}_k[\,i\,], \tag{3.27}$$

where $\boldsymbol{Z}_k[\,i\,]$ is the corresponding $k$th diagonal block of matrix $\mathcal{Z}[\,i\,]$ defined in Equation (3.23). Thus, from the definitions of $\boldsymbol{P}_k[\,i\,]$, $\boldsymbol{S}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$ as matrix blocks in Equation (3.26) and Equation (3.27), and from the distribution of $\hat{\underset{\sim}{\mathrm{w}}}$ in Equation (3.22), we can say that each local estimator $\hat{\underset{\sim}{\boldsymbol{w}}}_k$ is distributed, if $H_n$ is true, as

$$\hat{\underset{\sim}{\boldsymbol{w}}}_{n,k}[\,i\,] \sim \mathcal{N}(\boldsymbol{P}_k[\,i\,]\boldsymbol{\theta}_n + (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b},\, \boldsymbol{S}_k[\,i\,]). \tag{3.28}$$

Note that the distribution of $\hat{\underset{\sim}{\boldsymbol{w}}}_{n,k}[\,i\,]$ above depends on the active hypothesis (i.e., the state of nature at place) as the distribution of $\underset{\sim}{\boldsymbol{w}}_n$ in Equation (2.1) in Chapter 2. Therefore, we can build, for each node $k$, a hypothesis test using the local dLMS estimates, which are our *de facto* data sample. We can find this test by performing the same process described for the single node case in Chapter 2. The result is summarized in the following proposition.

**Proposition 5.** *A statistic $t_{n,k}[\,i\,]$ and the MAP hypothesis test to decide in favor of $H_n$ given the dLMS estimator in Equation (3.28) are*

$$\begin{gathered} t_{n,k}[\,i\,] = -\frac{1}{2}(\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,])^{\top}\boldsymbol{S}_k^{-1}[\,i\,](\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,]), \\ t_{n,k}[\,i\,] \underset{\varPhi_k \neq n}{\overset{\varPhi_k \neq m}{\gtrless}} t_{m,k}[\,i\,] + \gamma_{mn}, \end{gathered} \tag{3.29}$$

*where* $\boldsymbol{\epsilon}_{n,k}[\,i\,] = \mathrm{E}\big(\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}[\,i\,]\big)$ *and* $\gamma_{mn} = \ln(\mathbb{P}(H_m)/\mathbb{P}(H_n))$.

*Proof.* We start from the MAP test to compare a pair of hypotheses $\{H_n, H_m\}$ at node $k$ when $\hat{\boldsymbol{w}}_k[\,i\,]$ is our observed sample; i.e., we set for each local decision rule $\varPhi_k(\hat{\boldsymbol{w}}_k[\,i\,]) = n$ whenever

$$\mathbb{P}(H_n \mid \hat{\boldsymbol{w}}_k[\,i\,], \varUpsilon_k[\,i\,]) > \mathbb{P}(H_m \mid \hat{\boldsymbol{w}}_k[\,i\,], \varUpsilon_k[\,i\,]) \quad \forall m \neq n,$$

where $\varUpsilon_k[\,i\,]$ was defined in Equation (3.4). Using Bayes' theorem, we can compare using

$$f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,], H_n)\mathbb{P}(H_n) \underset{\varPhi_k \neq n}{\overset{\varPhi_k \neq m}{\gtrless}} f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,], H_m)\mathbb{P}(H_m), \tag{3.30}$$

where $f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,], H_n)$ is the p.d.f. of $\underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,]$ conditioned on $\{\varUpsilon_k[\,i\,], H_n\}$. From its distribution in Equation (3.28), this is

$$f(\hat{\boldsymbol{w}}_k[\,i\,] \mid \varUpsilon_k[\,i\,], H_n) = \Big((2\pi)^{\frac{i+1}{2}} \det^{\frac{1}{2}} \boldsymbol{S}_k[\,i\,]\Big)^{-1} \mathrm{e}^{t_{n,k}[\,i\,]}, \tag{3.31}$$

where $t_{n,k}[\,i\,]$ is a function of data given as

$$t_{n,k}[\,i\,] = -\frac{1}{2}(\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,])^{\top}\boldsymbol{S}_k^{-1}[\,i\,](\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,]), \tag{3.32}$$

where we define $\boldsymbol{\epsilon}_{n,k}[\,i\,] = \mathrm{E}\big(\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}[\,i\,]\big)$; thus, from Equation (3.28), we have

$$\boldsymbol{\epsilon}_{n,k}[\,i\,] = \boldsymbol{P}_k[\,i\,]\boldsymbol{\theta}_n + (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b}. \tag{3.33}$$

Hence, taking the natural logarithm on both sides in Equation (3.31) and canceling constants out, the test is reduced to that in Equation (3.29). □

In the next chapter, we will discuss the performance analysis of this detector and show that during a period of time that does not extend indefinitely the dLMS detector yields a performance that is independent of the step size and approximates an optimal detector, provided that its value is sufficiently small.

# 4 PERFORMANCE ANALYSIS

In this chapter, we will discuss the fact that, during a period of time that does not extend indefinitely, the dLMS detector developed in the previous chapter yields a performance that is independent of the step size, provided that its value is sufficiently small. Furthermore, it will be argued that as the step size is made smaller, and therefore the estimation algorithm at the core of the detection routine becomes slower, the performance of the proposed detector approximates an optimal Bayesian detector.

## 4.1 Criterion of performance

As a measure of performance, we consider the maximum probability of error

$$\xi_{\text{max}}[\,i\,] = \max\{\xi_1[\,i\,],\,\dots\,,\,\xi_k[\,i\,],\,\dots\,,\,\xi_K[\,i\,]\},$$

where $\xi_k[\,i\,]$ is the average probability of error at node $k$ as defined in Equation (2.7). According to Equation (3.29), an error occurs whenever $t_{n,k}[\,i\,] < t_{m,k}[\,i\,] + \gamma_{mn}$ for any $m$ when $H_n$ is true. Thus, the probability of an error when $H_n$ is true is

$$\begin{aligned}\xi_{n,k}[\,i\,] &= \mathbb{P}\left( \bigcup_{\substack{m=1\\m\neq n}}^{N} \{\underset{\sim}{t}_{n,k}[\,i\,] < \underset{\sim}{t}_{m,k}[\,i\,] + \gamma_{mn}\} \,\middle|\, H_n \right)\\ &= 1 - \mathbb{P}\left( \bigcap_{\substack{m=1\\m\neq n}}^{N} \{\underset{\sim}{t}_{m,k}[\,i\,] \leq \underset{\sim}{t}_{n,k}[\,i\,] - \gamma_{mn}\} \,\middle|\, H_n \right),\end{aligned} \tag{4.1}$$

where $\underset{\sim}{t}_{n,k}[\,i\,]$ is the random variable corresponding to $t_{n,k}[\,i\,]$.[1] Define $\zeta_{n,k}[\,i\,] = 1 - \xi_{n,k}[\,i\,]$ as the probability of a correct detection given $H_n$. Thus, the average probability of error at node $k$ is given as

$$\xi_k[\,i\,] = \sum_{n=1}^{N} \mathbb{P}(H_n)\xi_{n,k}[\,i\,] = 1 - \sum_{n=1}^{N} \mathbb{P}(H_n)\zeta_{n,k}[\,i\,], \tag{4.2}$$

where $\zeta_{n,k}[\,i\,] = 1 - \xi_{n,k}[\,i\,]$ is the probability of a correct detection given $H_n$. By the Law of Total Probability of continuous random variables, the probability of a correct detection given $H_n$ can be evaluated as

$$\zeta_{n,k}[\,i\,] = \int_{-\infty}^{\infty} \mathbb{P}\left( \bigcap_{\substack{m=1\\m\neq n}}^{N} \{\underset{\sim}{t}_{m,k}[\,i\,] \leq t - \gamma_{mn}\} \,\middle|\, \underset{\sim}{t}_{n,k}[\,i\,] = t,\, H_n \right) f_{\underset{\sim}{t}_{n,k}[\,i\,]|H_n}(t)\,\mathrm{d}t,$$

where $f_{\underset{\sim}{t}_{n,k}[\,i\,]|H_n}(t)$ denotes the marginal p.d.f. of $\underset{\sim}{t}_{n,k}[\,i\,]$ given $H_n$. But note that if we define

$$T_{mn,k}[\,i\,] = t_{m,k}[\,i\,] - t_{n,k}[\,i\,], \tag{4.3}$$

[1] As other random quantities in this thesis, for each state of nature $n$ there is a different random quantity corresponding to the various realizations $t_{m,k}[\,i\,]$, $\forall m, k$. As explained in subsection 1.4.7, in this case the dependence on a state $n$ is not denoted by an extra index in subscript, but by the conditioning symbol “|” followed by the corresponding active hypothesis $H_n$, as occurs in the expression of the probability of error given in Equation (4.1)

the probability of a correct detection can be expressed, from Equation (4.1), as

$$\zeta_{n,k}[\,i\,] = \mathbb{P}\left(\bigcap_{\substack{m=1\\ m\neq n}}^{N} \{\underset{\sim}{T}_{mn,k}[\,i\,] \le -\gamma_{mn}\} \,\middle|\, H_n\right), \tag{4.4}$$

where $\underset{\sim}{T}_{mn,k}[\,i\,]$ is the random variable corresponding to the realization $T_{mn,k}[\,i\,]$ and is conditioned on $H_n$. The expression in (4.4) can in turn be seen as a cumulative distribution function (c.d.f) of the random vector

$$\underset{\sim}{\boldsymbol{T}}_{n,k}[\,i\,] = \mathrm{col}\{\underset{\sim}{T}_{1n,k}[\,i\,],\, \dots\,,\, \underset{\sim}{T}_{(n-1)n,k}[\,i\,],\, \underset{\sim}{T}_{(n+1)n,k}[\,i\,],\, \dots\,,\, \underset{\sim}{T}_{Nn,k}[\,i\,]\}$$

evaluated at point $(-\gamma_{1n}\,,\, \dots\,,\, -\gamma_{Nn})$—it should be noted that all $\underset{\sim}{T}_{mn,k}[\,i\,]$ are conditioned on $H_n$. The following proposition states that we can find the distribution of the various $\underset{\sim}{T}_{mn,k}[\,i\,]$ given $H_n$ from the expressions of the $T_{mn,k}[\,i\,]$ in Equation (4.3).

**Proposition 6.** *The statistics $T_{mn,k}[\,i\,]$ in Equation (4.3) are the realization of normally distributed random variables and can be written as*

$$T_{mn,k}[\,i\,] = \left(\widehat{\boldsymbol{w}}_k[\,i\,] - \frac{\boldsymbol{\epsilon}_{m,k}[\,i\,] + \boldsymbol{\epsilon}_{n,k}[\,i\,]}{2}\right)^{\!\top} \boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n), \tag{4.5}$$

*where $\boldsymbol{\epsilon}_{n,k}[\,i\,] = \mathrm{E}\big(\underset{\sim}{\widehat{\boldsymbol{w}}}_{n,k}[\,i\,]\big)$ and $\boldsymbol{Q}_k[\,i\,] = \boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{P}_k[\,i\,]$.*

*Proof.* First, let us use the following result from Matrix Analysis: given a vector $\boldsymbol{x} \neq \boldsymbol{0} \in \mathbb{R}^D$ and a positive definite matrix $\boldsymbol{M} \in \mathbb{R}^{D\times D}$, we call the product $\boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x}$ a quadratic form, and for any two vectors $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^D$, it holds that

$$\boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x} - \boldsymbol{y}^\top\boldsymbol{M}\boldsymbol{y} = (\boldsymbol{x}+\boldsymbol{y})^\top\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y}).$$

Notice that $t_{n,k}[\,i\,]$ in Proposition 5 is in fact a quadratic form; therefore, $T_{mn,k}[\,i\,]$ as defined in Equation (4.3) is a difference of two quadratic forms:

$$\begin{aligned}
T_{mn,k}[\,i\,] &= t_{m,k}[\,i\,] - t_{n,k}[\,i\,]\\
&= -\frac{1}{2}(\widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{m,k}[\,i\,])^\top \boldsymbol{S}_k^{-1}[\,i\,](\widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{m,k}[\,i\,])\\
&\quad + \frac{1}{2}(\widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,])^\top \boldsymbol{S}_k^{-1}[\,i\,](\widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,])\\
&= \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x} - \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{M}\boldsymbol{y}\\
&= \frac{1}{2}(\boldsymbol{x}+\boldsymbol{y})^\top\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y}),
\end{aligned}$$

where we substituted $\boldsymbol{M} = \boldsymbol{S}_k^{-1}[\,i\,]$, $\boldsymbol{x} = \widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,]$ and $\boldsymbol{y} = \widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{m,k}[\,i\,]$. Therefore, we have

$$\begin{aligned}
T_{mn,k}[\,i\,] &= \tfrac{1}{2}(\boldsymbol{x}+\boldsymbol{y})^\top\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y})\\
&= \left(\widehat{\boldsymbol{w}}_k[\,i\,] - \frac{\boldsymbol{\epsilon}_{n,k}[\,i\,] + \boldsymbol{\epsilon}_{m,k}[\,i\,]}{2}\right)^{\!\top} \boldsymbol{S}_k^{-1}[\,i\,]\left(\boldsymbol{\epsilon}_{m,k}[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,]\right).
\end{aligned}$$

Note that, by the definition of $\boldsymbol{\epsilon}_{n,k}[\,i\,]$ as $\mathrm{E}\big(\underset{\sim}{\hat{\boldsymbol{w}}}_{n,k}[\,i\,]\big)$, we have from Equation (3.33) that

$$\begin{aligned}\boldsymbol{\epsilon}_{m,k}[\,i\,]-\boldsymbol{\epsilon}_{n,k}[\,i\,] &= \boldsymbol{P}_k[\,i\,]\boldsymbol{\theta}_m + (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b} - (\boldsymbol{P}_k[\,i\,]\boldsymbol{\theta}_n + (\boldsymbol{I}_D-\boldsymbol{P}_k[\,i\,])\boldsymbol{b})\\ &= \boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n).\end{aligned}\tag{4.6}$$

Thus, defining $\boldsymbol{Q}_k[\,i\,] = \boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{P}_k[\,i\,]$ results in the expression given in Proposition 6. Finally, since $\boldsymbol{x} = \hat{\boldsymbol{w}}_k[\,i\,]$ is a normally distributed vector (see Equation (3.28) ), $T_{mn,k}[\,i\,]$ is a linear combination of correlated normally distributed variables and, therefore, is also normally distributed. □

We can now obtain from Equation (4.5) expressions for its statistical parameters of $\underset{\sim}{T}_{mn,k}[\,i\,]$ necessary to calculate the probability in Equation (4.4). Those are given in the following proposition.

**Proposition 7.** *The expectation, variance and covariance of* $\underset{\sim}{T}_{mn,k}[\,i\,]$ *are given as*

$$\begin{aligned}\mathrm{E}\big(\underset{\sim}{T}_{mn,k}[\,i\,]\,|\,H_n\big) &= -\frac{1}{2}(\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n),\\ \mathrm{Var}\big(\underset{\sim}{T}_{mn,k}[\,i\,]\big) &= (\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n),\\ \mathrm{Cov}\big(\underset{\sim}{T}_{mn,k}[\,i\,],\underset{\sim}{T}_{\nu n,k}[\,i\,]\big) &= (\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_\nu-\boldsymbol{\theta}_n).\end{aligned}\tag{4.7}$$

*where* $\boldsymbol{R}_k[\,i\,] = \boldsymbol{P}_k^{\top}[\,i\,]\boldsymbol{Q}_k[\,i\,]$.

*Proof.* Finding the expectation is straightforward:

$$\begin{aligned}\mathrm{E}\big(\underset{\sim}{T}_{mn,k}[\,i\,]\,|\,H_n\big) &= \left(\boldsymbol{\epsilon}_{n,k}[\,i\,]-\frac{\boldsymbol{\epsilon}_{m,k}[\,i\,]+\boldsymbol{\epsilon}_{n,k}[\,i\,]}{2}\right)^{\top}\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)\\ &= \frac{1}{2}(\boldsymbol{\epsilon}_{n,k}[\,i\,]-\boldsymbol{\epsilon}_{m,k}[\,i\,])^{\top}\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)\\ &= \frac{1}{2}(\boldsymbol{\theta}_n-\boldsymbol{\theta}_m)^{\top}\boldsymbol{P}_k^{\top}[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)\\ &= -\frac{1}{2}(\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n),\end{aligned}$$

where we used the fact that $\boldsymbol{\epsilon}_{m,k}[\,i\,]-\boldsymbol{\epsilon}_{n,k}[\,i\,] = \boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)$ as shown in Equation (4.6). The variance can be easily found by applying some of its intrinsic properties: given a vector $\boldsymbol{y}$ and matrix $\boldsymbol{M}$, both deterministic, and a constant $c$, we have that

$$\mathrm{Var}(\underset{\sim}{\boldsymbol{x}}^{\top}\boldsymbol{M}\boldsymbol{y}+c) = \boldsymbol{y}^{\top}\boldsymbol{M}^{\top}\mathrm{Var}(\underset{\sim}{\boldsymbol{x}})\boldsymbol{M}\boldsymbol{y};$$

thus, removing the constant term in Equation (4.5), we have

$$\begin{aligned}\mathrm{Var}\big(\underset{\sim}{T}_{mn,k}[\,i\,]\big) &= \mathrm{Var}(\hat{\boldsymbol{w}}_k^{\top}[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n))\\ &= (\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{Q}_k^{\top}[\,i\,]\,\mathrm{Var}(\hat{\boldsymbol{w}}_k[\,i\,])\,\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)\\ &= (\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{P}_k^{\top}[\,i\,]\boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{S}_k[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)\\ &= (\boldsymbol{\theta}_m-\boldsymbol{\theta}_n)^{\top}\boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_m-\boldsymbol{\theta}_n).\end{aligned}$$

It can be proven that the covariance has a similar property:

$$\mathrm{Cov}(\underset{\sim}{\boldsymbol{x}}_1^\top \boldsymbol{M}_1 \boldsymbol{y}_1 + c_1,\, \underset{\sim}{\boldsymbol{x}}_2^\top \boldsymbol{M}_2 \boldsymbol{y}_2 + c_2) = \boldsymbol{y}_1^\top \boldsymbol{M}_1^\top \mathrm{Cov}(\underset{\sim}{\boldsymbol{x}}_1,\, \underset{\sim}{\boldsymbol{x}}_2) \boldsymbol{M}_2 \boldsymbol{y}_2;$$

thus, the covariance between $\underset{\sim}{T}_{mn,k}[\,i\,]$ and $\underset{\sim}{T}_{\nu n,k}[\,i\,]$ can be found similarly:

$$\begin{aligned}
\mathrm{Cov}\big(\underset{\sim}{T}_{mn,k}[\,i\,], \underset{\sim}{T}_{\nu n,k}[\,i\,]\big) &= \mathrm{Cov}(\hat{\boldsymbol{w}}_k^\top[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n), \hat{\boldsymbol{w}}_k^\top[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_\nu - \boldsymbol{\theta}_n)) \\
&= (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \boldsymbol{Q}_k^\top[\,i\,]\mathrm{Cov}(\hat{\boldsymbol{w}}_k[\,i\,], \hat{\boldsymbol{w}}_k[\,i\,])\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_\nu - \boldsymbol{\theta}_n) \\
&= (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \boldsymbol{P}_k^\top[\,i\,]\boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{S}_k[\,i\,]\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_\nu - \boldsymbol{\theta}_n) \\
&= (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_\nu - \boldsymbol{\theta}_n),
\end{aligned}$$

since $\mathrm{Cov}(\hat{\boldsymbol{w}}_k[\,i\,], \hat{\boldsymbol{w}}_k[\,i\,])$ must be equal to $\mathrm{Var}(\hat{\boldsymbol{w}}_k[\,i\,]) = \boldsymbol{S}_k[\,i\,]$ by definition. □

Since all statistical information we need in order to infer the probability in Equation (4.4) is given by the parameters in Equation (4.7), it is evident that the analysis of matrix $\boldsymbol{R}_k[\,i\,]$ is mandatory. Therefore, we need to describe how both matrices $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{S}_k[\,i\,]$ evolve with time in order to evaluate those quantities. We will do this analysis in the next section, assuming that the step size of the dLMS algorithm is sufficiently small.

## 4.2 The influence of the initial estimate

Before we move forward, there is yet another important property of the detector in Equation (4.5), which concerns the influence of the initial estimate. In order to illustrate this property, write the following alternative expression for $\hat{\boldsymbol{w}}_k[\,i\,]$ based on its distribution in Equation (3.28):

$$\hat{\boldsymbol{w}}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{d}_{0:i} + (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b}\,, \tag{4.8}$$

where we set $\hat{\boldsymbol{w}}_k[-1] = \boldsymbol{b}$, and $\boldsymbol{L}_k[\,i\,] \in \mathbb{R}^{D\times(i+1)K}$ is a *deterministic* matrix defined in such a way that $\boldsymbol{P}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{U}_{0:i}$. It is shown in Appendix B that $\boldsymbol{L}_k[\,i\,]$ is in fact well defined. Finally, note that choosing $\boldsymbol{b} = \boldsymbol{0}$ in Equation (4.8), we obtain $\hat{\boldsymbol{w}}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{d}_{0:i}$; therefore, we can write

$$\hat{\boldsymbol{w}}_k[\,i\,]\Big|_{\boldsymbol{b}} = \hat{\boldsymbol{w}}_k[\,i\,]\Big|_{\boldsymbol{0}} + (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b}, \tag{4.9}$$

where the vertical bar with a subscript "$|_{\boldsymbol{b}}$" indicates that $\hat{\boldsymbol{w}}_k[\,i\,]$ was initialized as $\hat{\boldsymbol{w}}_k[-1] = \boldsymbol{b}$. Now, also note from Equation (3.28) that

$$\frac{1}{2}(\boldsymbol{\epsilon}_{m,k}[\,i\,] + \boldsymbol{\epsilon}_{n,k}[\,i\,]) = (\boldsymbol{I}_D - \boldsymbol{P}_k[\,i\,])\boldsymbol{b} + \boldsymbol{P}_k[\,i\,]\bar{\boldsymbol{\theta}}_{mn},$$

where $\bar{\boldsymbol{\theta}}_{mn} = (\boldsymbol{\theta}_m + \boldsymbol{\theta}_n)/2$. We use this result and also Equation (4.9) to rewrite the statistic $T_{mn,k}[\,i\,]$ in Equation (4.5) as

$$T_{mn,k}[\,i\,] = (\hat{\boldsymbol{w}}_k[\,i\,]\Big|_{\boldsymbol{0}} - \boldsymbol{P}_k[\,i\,]\bar{\boldsymbol{\theta}}_{mn})^\top \boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n). \tag{4.10}$$

Therefore, equations Equation (4.5) and Equation (4.10) describe the same quantity. But notice that the estimate $\boldsymbol{w}_k[\,i\,]$ in Equation (4.5) is not restricted to any particular initial value $\boldsymbol{b}$; at the same time, $T_{mn,k}[\,i\,]$ in Equation (4.10) does not depend on the value $\boldsymbol{b}$. Therefore, we must conclude that $T_{mn,k}[\,i\,]$ does not depend on the value of $\boldsymbol{b}$; furthermore, because comparing

$$T_{mn,k}[\,i\,] \underset{\Phi_k \neq n}{\overset{\Phi_k \neq m}{\gtrless}} -\gamma_{mn}$$

is fundamentally the same as testing hypothesis $H_n$ against $H_m$ due to the definition of $T_{mn,k}[\,i\,]$ in Equation (4.3), the performance of this detector does not depend on the value of the initial estimate, as the optimal detector for the single node case in Chapter 2. Hence, one can choose any value $\boldsymbol{b}$ in order to initiate the dLMS algorithm and calculate the test statistic accordingly to Equation (4.5), and would obtain the same detection performance as if the algorithm were initialized at the zero vector, using the test statistic in Equation (4.10) instead. Although the choice $\boldsymbol{b} = \boldsymbol{0}$ seems the most natural choice for the initial estimate in the context of estimation problems, there is in fact a more convenient value for $\boldsymbol{b}$ for our detection context, as we will see later when we discuss the implementation of a feasible and low cost detector suitable for real world applications.

## 4.3 Analysis for small step sizes

As we discussed in the previous section, we must describe how matrices $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Q}_k[\,i\,]$ in Equation (4.10) evolve with time in order to describe the detector performance in terms of the probability of error in Equation (4.2). Let us start by analyzing the behavior of matrix $\boldsymbol{P}_k[\,i\,]$; recall from Equation (3.26) that $\boldsymbol{P}_k[\,i\,]$ is a block of $\mathcal{P}[\,i\,]$. Recall that while proofing Proposition 4, we saw that $\mathcal{P}[\,i\,]$ can be written as a sum given in Equation (3.20), which we repeat herein for future reference:

$$\mathcal{P}[\,i\,] = \sum_{j=0}^{i} \mathcal{F}[\,i, j+1\,]\mathcal{A}\mathcal{M}\mathcal{U}[\,j\,]\boldsymbol{U}[\,j\,], \tag{4.11}$$

(also recall the definitions of $\mathcal{F}[\,i,j\,]$, $\mathcal{A}$, $\mathcal{M}$, $\mathcal{U}[\,i\,]$ and $\boldsymbol{U}[\,i\,]$ in Equation (3.7), Equation (3.10) and Equation (3.13) ). For the general case, describing the evolution of $\mathcal{P}[\,i\,]$ with relation to time is a intractable mathematical problem. Notwithstanding, we will show later on that we can choose a value of the step size as small as we want without damaging performance. Thus, we proceed the analysis for the special case of small LMS step sizes.

Hereafter, we describe how to obtain an appropriate approximation of $\mathcal{P}[\,i\,]$ for small step sizes. First, it is convenient to define the local step sizes as $\mu_k = \mu\mu'_k$, with $\mu$, $\mu'_k \in \mathbb{R}^+$. Thus, we can substitute in Equation (3.20) $\mathcal{M} = \mu\mathcal{M}'$, where

$$\mathcal{M}' = \operatorname{diag}\{\mu'_1\boldsymbol{I}_D, \dots, \mu'_k\boldsymbol{I}_D, \dots, \mu'_K\boldsymbol{I}_D\}. \tag{4.12}$$

Thereby, $\mathcal{P}[\,i\,]$ in Equation (3.20) can be expanded as a polynomial on $\mu$, resulting in

$$\mathcal{P}[\,i\,] = \mathcal{B}_0[\,i\,] + \mu\mathcal{B}_1[\,i\,] + \cdots + \mu^{i+1}\mathcal{B}_{i+1}[\,i\,], \tag{4.13}$$

where the matrices $\mathcal{B}_\kappa[j]$ are the coefficients of the expansion and do not depend on $\mu$. The first three coefficients are given as (see Appendix C)

$$\begin{aligned}
\mathcal{B}_0[\,i\,] &= \mathbf{0},\\
\mathcal{B}_1[\,i\,] &= \sum_{j=0}^{i} \mathcal{A}^{i-j+1}\boldsymbol{H}[\,j\,],\\
\mathcal{B}_2[\,i\,] &= -\sum_{j=0}^{i}\left(\sum_{\iota=j+1}^{i} \mathcal{A}^{i-\iota+1}\mathcal{H}[\,\iota\,]\mathcal{A}^{\iota-j}\right)\boldsymbol{H}[\,j\,],
\end{aligned} \tag{4.14}$$

where we define

$$\begin{aligned}
\boldsymbol{H}[\,j\,] &= \mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{U}[\,j\,], \text{ and}\\
\mathcal{H}[\,\iota\,] &= \mathcal{M}'\mathcal{U}[\,\iota\,]\,\mathcal{U}^{\mathsf{T}}[\,\iota\,].
\end{aligned} \tag{4.15}$$

It is important to ask for what conditions it is reasonable to approximate $\mathcal{P}[\,i\,]$ as a first-order polynomial on $\mu$. That is, since $\mathcal{B}_0[\,i\,] = \mathbf{0}$,

$$\mathcal{P}[\,i\,] \approx \mu\mathcal{B}_1[\,i\,].$$

As we said in Chapter 1, we are interested in an algorithm that decides the current state within a *finite* time interval; thus, let us limit our analysis for $i \le I_{\mathcal{B}_1}$, where $I_{\mathcal{B}_1}$ denotes the time limit in which this approximation is to be valid. Under this condition, notice that $\mathcal{B}_2[\,i\,]$ is upper bounded: the summations in Equation (4.14) have at most $i \le I_{\mathcal{B}_1}$ terms, and matrices $\mathcal{H}[\,i\,]$ and $\boldsymbol{H}[\,i\,]$ always have finite entries by their definition in Equation (4.15). In the case of matrix $\mathcal{A}$, it follows from its definition in Equation (3.7) that $\mathcal{A}^i$ always converges as $i$ increases according to the *Perron-Frobenius theorem* [101]—we will discuss this more carefully ahead in Section 4.4. At the same time, $\mathcal{B}_1[\,i\,]$ is always different from the zero matrix if any of the regressors $\boldsymbol{u}_k[\,i\,] \neq \mathbf{0}$ for $i \geq 0$. Therefore, we can choose a value of the step size $\mu \le \mu_{\mathcal{B}_1}$ sufficiently small so that, for $i \le I_{\mathcal{B}_1}$, the second order term $\mu^2\mathcal{B}_2[\,i\,]$ is negligible in comparison to the first order term $\mu\mathcal{B}_1[\,i\,]$. This argument is valid for all terms $\mu^\kappa\mathcal{B}_\kappa[\,i\,]$, $\kappa \geq 2$ by the same reasons. Thus, from Equation (4.14), we can approximate $\mathcal{P}[\,i\,]$ around $\mu = 0$ as

$$\mathcal{P}[\,i\,] \approx \mu\sum_{j=0}^{i} \mathcal{A}^{i-j+1}\boldsymbol{H}[\,i\,], \tag{4.16}$$

provided that, for a given interval $i \le I_{\mathcal{B}_1}$, we guarantee that the higher order terms in Equation (4.13) are negligible for all $\mu \le \mu_{\mathcal{B}_1}$; in other words, this approximation is valid for a period of time lasting until time instant $I_{\mathcal{B}_1}$. It is very important to note that if we want this approximation to be valid for a larger interval, which means a larger value of $I_{\mathcal{B}_1}$, we only need, at least in principle, to choose a smaller value of the step size $\mu$ such that the higher order terms in Equation (4.13) take more time to really affect the behavior of $\mathcal{P}[\,i\,]$ and, consequently, the detector performance. As we will show later in this chapter, we can always choose smaller values for $\mu$ with no degradation in performance.

Finally, we are able to find an approximation for $\boldsymbol{P}_k[\,i\,]$. First, we need the following proposition:

**Proposition 8.** *For matrix $\mathcal{A} = \boldsymbol{A}^\top \otimes \boldsymbol{I}_D$, it holds that*

$$\mathcal{A}^i = \left(\boldsymbol{A}^\top\right)^i \otimes \boldsymbol{I}_D,\ i \geq 0. \tag{4.17}$$

*Proof.* For $i = 0$ and $i = 1$, the proof is trivial. For $i \geq 2$, we use the property of Kronecker products that, for any matrices $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$ with appropriate dimensions, it holds $(\boldsymbol{M}_1 \otimes \boldsymbol{M}_2) \cdot (\boldsymbol{M}_3 \otimes \boldsymbol{M}_4) = (\boldsymbol{M}_1\boldsymbol{M}_3) \otimes (\boldsymbol{M}_2\boldsymbol{M}_4)$. Therefore, for $i = 2$, we have

$$\begin{aligned} \mathcal{A}^2 &= \mathcal{A} \cdot \mathcal{A} = (\boldsymbol{A}^\top \otimes \boldsymbol{I}_D) \cdot (\boldsymbol{A}^\top \otimes \boldsymbol{I}_D) \\ &= (\boldsymbol{A}^\top\boldsymbol{A}^\top) \otimes (\boldsymbol{I}_D\boldsymbol{I}_D) = \left(\boldsymbol{A}^\top\right)^2 \otimes \boldsymbol{I}_D. \end{aligned}$$

Suppose that $\mathcal{A}^{i-1} = \left(\boldsymbol{A}^\top\right)^{i-1} \otimes \boldsymbol{I}_D$ is a true statement. Thus,

$$\begin{aligned} \mathcal{A}^i &= \mathcal{A}^{i-1}\mathcal{A} \\ &= \left(\left(\boldsymbol{A}^\top\right)^{i-1} \otimes \boldsymbol{I}_D\right) \cdot (\boldsymbol{A}^\top \otimes \boldsymbol{I}_D) \\ &= \left(\left(\boldsymbol{A}^\top\right)^{i-1}\boldsymbol{A}^\top\right) \otimes (\boldsymbol{I}_D\boldsymbol{I}_D) \\ &= \left(\boldsymbol{A}^\top\right)^i \otimes \boldsymbol{I}_D. \end{aligned}$$

Therefore, by a mathematical induction process, the proposition is proved for $i \geq 2$. □

Next, using Proposition 8, and seeing $\mathcal{P}[\,i\,]$ divided into block matrices $\boldsymbol{P}_k[\,i\,]$ in Equation (3.26), we can prove the following:

**Proposition 9.** *A first-order polynomial approximation of matrices $\boldsymbol{P}_k[\,i\,]$, $k = 1, \dots, K$, defined in Equation (3.26), is given as*

$$\boldsymbol{P}_k[\,i\,] \approx \mu \sum_{j=0}^{i} \sum_{\ell=1}^{K} \left(\boldsymbol{A}^{i-j+1}\right)_{\ell k} \mu'_\ell \boldsymbol{u}_\ell[\,j\,]\boldsymbol{u}_\ell^\top[\,j\,], \tag{4.18}$$

*where $\left(\boldsymbol{A}^{i-j+1}\right)_{\ell k}$ denotes the entry of matrix $\boldsymbol{A}^{i-j+1}$ at the $\ell$h row and kth column. This approximation if valid for $i \leq I_{\mathcal{B}_1}$ for any $\mu \leq \mu_{\mathcal{B}_1}$ such that the higher order terms $\mu^\kappa\mathcal{B}_\kappa[\,i\,]$, $\kappa \geq 2$ in Equation (4.13) are negligible.*

*Proof.* See Appendix D. □

Very similarly and under the same conditions, we can find a suitable approximation to $\boldsymbol{S}_k[\,i\,]$ defined in Equation (3.27). Substituting $\mathcal{L}[\,i\,]$ and $\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}$ by its definitions given in

Equation (3.15) and Equation (3.24), respectively, in the expression of $\mathcal{Z}[\,i\,]$ in Equation (3.25) and performing the resultant matrix product, we can find a new expression of $\mathcal{Z}[\,i\,]$ given as

$$
\begin{aligned}
\mathcal{Z}[\,i\,] &= \mathcal{L}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}^{\top}[\,i\,] \\
\mathcal{L}[\,i\,] &= \Big[\mathcal{AMU}[\,i\,] \ \ \ldots \ \ \mathcal{F}[\,i,1\,]\mathcal{AMU}[\,0\,]\Big]\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{v}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{v}}\end{bmatrix}\begin{bmatrix}(\mathcal{AMU}[\,i\,])^{\top} \\ \vdots \\ (\mathcal{F}[\,i,1\,]\mathcal{AMU}[\,0\,])^{\top}\end{bmatrix} \\
&= \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\mathcal{D}[\,j\,]\mathcal{A}^{\top}\mathcal{F}^{\top}[\,i,j+1\,],
\end{aligned}
\tag{4.19}
$$

where we define $\mathcal{D}[\,j\,] = \mathcal{MU}[\,j\,]\boldsymbol{\Sigma}_{\boldsymbol{v}}\mathcal{U}^{\top}[\,j\,]\mathcal{M}^{\top}$. Expanding $\mathcal{Z}[\,i\,]$ as a polynomial on $\mu$ as in Equation (4.13), let $\mathcal{C}_{\kappa}[\,i\,]$, $\kappa \in \mathbb{N}$, be the matrix coefficients of such an expansion; i.e.,

$$
\mathcal{Z}[\,i\,] = \mathcal{C}_0[\,i\,] + \mu\mathcal{C}_1[\,i\,] + \cdots + \mu^{i+1}\mathcal{C}_{i+1}[\,i\,].
\tag{4.20}
$$

The first three coefficients are given as (see Appendix C)

$$
\begin{aligned}
\mathcal{C}_0[\,i\,] &= \mathbf{0}, \\
\mathcal{C}_1[\,i\,] &= \mathbf{0}, \\
\mathcal{C}_2[\,i\,] &= \sum_{j=0}^{i}\mathcal{A}^{i-j+1}\mathcal{M}'\,\mathcal{U}[\,j\,]\boldsymbol{\Sigma}_{v}\,\mathcal{U}^{\top}[\,j\,]\mathcal{M}'\,(\mathcal{A}^{\top})^{i-j+1}.
\end{aligned}
$$

Thus, following the same reasoning used for $\mathcal{P}[\,i\,]$, given a $\mu \leq \mu_{\mathcal{C}_2}$ such that all $\mathcal{C}_{\kappa}[\,i\,]$, $\kappa \geq 3$, are negligible for $i \leq I_{\mathcal{C}_2}$, we can approximate[2]

$$
\mathcal{Z}[\,i\,] \approx \mu^2\sum_{j=0}^{i}\mathcal{A}^{i-j+1}\mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{\Sigma}_{\boldsymbol{v}}\,\mathcal{U}^{\top}[\,j\,]\mathcal{M}'(\mathcal{A}^{\top})^{i-j+1}.
$$

Recalling that $\boldsymbol{Z}_k[\,i\,]$ is the $k$th $D \times D$ diagonal block of $\mathcal{Z}[\,i\,]$, we can prove the following proposition:

**Proposition 10.** *A second-order polynomial approximation of matrices $\boldsymbol{Z}_k[\,i\,]$, $k = 1, \ldots, K$, defined as the kth $D \times D$ diagonal block of $\mathcal{Z}[\,i\,]$, is given as*

$$
\boldsymbol{Z}_k[\,i\,] \approx \mu^2\sum_{j=0}^{i}\sum_{\ell=1}^{K}(\boldsymbol{A}^{i-j+1})^2_{\ell k}\,\mu_{\ell}'^{2}\sigma^2_{v_k}\boldsymbol{u}_{\ell}[\,j\,]\boldsymbol{u}_{\ell}^{\top}[\,j\,].
\tag{4.21}
$$

*where $(\boldsymbol{A}^{i-j+1})_{\ell k}$ denotes the entry of matrix $\boldsymbol{A}^{i-j+1}$ at the $\ell$h row and kth column. This approximation if valid for $i \leq I_{\mathcal{C}_2}$ for any $\mu \leq \mu_{\mathcal{C}_2}$ such that the higher order terms $\mu^{\kappa}\mathcal{C}_{\kappa}[\,i\,]$, $\kappa \geq 3$ in Equation (4.20) are negligible.*

*Proof.* See Appendix D. □

[2] The attentive reader may question why $\mathcal{Z}[\,i\,]$ is approximated by a second order polynomial on $\mu$, whereas $\mathcal{P}[\,i\,]$ is approximated by a first order polynomial instead. This is so because $\mathcal{P}[\,i\,]$ relates the estimators $\underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,]$ with its expected values, as we can see in Equation (3.22), whereas $\mathcal{Z}[\,i\,]$ is related to their variances by Equation (3.23).

Now, we choose the value of $\mu$ such that

$$\mu < \mu_{\text{sup}} = \min\{\mu_{\mathcal{B}_1},\, \mu_{\mathcal{C}_2}\},$$

in order to guarantee that our approximations for $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{S}_k[\,i\,]$ are valid for any

$$i < I_{\text{sup}} = \min\{I_{\mathcal{B}_1}, I_{\mathcal{C}_2}\}.$$

Under such conditions, define the matrices $\widehat{\boldsymbol{P}}_k[\,i\,]$ and $\widehat{\boldsymbol{Z}}_k[\,i\,]$ such that $\boldsymbol{P}_k[\,i\,] = \mu\widehat{\boldsymbol{P}}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,] = \mu^2\widehat{\boldsymbol{Z}}_k[\,i\,]$. Note that, by definition, $\widehat{\boldsymbol{P}}_k[\,i\,]$ and $\widehat{\boldsymbol{Z}}_k[\,i\,]$ do not depend on $\mu$. Applying the approximations for $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$ obtained above, and the definition of $\boldsymbol{S}_k[\,i\,]$ in Equation (3.27), we have

$$\begin{aligned}\boldsymbol{R}_k[\,i\,] &\approx \mu\widehat{\boldsymbol{P}}_k^{\top}[\,i\,]\Big(\mu\widehat{\boldsymbol{P}}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mu\widehat{\boldsymbol{P}}_k^{\top}[\,i\,] + \mu^2\widehat{\boldsymbol{Z}}_k[\,i\,]\Big)^{-1}\mu\widehat{\boldsymbol{P}}_k[\,i\,] \\ &= \widehat{\boldsymbol{P}}_k^{\top}[\,i\,]\Big(\widehat{\boldsymbol{P}}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\widehat{\boldsymbol{P}}_k^{\top}[\,i\,] + \widehat{\boldsymbol{Z}}_k[\,i\,]\Big)^{-1}\widehat{\boldsymbol{P}}_k[\,i\,],\end{aligned} \tag{4.22}$$

where we can see that the resulting matrix product does not depend on $\mu$; therefore, the parameters of $\underset{\sim}{T}_{mn,k}[\,i\,]$ in Equation (4.7) do not depend on $\mu$ as well, provided $\mu < \mu_{\text{sup}}$ whenever $i < I_{\text{sup}}$. Hence, for sufficient small step sizes and for a period of time, the performance of the detector does not depend on the value of this step size. This results in the fact that its performance is not affected by smaller choices of values for $\mu$ in terms of convergence rate, since smaller values of $\mu$ extend the period during which our approximations are valid. That means that desirable values for the convergence rate and steady state performance of the detector are not conflicting, in contrast to what happens in the estimation problem using the LMS algorithm. In fact, the dLMS at the core of the detector is prevented from reaching the steady state, which in turn reflects in the matrices $\boldsymbol{R}_k[\,i\,]$ being independent of the step size.

This counterintuitive result can be more easily explained for $N = 2$ hypotheses in the small step size regime: as shown in the predecessor works [60] and [88], the detector performance in that case depends on the ratio of the expected value of the detector statistic (which grows with $\mu$) and the square root of the variance (which also grows with $\mu$); more specifically, the probability of error at node $k$ is given as

$$\xi_k[\,i\,] = \mathbb{P}(H_1)\mathcal{Q}\left(\frac{\mu\gamma - \mathrm{E}(t_k[\,i\,] \mid H_1)}{\sigma_k[\,i\,]}\right) + \mathbb{P}(H_2)\mathcal{Q}\left(\frac{\mathrm{E}(t_k[\,i\,] \mid H_2) - \mu\gamma}{\sigma_k[\,i\,]}\right),$$

where $t_k[\,i\,]$ represents the $N = 2$ test statistic, $\sigma_k[\,i\,] = \sqrt{\mathrm{Var}(t_k[\,i\,])}$, $\gamma$ is the proper decision threshold and $\mathcal{Q}(z) = \mathbb{P}(Z > z)$ for a standard normally distributed variable $Z$. As we can see, the step size $\mu$ cancels out in the expression above. Unfortunately, such easy visualization by a simple expression is not possible for $N \geq 3$.

## 4.4 Analysis as smaller step sizes are chosen

Let us proceed our analysis in order to show how the proposed detector approximates the minimum possible probability of error, which is the performance achieved by optimal ideal

Bayesian detector. We do it by showing, under a few assumptions we discuss next, that the local test statistics $T_{n,k}[\,i\,]$ in Equation (4.10) approximates an ideal global optimal test statistic. We start by discussing how the coefficients $(\boldsymbol{A}^i)_{\ell k}$ can be approximated; recall that $\boldsymbol{A}$, as defined in Section 3.2, is a *left stochastic* matrix. Thus, it follows from the *Perron-Frobenius theorem* [101] that $\boldsymbol{A}$ has a single (right) eigenvalue at one and its spectral radius is also one[3]—i.e., $\rho(\boldsymbol{A}) = 1$—and there is a vector $\boldsymbol{\pi} \in \mathbb{R}^K$ with a sequence of entries $\{\pi_k\}_{k=1}^K$, $\pi_k > 0$, such that

$$\boldsymbol{A}\boldsymbol{\pi} = \boldsymbol{\pi}, \quad \mathbb{1}_K^\top \boldsymbol{\pi} = 1.$$

In other words, $\boldsymbol{\pi}$ is the (right) eigenvector of $\boldsymbol{A}$ associated with the unitary eigenvalue. It follows again from the Perron-Frobenius theorem that, for a sufficiently large $i > I_{\boldsymbol{\pi}}$,

$$\boldsymbol{A}^i \approx \boldsymbol{\pi}\,\mathbb{1}_K^\top = [\,\pi_1 \;\; \pi_2 \;\; \ldots \;\; \pi_K\,];$$

that is, as $i$ increases, the coefficients $(\boldsymbol{A}^i)_{\ell k}$ can be approximated by $\pi_\ell$ for each $\ell \in \{1, \ldots, K\}$. Let us see how this is the case in Equation (4.18); after changing $i - j = \iota$ for simplicity, we can approximate the inner summation as

$$\sum_{\iota=0}^{i} (\boldsymbol{A}^{\iota+1})_{\ell k} \boldsymbol{u}_\ell[\,i-\iota\,]\boldsymbol{u}_\ell^\top[\,i-\iota\,] \approx \sum_{\iota=0}^{i} \pi_\ell\, \boldsymbol{u}_\ell[\,i-\iota\,]\boldsymbol{u}_\ell^\top[\,i-\iota\,]. \tag{4.23}$$

Note that as $i$ increases more terms $(\boldsymbol{A}^{\iota+1})_{\ell k}$ in the summation can be better approximated by $\pi_\ell$, since $\boldsymbol{A}^{\iota+1}$ becomes closer to $\boldsymbol{\pi}\,\mathbb{1}_K^\top$. Also note that this approximation is equivalent to assuming that the network is totally connected with a combination matrix $\boldsymbol{A} = \boldsymbol{\pi}\,\mathbb{1}_K^\top$. [4] The reader should be convinced that this is a reasonable argument, since due to the diffusion process among nodes in a connected network, every node eventually receives information from every other node as time increases. Therefore, according to this reasoning, this approximation is better when the network has a large number of connections among nodes, as information diffuses faster. Finally, all the reasoning about approximating $(\boldsymbol{A}^i)_{\ell k}$ by $\pi_\ell$ also applies to $\boldsymbol{Z}_k[\,i\,]$ in Equation (4.21).

Let us move forward and apply these results in order to find adequate approximations for $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$. In addition to the criterion above about a sufficiently large $i > I_{\boldsymbol{\pi}}$, also consider that we choose a $\mu < \mu_{\text{sup}}$ to approximate $\boldsymbol{P}_k[\,i\,]$ in Equation (4.18), valid for $i < I_{\text{sup}}$. Therefore, during a time interval such that $I_{\boldsymbol{\pi}} < i < I_{\text{sup}}$, $\boldsymbol{P}_k[\,i\,]$ can be approximated as follows[5]:

$$\boldsymbol{P}_k[\,i\,] \approx \mu \sum_{j=0}^{i} \sum_{\ell=1}^{K} \mu'_\ell\, \pi_\ell\, \boldsymbol{u}_\ell[\,j\,]\, \boldsymbol{u}_\ell^\top[\,j\,], \tag{4.24}$$

Under the same conditions, each matrix block $\boldsymbol{Z}_k[\,i\,]$ of $\mathcal{Z}[\,i\,]$ can also be approximated as

$$\boldsymbol{Z}_k[\,i\,] \approx \mu^2 \sum_{j=0}^{i} \sum_{\ell=1}^{K} \mu'^2_\ell \sigma^2_{v_k} \pi^2_\ell\, \boldsymbol{u}_\ell[\,j\,]\, \boldsymbol{u}_\ell^\top[\,j\,]. \tag{4.25}$$

[3] The spectral radius of a square matrix $\boldsymbol{M} \in \mathbb{R}^{D\times D}$ is the maximum of the absolute values of its eigenvalues; i.e., $\rho(\boldsymbol{M}) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \boldsymbol{M}\}$.

[4] By the Perron-Frobenius theorem, the vector $\boldsymbol{\pi}$ has only strictly positive entries. Therefore, the same applies to the combination matrix $\boldsymbol{A} = \boldsymbol{\pi}\,\mathbb{1}_K^\top$, and thus the network it describes has all nodes connected to all others.

[5] As discussed in Section 4.3, we can guarantee, in principle, that $I_{\boldsymbol{\pi}} < I_{\text{sup}}$ by choosing a sufficiently small value of $\mu$ to assure a sufficiently large $I_{\text{sup}}$.

Now, suppose that we are able to set, for all $k$,

$$\mu_k' = \left(\pi_k\, \sigma_{v_k}^2\right)^{-1}, \tag{4.26}$$

which is possible assuming that the noise variances $\sigma_{v_k}^2$ at nodes are known or can be satisfactorily estimated. Therefore, we have from Equation (4.24) and Equation (4.25) that

$$\begin{aligned}\boldsymbol{P}_k[\,i\,] &\approx \mu \sum_{j=0}^{i}\sum_{\ell=1}^{K} \frac{\boldsymbol{u}_\ell[\,j\,]\,\boldsymbol{u}_\ell^\top[\,j\,]}{\sigma_{v_\ell}^2},\\ \boldsymbol{Z}_k[\,i\,] &\approx \mu^2 \sum_{j=0}^{i}\sum_{\ell=1}^{K} \frac{\boldsymbol{u}_\ell[\,j\,]\,\boldsymbol{u}_\ell^\top[\,j\,]}{\sigma_{v_\ell}^2}.\end{aligned} \tag{4.27}$$

Also, examining closely, we can notice that

$$\sum_{j=0}^{i}\sum_{\ell=1}^{K} \frac{\boldsymbol{u}_\ell[\,j\,]\,\boldsymbol{u}_\ell^\top[\,j\,]}{\sigma_{v_\ell}^2} = \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i}; \tag{4.28}$$

thus, we have in Equation (4.27) that

$$\boldsymbol{P}_k[\,i\,] \approx \mu\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i}, \text{ and } \boldsymbol{Z}_k[\,i\,] \approx \mu^2\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i}. \tag{4.29}$$

Now, under the conditions assumed so far, we apply the approximations of $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{S}_k[\,i\,]$ in the expression of the local estimate $\hat{\boldsymbol{w}}_k[\,i\,]$. First, recall that we saw in Section 4.2 that the detector performance does not depend on the initial estimate; thus, for simplicity, we assume $\boldsymbol{b} = \boldsymbol{0}$. Hence, from Equation (2.3) and Equation (4.8), we can write

$$\hat{\boldsymbol{w}}_k[\,i\,]\big|_{\boldsymbol{0}} = \boldsymbol{P}_k[\,i\,]\boldsymbol{w}_n + \boldsymbol{z}_k[\,i\,], \tag{4.30}$$

where $\boldsymbol{z}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{v}_{0:i}$ and $\boldsymbol{w}_k[-1] = \boldsymbol{0}$. In order to find an approximation for $\boldsymbol{z}_k[\,i\,]$ under the conditions we are assuming so far, since $\boldsymbol{P}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{U}_{0:i}$ and from the approximation of $\boldsymbol{P}_k[\,i\,]$ above, we conclude that matrix $\boldsymbol{L}_k[\,i\,]$ can be approximate as

$$\boldsymbol{L}_k[\,i\,] \approx \mu\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}$$

and, therefore,

$$\boldsymbol{z}_k[\,i\,] \approx \mu\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{v}_{0:i}.$$

To show that this is indeed an adequate approximation, we use the fact that

$$\boldsymbol{Z}_k[\,i\,] = \mathrm{E}\big(\underset{\sim}{\boldsymbol{z}}_k[\,i\,]\underset{\sim}{\boldsymbol{z}}_k^\top[\,i\,]\big) \tag{4.31}$$

(see proof in Appendix A). Thus,

$$\begin{aligned}\mathrm{E}\big(\underset{\sim}{\boldsymbol{z}}_k[\,i\,]\underset{\sim}{\boldsymbol{z}}_k^\top[\,i\,]\big) &\approx \mathrm{E}\big(\mu \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \underset{\sim}{\boldsymbol{v}}_{0:i}\big)\big(\mu \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \underset{\sim}{\boldsymbol{v}}_{0:i}\big)^\top\big)\\ &= \mu^2 \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \mathrm{E}\big(\underset{\sim}{\boldsymbol{v}}_{0:i}\underset{\sim}{\boldsymbol{v}}_{0:i}^\top\big) \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i}\\ &= \mu^2 \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}} \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i}\\ &= \mu^2 \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1} \boldsymbol{U}_{0:i},\end{aligned}$$

which is precisely, from Equation (4.28), the approximation for $\boldsymbol{Z}_k[\,i\,]$. Thus, from the approximations for $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{z}_k[\,i\,]$ above, we have the following approximation for $\hat{\boldsymbol{w}}_k[\,i\,]$ in Equation (4.30):

$$\hat{\boldsymbol{w}}_k[\,i\,]\big|_{\boldsymbol{0}} \approx \mu\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}(\boldsymbol{U}_{0:i}\boldsymbol{w}_n + \boldsymbol{v}_{0:i}) = \mu\, \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{d}_{0:i}. \tag{4.32}$$

Finally, we can approximate $T_{mn,k}[\,i\,]$ in Equation (4.10) substituting the quantities $\hat{\boldsymbol{w}}_k[\,i\,]$, $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$ by their approximations, which is presented in the following proposition:

**Proposition 11.** *An approximation for $T_{mn,k}[\,i\,]$ for $\mu < \mu_{sup}$ and $I_{\boldsymbol{\pi}} < i < i_{sup}$, and if $\mu_k = \left(\pi_k\, \sigma_{v_k}^2\right)^{-1}\mu$ is given as*

$$T_{mn,k}[\,i\,] \approx (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\bar{\boldsymbol{\theta}}_{nm})^\top \boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}\boldsymbol{U}_{0:i}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n). \tag{4.33}$$

*Proof.* We will use the approximations of $\hat{\boldsymbol{w}}_k[\,i\,]\big|_{\boldsymbol{0}}$, $\boldsymbol{P}_k[\,i\,]$, $\boldsymbol{Z}_k[\,i\,]$ in Equation (4.32) and Equation (4.29) to substitute them in the expression of $T_{mn,k}[\,i\,]$ in Equation (4.10). But first, let us recall the definitions of $\boldsymbol{Q}_k[\,i\,]$ (and also of $\boldsymbol{S}_k[\,i\,]$ in Equation (3.27)) and rewrite it as

$$\begin{aligned}\boldsymbol{Q}_k[\,i\,] &= \boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{P}_k[\,i\,]\\ &= \left(\boldsymbol{P}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^\top[\,i\,] + \boldsymbol{Z}_k[\,i\,]\right)^{-1}\boldsymbol{P}_k[\,i\,]\\ &= \left(\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^\top[\,i\,] + \boldsymbol{P}_k^{-1}[\,i\,]\boldsymbol{Z}_k[\,i\,]\right)^{-1}.\end{aligned}$$

Note that using the approximations of $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$ in Equation (4.29) we have that

$$\boldsymbol{P}_k^{-1}[\,i\,]\boldsymbol{Z}_k[\,i\,] \approx \mu \boldsymbol{I}_M;$$

therefore, we can approximate of $\boldsymbol{Q}_k[\,i\,]$ as

$$\begin{aligned}\boldsymbol{Q}_k[\,i\,] &\approx \left(\mu\, \boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i} + \mu \boldsymbol{I}_M\right)^{-1}\\ &= \frac{1}{\mu}\left(\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i} + \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}.\end{aligned} \tag{4.34}$$

Thus, we have

$$\begin{aligned}T_{mn,k}[\,i\,] &= (\hat{\boldsymbol{w}}_k[\,i\,]\big|_{\boldsymbol{0}} - \boldsymbol{P}_k[\,i\,]\bar{\boldsymbol{\theta}}_{mn})^\top \boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)\\ &\approx \left(\mu \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{d}_{0:i} - \mu \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\bar{\boldsymbol{\theta}}_{mn}\right)^\top\\ &\quad \times \frac{1}{\mu}\left(\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i} + \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)\\ &= (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\bar{\boldsymbol{\theta}}_{mn})^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i} + \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n).\end{aligned} \tag{4.35}$$

Now we need to manipulate the matrix product above as follows:

$$\begin{aligned}
&\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \qquad (4.36)\\
&=\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}+\underbrace{\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}-\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}}_{\mathbf{0}}\right)\\
&=\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\underbrace{\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}+\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\right)}_{\boldsymbol{I}_{i+1}}\\
&\quad-\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\\
&=\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}-\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\\
&=\left(\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}-\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\right)\boldsymbol{U}_{0:i}.
\end{aligned}$$

By the Matrix Inversion Lemma[6], it holds that

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}-\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}=\left(\boldsymbol{U}_{0:i}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{U}_{0:i}^{\top}+\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\right)^{-1}=\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1},$$

since $\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}=\boldsymbol{U}_{0:i}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{U}_{0:i}^{\top}+\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}$ (see Equation (2.10) ). Thus, we can write (4.36) as

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}\left(\boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}+\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}=\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}\boldsymbol{U}_{0:i},$$

and rewrite Equation (4.35) as given in Equation (4.33). □

In order to make sense of this result, let us define an optimal statistic test direct from the global linear model $\boldsymbol{d}_{0:i}=\boldsymbol{U}_{0:i}\boldsymbol{w}_n+\boldsymbol{v}_{0:i}$. Consequently, the corresponding statistic must encompass all data of all nodes and at all times; thus, it would be equivalent to the statistic of a single omniscient sensor node where all data are received. Therefore, following a very similar procedure described in Section 2.2 for the single node scenario (the only different is that $\boldsymbol{d}_{0:i}$ collects data of $K>1$ nodes), the optimal global statistic to test hypothesis $H_n$ results to be given just as presented in Proposition 1; i.e.,

$$o_n[\,i\,]=-\frac{1}{2}(\boldsymbol{d}_{0:i}-\boldsymbol{U}_{0:i}\boldsymbol{\theta}_n)^{\top}\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}(\boldsymbol{d}_{0:i}-\boldsymbol{U}_{0:i}\boldsymbol{\theta}_n). \qquad (4.37)$$

Define $O_{mn}[\,i\,]=o_m[\,i\,]-o_n[\,i\,]$, which can be reduced, using the fact that this is a difference of two quadratic forms, to[7]

$$O_{mn}[\,i\,]=(\boldsymbol{d}_{0:i}-\boldsymbol{U}_{0:i}\bar{\boldsymbol{\theta}}_{mn})^{\top}\boldsymbol{\Sigma}_{\boldsymbol{d}_{0:i}}^{-1}\boldsymbol{U}_{0:i}(\boldsymbol{\theta}_m-\boldsymbol{\theta}_n), \qquad (4.38)$$

which is exactly the same expression obtained in Equation (4.33). Thus, the dLMS detector approximates the performance of the global optimal detector as we make the estimation learning run slower by choosing smaller values of $\mu$, provided that the diffusion algorithm runs for a sufficient period of time for the approximation in Equation (4.23) to be valid.

[6] For conformable matrices $\boldsymbol{S}$, $\boldsymbol{U}$, $\boldsymbol{Z}$ and $\boldsymbol{V}$, where $\boldsymbol{S}$ and $\boldsymbol{Z}$ are nonsingular matrices, it holds

$$(\boldsymbol{S}+\boldsymbol{UZV})^{-1}=\boldsymbol{S}^{-1}-\boldsymbol{S}^{-1}\boldsymbol{U}(\boldsymbol{Z}^{-1}+\boldsymbol{VS}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VS}^{-1}.$$

[7] As shown in the proof of Proposition 6, for $\boldsymbol{M}$ positive definite and a vector $\boldsymbol{x}$, $\boldsymbol{x}^{\top}\boldsymbol{Mx}$ is a quadratic form, and for any two vectors $\boldsymbol{x}$ and $\boldsymbol{y}\in\mathbb{R}^D$, it holds that $\boldsymbol{x}^{\top}\boldsymbol{Mx}-\boldsymbol{y}^{\top}\boldsymbol{My}=(\boldsymbol{x}+\boldsymbol{y})^{\top}\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y})$.

## 4.5 Synthesis

In order to help assimilate the result just discussed and its correct interpretation, let us synthesize the conditions on which this result depends and its consequences:

1. This chapter described the performance of the ideal dLMS detector in terms of statistical quantities of the local dLMS estimates, namely their expectations, variances and covariances, obtained in Proposition 7. It was shown that if we choose sufficient small step size these quantities are approximated by a linear and a quadratic function of the step size (Propositions 9 and 10), and the performance of the detector does not depend on its specific value (Equation (4.22)). The period of time $i < I_{\text{sup}}$ during which this is valid depends on how small we choose the value of the step size, and larger periods are achievable by means of smaller step sizes.

2. It was assumed that the noise level to which the network is submitted is known or can be estimated, since the noise variances at sensor nodes are necessary to "tune" the detector accordingly (see Equation (4.26) ).

3. During the first iterations of the algorithm, as time increases to $i > I_{\boldsymbol{\pi}}$, the network becomes more similar to one that is totally connected, due to the diffusion process. Using that we approximated the expressions further—note that this state of approximate "total connectivity" of the network is completely independent of the value of the step size; it only depends on the network topology and the combination coefficients.

4. Finally and most important, after a certain number of algorithm iterations $I_{\boldsymbol{\pi}}$ and until a time instant $I_{\text{sup}}$—which we can choose to be as large as we wish by selecting a sufficiently small value of $\mu$—, the dLMS detector approximates the performance of the optimal detector that considers all data at all nodes at all times, although the estimation algorithm at the detector core becomes slower in terms of its learning rate. Therefore, as in the classic Aesop's fable, we should favor the Tortoise over the Hare.

# 5 LOW-COMPLEXITY DISTRIBUTED DETECTOR

In this chapter, we discuss the approximations we can take to produce a detector with low complexity in terms of the number of arithmetic operations required to run the detection routine, aiming at a feasible detector suitable for applications under stringent power limitations. Also, we discuss how the initial estimate of the dLMS can be chosen having in mind a simpler detector and a better performance. To fulfill one of the objectives of this thesis, it is shown how we can estimate the time required (the *stopping time*) for this low-complexity detector to reach a desired minimum performance. It is also explained how this stopping time can be used for the strategy devised to track changes in the state of nature.

## 5.1 Developing the low-complexity detector

Using the results of Section 4.4, we can now find appropriate approximations for matrices $\boldsymbol{R}_k[\,i\,]$ and $\boldsymbol{Q}_k[\,i\,]$. Note that their evaluation, by definition, would be unfeasible in practice, since it implies a tremendous amount of data to be shared between nodes, and matrix inversions are required as well, which is highly costly and, therefore, should not compose an implementable low-cost algorithm.

The following proposition provide us with a more useful and practical expression for the analysis we conduct in this section:

**Proposition 12.** *The test statistic*

$$t'_{n,k}[\,i\,] = (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^\top \boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b}) - \frac{1}{2}\delta^2_{n,k}[\,i\,], \tag{5.1}$$

*where $\delta^2_{n,k}[\,i\,] = (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{R}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b})$, is equivalent to the statistic $t_{n,k}[\,i\,]$ to perform the dLMS detector test in Equation* (3.29).

*Proof.* First, let us reformulate the expression of $t_{n,k}[\,i\,]$ in Equation (3.29) by using a few algebraic manipulations. We start by using

$$\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{\epsilon}_{n,k}[\,i\,] = (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b}) + (\boldsymbol{b} - \boldsymbol{\epsilon}_{n,k}[\,i\,])$$

to rewrite Equation (3.29) as

$$\begin{aligned} t_{n,k}[\,i\,] = &-\frac{1}{2}(\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^\top \boldsymbol{S}_k^{-1}[\,i\,](\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b}) \\ &+ (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^\top \boldsymbol{S}_k^{-1}[\,i\,](\boldsymbol{\epsilon}_{n,k}[\,i\,] - \boldsymbol{b}) \\ &- \frac{1}{2}(\boldsymbol{\epsilon}_{n,k}[\,i\,] - \boldsymbol{b})^\top \boldsymbol{S}_k^{-1}[\,i\,](\boldsymbol{\epsilon}_{n,k}[\,i\,] - \boldsymbol{b}). \end{aligned} \tag{5.2}$$

Now, we use the fact that $\boldsymbol{\epsilon}_{n,k}[\,i\,] - \boldsymbol{b} = \boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b})$, and recall the definitions $\boldsymbol{Q}_k[\,i\,] = \boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{R}_k[\,i\,] = \boldsymbol{P}_k^\top \boldsymbol{Q}_k[\,i\,]$. We now substitute these in Equation (5.2), and removing the term that does not depend on the hypothesis being considered, we obtain the expression in Equation (5.1). □

The first step towards a feasible detector is to simplify the expression in Equation (5.1) and try to minimize the number of arithmetic operations; more specifically, we can try to remove the terms $\delta^2_{n,k}[\,i\,]$ from the hypothesis test by some means. Suppose that we are able to choose the initial estimate $\boldsymbol{b}$ such that we minimize

$$\sum_{n=1}^{N-1}\sum_{m=n+1}^{N}\left(\delta^2_{n,k}[\,i\,]-\delta^2_{m,k}[\,i\,]\right)^2, \tag{5.3}$$

so that we can approximate the terms $\delta^2_{n,k}[\,i\,]$ in Equation (5.1) by a common value (we discuss how this can be done in detail in Section 5.2 ahead). Thus, when performing the hypothesis test, these terms can be canceled out, and we can simplify the statistic simply to

$$t'_{n,k}[\,i\,]=(\hat{\boldsymbol{w}}_k[\,i\,]-\boldsymbol{b})^{\top}\boldsymbol{Q}_k[\,i\,](\boldsymbol{\theta}_n-\boldsymbol{b}). \tag{5.4}$$

Before we find a vector $\boldsymbol{b}$ which satisfies this condition in practice, we need first to find an appropriated approximation for matrices $\boldsymbol{R}_k[\,i\,]$ $\forall k$, as we will see in the next step.

The second step towards an implementable detector is to find an appropriate approximation for both matrices $\boldsymbol{R}_k[\,i\,]$ and $\boldsymbol{Q}_k[\,i\,]$. Using the approximations of $\boldsymbol{P}_k[\,i\,]$ and $\boldsymbol{Z}_k[\,i\,]$ in Equation (4.27), we have that $\boldsymbol{P}_k^{-1}[\,i\,]\boldsymbol{Z}_k[\,i\,]=\mu\boldsymbol{I}_M$ and, therefore, we can approximate $\boldsymbol{Q}[\,i\,]$ as

$$\begin{aligned}\boldsymbol{Q}_k[\,i\,]&=\boldsymbol{S}_k^{-1}[\,i\,]\boldsymbol{P}_k[\,i\,]\\&=\left(\boldsymbol{P}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,]+\boldsymbol{Z}_k[\,i\,]\right)^{-1}\boldsymbol{P}_k[\,i\,]\\&=\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,]+\boldsymbol{P}_k^{-1}[\,i\,]\boldsymbol{Z}_k[\,i\,]\right)^{-1}\\&\approx\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,]+\mu\boldsymbol{I}_D\right)^{-1}\end{aligned}$$

and from the definition of $\boldsymbol{R}_k[\,i\,]$, we have

$$\begin{aligned}\boldsymbol{R}_k[\,i\,]&=\boldsymbol{P}_k^{\top}[\,i\,]\boldsymbol{Q}_k[\,i\,]\\&\approx\boldsymbol{P}_k^{\top}[\,i\,]\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,]+\mu\boldsymbol{I}_D\right)^{-1}\\&=\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}+\mu\boldsymbol{P}_k^{-1}[\,i\,]\right)^{-1}\end{aligned}$$

using the fact that the approximation of $\boldsymbol{P}_k[\,i\,]$ in Equation (4.27) is a symmetric matrix. We now recall the fact that $\boldsymbol{u}_k[\,i\,]$ was modeled as a *realization* of a random vector $\underset{\sim}{\boldsymbol{u}}_k[\,i\,]$. Suppose that it is modeled as a *wide sense stationary* process[1] and $\underset{\sim}{\boldsymbol{u}}_k[\,i\,]\sim\mathcal{N}\left(\boldsymbol{0},\boldsymbol{\Sigma}_{\boldsymbol{u}_k}\right)$; thus, let us use the fact that

$$\sum_{j=0}^{i}\frac{\boldsymbol{u}_k[\,j\,]\boldsymbol{u}_k^{\top}[\,j\,]}{i+1}\Longrightarrow\boldsymbol{\Sigma}_{\boldsymbol{u}_k},\ \text{ as } i \text{ increases}, \tag{5.5}$$

[1] The stationary condition is required to keep the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{u}_k}$ constant during the detection routine; otherwise, it must be updated as time progresses. Also note that the stationary condition only need to be true during the duration of the detection routine: once it is finished, a new updated covariance matrix can be used if necessary, supposing that it is constant during the next detection routine.

to further approximate $\boldsymbol{P}_k[\,i\,]$ for a $i > I_{\boldsymbol{u}}$ sufficiently large. Therefore, we have from Equation (4.27) that

$$\boldsymbol{P}_k[\,i\,] \approx (i+1)\,\mu \sum_{\ell=1}^{K} \frac{\boldsymbol{\Sigma}_{\boldsymbol{u}_\ell}}{\sigma^2_{v_\ell}} = (i+1)\,\mu \boldsymbol{P}_{\mathrm{ct}}\,, \tag{5.6}$$

where we define the constant matrix $\boldsymbol{P}_{\mathrm{ct}} = \sum_{\ell=1}^{K} \boldsymbol{\Sigma}_{\boldsymbol{u}_\ell}/\sigma^2_{v_\ell}$. Hence, our approximations of $\boldsymbol{Q}_k[\,i\,]$ and $\boldsymbol{R}_k[\,i\,]$ can now be written as

$$\begin{aligned}\boldsymbol{Q}_k[\,i\,] &\approx \left((i+1)\mu\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_{\mathrm{ct}} + \mu\boldsymbol{I}_D\right)^{-1}\\ \boldsymbol{R}_k[\,i\,] &\approx \left(\boldsymbol{\Sigma}_{\boldsymbol{w}} + (i+1)^{-1}\boldsymbol{P}_{\mathrm{ct}}^{-1}\right)^{-1},\end{aligned} \tag{5.7}$$

valid for $I_{\mathrm{inf}} = \max\{I_{\boldsymbol{\pi}},\, I_{\boldsymbol{u}}\} < i < I_{\mathrm{sup}}$ and $\mu < \mu_{\mathrm{sup}}$. Therefore, we can further approximate the matrix inversions above, as $i$ increases, as

$$\boldsymbol{Q}_k[\,i\,] \approx \left((i+1)\mu\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_{\mathrm{ct}}\right)^{-1}, \quad \boldsymbol{R}_k[\,i\,] \approx \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}. \tag{5.8}$$

As we can see, all quantities, except for $i$, are known and constant by assumption. Next, let us substitute Equation (5.5) in the test statistic in Equation (5.4), which results in an approximation of it, assuming that we choose am adequate $\boldsymbol{b}$:

$$t'_{n,k}[\,i\,] \approx (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^{\top}\left((i+1)\mu\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_{\mathrm{ct}}\right)^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}).$$

Defining the constant vector

$$\boldsymbol{q}_n = \left(\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_{\mathrm{ct}}\right)^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}), \tag{5.9}$$

the test using this approximation of $t'_{n,k}[\,i\,]$ is given as

$$\frac{(\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^{\top}\boldsymbol{q}_n}{(i+1)\mu} \underset{\Phi_k \neq n}{\overset{\Phi_k \neq m}{\gtrless}} \frac{(\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^{\top}\boldsymbol{q}_n}{(i+1)\mu} + \gamma_{mn}.$$

Thus, multiplying both sides of the test by $(i+1)\mu$ and defining $t''_{n,k}[\,i\,] = (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^{\top}\boldsymbol{q}_n$, we have the following new low-complexity test:

$$\begin{aligned} t''_{n,k}[\,i\,] &= (\hat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b})^{\top}\boldsymbol{q}_n,\\ t''_{n,k}[\,i\,] &\underset{\Phi_k \neq n}{\overset{\Phi_k \neq m}{\gtrless}} t''_{m,k}[\,i\,] + (i+1)\mu\gamma_{mn},\end{aligned} \tag{5.10}$$

valid for $I_{\mathrm{inf}} < i < I_{\mathrm{sup}}$ and $\mu < \mu_{\mathrm{sup}}$. Note that the test statistic is a simple dot product between two vectors: our local diffusion estimate $\hat{\boldsymbol{w}}_k[\,i\,]$ and two constant vectors $\boldsymbol{b}$ and $\boldsymbol{q}_n$. Therefore, this version of the proposed detector is certainly feasible. Furthermore, from a computational complexity viewpoint, it is surprisingly cheap: it requires for each iteration $i$ only $D$ products and $2D - 1$ sums to evaluate the test statistic. Since there are $N$ statistics, there are $ND$ products and $2ND - N$ sums.

We must recall that the low-complexity detector expressed in Equation 5.10 depends on the assumption that the initial estimate $\boldsymbol{b}$ is chosen so that the terms $\delta^2_{n,k}[\,i\,]$ in Equation (5.1) are sufficiently close to be removed from the hypothesis test. In the next section, we see how to choose $\boldsymbol{b}$ in such a way as to meet this requirement. Just as importantly, we should be able to estimate the number of iterations that is needed to the the detector to reach a certain desired performance level. This topic will be discussed ahead in Section 5.3.

## 5.2 Choosing the initial estimate

In Chapter 4, choosing $\boldsymbol{b} = \boldsymbol{0}$ was useful to describe mathematically the performance of the proposed detector, but it is not necessarily the best choice for an implementable low-cost version. We will see in this section how to choose $\boldsymbol{b}$ that simplifies calculations.

We can now use the approximation of $\boldsymbol{R}_k[\,i\,]$ obtained in Equation (5.8) to find the initialization $\boldsymbol{b}$ such that it minimizes Equation (5.3), which allows us to simplify the detector by using the test statistic in Equation (5.10) for $I_{\text{inf}} < i < I_{\text{sup}}$ and $\mu < \mu_{\text{sup}}$. First, note that the cost function in Equation (5.3) can be made equal to zero if there exists a value of $\boldsymbol{b}$ such that

$$(\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b})^\top = (\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{b})^\top,\ \forall m \neq n.$$

Such value can be found, if it exists, by solving

$$\begin{aligned}
(\boldsymbol{\theta}_1 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_1 - \boldsymbol{b}) &= \delta^2 \\
(\boldsymbol{\theta}_2 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_2 - \boldsymbol{b}) &= \delta^2 \\
&\vdots \\
(\boldsymbol{\theta}_N - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_N - \boldsymbol{b}) &= \delta^2.
\end{aligned} \tag{5.11}$$

Without loss of generality, for a $n \in \{1, \ldots, N\}$ let us substitute $\delta^2 = (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b})$ in the other $N-1$ equations; thus,

$$\begin{aligned}
(\boldsymbol{\theta}_1 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_1 - \boldsymbol{b}) &= (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}) \\
(\boldsymbol{\theta}_2 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_2 - \boldsymbol{b}) &= (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}) \\
&\vdots \\
(\boldsymbol{\theta}_N - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_N - \boldsymbol{b}) &= (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}).
\end{aligned} \tag{5.12}$$

Note, for $m \neq n$, that

$$\begin{aligned}
&(\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{b}) = (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}) \\
\Rightarrow\ &(\boldsymbol{\theta}_m + \boldsymbol{\theta}_n - 2\boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) = 0 \\
\Rightarrow\ &(\boldsymbol{\theta}_m + \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) - 2\boldsymbol{b}^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) = 0 \\
\Rightarrow\ &(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{b} - \frac{1}{2} (\boldsymbol{\theta}_m + \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) = 0 \\
\Rightarrow\ &(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{b} - \frac{1}{2} \boldsymbol{\theta}_m^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_m + \frac{1}{2} \boldsymbol{\theta}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_n = 0,
\end{aligned}$$

which is a linear equation on $\boldsymbol{b}$. Therefore, we can rewrite the equations in Equation (5.12) into a system of $N-1$ linear equations and $D$ unknowns; in matrix form it is given as

$$\begin{bmatrix} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_n)^\top \\ \vdots \\ (\boldsymbol{\theta}_N - \boldsymbol{\theta}_n)^\top \end{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{b} = \frac{1}{2} \begin{bmatrix} \boldsymbol{\theta}_1^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_1 - \boldsymbol{\theta}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_n \\ \vdots \\ \boldsymbol{\theta}_N^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_N - \boldsymbol{\theta}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \boldsymbol{\theta}_n \end{bmatrix}. \tag{5.13}$$

This system can have a unique solution, an infinity of solution or no solutions at all. The latter happens if the vector on the right-hand side of Equation (5.13) is not in the range space of the matrix multiplying $\boldsymbol{b}$ in the left-hand side. In this case, the best we can do is to choose a $\boldsymbol{b}$ that minimizes Equation (5.3). Having said that, let us know investigate the possible solutions to this system.

### 5.2.1 Case I: $D = N - 1$

Note that the system in Equation (5.13) has a unique solution only if $D = N - 1$. In this case, the matrix multiplying $\boldsymbol{b}$ on the left must have full rank; in other words, if no row in that matrix can be written as a linear combination of the other rows. i.e.,

$$\alpha_1(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \ldots + \alpha_N(\boldsymbol{\theta}_N - \boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} = \mathbf{0} \iff \alpha_m = 0 \text{ for } m = 1, \ldots, N, m \neq n.$$

We can rearrange the terms of the sum above to obtain

$$\left(\alpha_1\boldsymbol{\theta}_1 + \ldots + \alpha_N\boldsymbol{\theta}_N - (\alpha_1 + \ldots + \alpha_N)\boldsymbol{\theta}_n\right)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} = \mathbf{0}.$$

Since the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ has full rank by assumption, we have two types of solution to the equation above:

$$\alpha_m = 0 \text{ for } m = 1, \ldots, N, m \neq n, \text{ or}$$
$$\boldsymbol{\theta}_n = \frac{\alpha_1\boldsymbol{\theta}_1 + \ldots + \alpha_N\boldsymbol{\theta}_N}{\alpha_1 + \ldots + \alpha_N}$$

Thus, if $\boldsymbol{\theta}_n$ is an *affine combination* of $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$, the linear system in Equation (5.13) has no solution. Interestingly, as $\boldsymbol{\theta}_n \in \mathbb{R}^{N-1}$, if $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\} \setminus \{\boldsymbol{\theta}_n\}$ is a linearly independent set, $\boldsymbol{\theta}_n$ necessarily is a linear combination of the others, since they form a basis of $\mathbb{R}^{N-1}$; however, only if $\boldsymbol{\theta}_n$ is an affine combination of the other vectors that the system in Equation (5.13) has no solution. For example, when $D = N - 1 = 2$, there exists no $\boldsymbol{b}$ that satisfies Equation (5.13) if $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ are collinear in $\mathbb{R}^2$, since in this case any vector in the set is an affine combination of the others. Similarly, for $D = N - 1 = 3$, there is no solution if $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4\}$ are coplanar in $\mathbb{R}^3$.

### 5.2.2 Case II: $N - 1 < D$

In this case, if the system in Equation (5.13) has solution, there are an infinity of them. However, we should be cautious about our choice of $\boldsymbol{b}$: as we will see in this subsection, when we approximated the dLMS statistic $t'_{n,k}[i]$ defined in Section 3.2 by the low-complexity statistic $t''_{n,k}[i]$ in Equation (5.10), the property that the detection performance does not depend on a specific value of $\boldsymbol{b}$ does not hold anymore, and a good choice for $\boldsymbol{b}$ is, among all that satisfy the system in Equation (5.13), the one with the least norm. To show that, let us define the error in approximation

$$e_{n,k}[i] = t'_{n,k}[i] - t''_{n,k}[i], \tag{5.14}$$

and suppose we choose a $\boldsymbol{b}$ that is a solution of Equation (5.13). Also, define the following error matrices with respect to the approximation in Equation (5.8):

$$\begin{aligned}\widetilde{\boldsymbol{Q}}_k[i] &= \boldsymbol{Q}[i] - ((i+1)\mu\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_{\text{ct}})^{-1},\\ \widetilde{\boldsymbol{R}}_k[i] &= \boldsymbol{R}_k[i] - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}.\end{aligned}$$

Thus, we can rewrite Equation (5.14) as follows:

$$e_{n,k}[i] = (\hat{\boldsymbol{w}}_k[i] - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{Q}}_k[i](\boldsymbol{\theta}_n - \boldsymbol{b}) - \frac{1}{2}\delta^2_{n,k}[i]. \tag{5.15}$$

When comparing a pair of statistics $\{t'_{n,k}[i], t'_{m,k}[i]\}$ of the dLMS detector, we have

$$\begin{aligned} t'_{n,k}[i] &\underset{\Phi_k\neq n}{\overset{\Phi_k\neq m}{\gtrless}} t'_{m,k}[i] + \gamma_{mn} \Rightarrow\\ t''_{n,k}[i] + e_{n,k}[i] &\underset{\Phi_k\neq n}{\overset{\Phi_k\neq m}{\gtrless}} t''_{m,k}[i] + e_{m,k}[i] + \gamma_{mn} \Rightarrow\\ t''_{n,k}[i] &\underset{\Phi_k\neq n}{\overset{\Phi_k\neq m}{\gtrless}} t''_{m,k}[i] + \gamma_{mn} + (e_{m,k}[i] - e_{n,k}[i]); \end{aligned}$$

therefore, we can notice that the error committed when comparing $\{t''_{n,k}[i], t''_{m,k}[i]\}$ instead of $\{t'_{n,k}[i], t'_{m,k}[i]\}$ is embedded in $e_{m,k}[i] - e_{n,k}[i]$. Let us call $E_{mn,k}[i]$ this detection error due to approximations when comparing $H_n$ and $H_n$, which can be written, from Equation (5.15), as

$$E_{mn,k}[i] = (\hat{\boldsymbol{w}}_k[i] - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{Q}}_k[i](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) - \frac{1}{2}(\delta^2_{m,k}[i] - \delta^2_{n,k}[i]). \tag{5.16}$$

Recall that $\delta^2_{n,k}[i] = (\boldsymbol{\theta}_n - \boldsymbol{b})^{\top}\boldsymbol{R}_k[i](\boldsymbol{\theta}_n - \boldsymbol{b})$; since $\boldsymbol{R}_k[i] = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \widetilde{\boldsymbol{R}}_k[i]$, and given that we chose a $\boldsymbol{b}$ such that $(\boldsymbol{\theta}_n - \boldsymbol{b})^{\top}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}) = (\boldsymbol{\theta}_m - \boldsymbol{b})^{\top}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{b})\ \forall m \neq n$, we have

$$\delta^2_{m,k}[i] - \delta^2_{n,k}[i] = (\boldsymbol{\theta}_m - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{R}}_k[i](\boldsymbol{\theta}_m - \boldsymbol{b}) - (\boldsymbol{\theta}_n - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{R}}_k[i](\boldsymbol{\theta}_n - \boldsymbol{b}).$$

Also notice that $\delta^2_{m,k}[i] - \delta^2_{n,k}[i]$ above is a difference of two quadratic forms; hence,

$$\delta^2_{m,k}[i] - \delta^2_{n,k}[i] = (\boldsymbol{\theta}_m + \boldsymbol{\theta}_n - 2\boldsymbol{b})^{\top}\widetilde{\boldsymbol{R}}_k[i](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n);$$

thus, we can rewrite Equation (5.16) as

$$E_{mn,k}[i] = (\hat{\boldsymbol{w}}_k[i] - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{Q}}_k[i](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) - (\bar{\boldsymbol{\theta}}_{mn} - \boldsymbol{b})^{\top}\widetilde{\boldsymbol{R}}_k[i](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n), \tag{5.17}$$

where $\bar{\boldsymbol{\theta}}_{mn} = \frac{\boldsymbol{\theta}_m + \boldsymbol{\theta}_n}{2}$.

It makes sense to choose $\boldsymbol{b}$ such that it makes the expression in Equation (5.17) as close to zero as possible when $H_n$ or $H_m$ is the true hypothesis [2]. A suitable candidate for an objective

[2] Note that comparing $\{t''_{n,k}[i], t''_{m,k}[i]\}$ when $H_\nu$ is true, $\nu \neq m$ and $\nu \neq n$, does not influence the detector performance; therefore, these cases can be ignored.

function in this case is the *maximum* mean square detection error $\mathrm{E}\big(E^2_{mn,k}[\,i\,]\big)$; in other words, we want

$$\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}} \quad \max\, \mathrm{E}\big(E^2_{mn,k}[\,i\,]\big),$$
$$\text{subject to: } (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}) = (\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{b}),\ \forall m \neq n.$$

Since $\mathrm{E}\big(E^2_{mn,k}[\,i\,]\big)$ depends on the active hypothesis, let us consider as a mean square detection error a *weighted* mean square error, given as

$$\mathrm{E}\big(E^2_{mn,k}[\,i\,]\big) = \frac{p(H_m)\mathrm{E}\big(E^2_{mn,k}[\,i\,] \,|\, H_m\big) + p(H_n)\mathrm{E}\big(E^2_{mn,k}[\,i\,] \,|\, H_n\big)}{p(H_m) + p(H_n)}. \tag{5.18}$$

Also, define for later use the following normalize probabilities:

$$p'(H_m) = \frac{p(H_m)}{p(H_m) + p(H_n)}, \quad \text{and} \quad p'(H_n) = \frac{p(H_n)}{p(H_m) + p(H_n)}. \tag{5.19}$$

In order to evaluate the expression above, we use the fact that

$$\mathrm{E}\big(E^2_{mn,k}[\,i\,] \,|\, H_\nu\big) = \mathrm{Var}\big(E_{mn,k}[\,i\,]\big) + \big(\mathrm{E}\big(E_{mn,k}[\,i\,] \,|\, H_\nu\big)\big)^2;$$

thus, from Equation (5.17), we have

$$\mathrm{Var}\big(E_{mn,k}[\,i\,]\big) = (\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top \widetilde{\boldsymbol{Q}}_k^\top[\,i\,]\boldsymbol{S}_k[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n),$$

which does not depend on $\boldsymbol{b}$. Next, calculate

$$\mathrm{E}\big(E_{mn,k}[\,i\,] \,|\, H_\nu\big) = (\boldsymbol{\theta}_\nu - \boldsymbol{b})^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) - (\bar{\boldsymbol{\theta}}_{mn} - \boldsymbol{b})^\top \widetilde{\boldsymbol{R}}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n), \tag{5.20}$$

where we used the fact from Equation (4.8) that

$$\mathrm{E}(\widehat{\boldsymbol{w}}_k[\,i\,] - \boldsymbol{b} \,|\, H_\nu) = \mathrm{E}(\boldsymbol{L}_k[\,i\,]\boldsymbol{d}_{0:i} - \boldsymbol{P}_k[\,i\,]\boldsymbol{b} \,|\, H_\nu) = \boldsymbol{P}_k[\,i\,]\boldsymbol{\theta}_\nu.$$

Let us rewrite this expression more adequately as follows:

$$\mathrm{E}\big(E_{mn,k}[\,i\,] \,|\, H_\nu\big) = \boldsymbol{b}^\top(\widetilde{\boldsymbol{R}}_k[\,i\,] - \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,])(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) + \varepsilon_{mn|\nu}[\,i\,], \tag{5.21}$$

where $\varepsilon_{mn|\nu}[\,i\,]$ represents the terms of $\mathrm{E}\big(E_{mn,k}[\,i\,] \,|\, H_\nu\big)$ that do not depend on $\boldsymbol{b}$; i.e.,

$$\varepsilon_{mn|\nu}[\,i\,] = \boldsymbol{\theta}_\nu^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n) - \bar{\boldsymbol{\theta}}_{mn}^\top \widetilde{\boldsymbol{R}}_k[\,i\,](\boldsymbol{\theta}_m - \boldsymbol{\theta}_n). \tag{5.22}$$

Let us define the vector $\boldsymbol{\chi}_{mn}[\,i\,] = (\widetilde{\boldsymbol{R}}_k[\,i\,] - \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,])(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)$; thus, from Equation (5.21) we have

$$\begin{aligned}\big(\mathrm{E}\big(E_{mn,k}[\,i\,] \,|\, H_\nu\big)\big)^2 &= \big(\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,] + \varepsilon_{mn|\nu}[\,i\,]\big)^2 \\ &= \big(\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,]\big)^2 + 2\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,]\varepsilon_{mn|\nu}[\,i\,] + \varepsilon^2_{mn|\nu}[\,i\,].\end{aligned}$$

Therefore, from Equation (5.18) and Equation (5.19), we obtain

$$\begin{aligned}\mathrm{E}\Big(E_{mn,k}^2[\,i\,]\Big) = {}& (\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,])^2 + \mathrm{Var}\big(E_{mn,k}[\,i\,]\big)\\ &+ 2\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,]\Big(p'(H_m)\varepsilon_{mn|m}[\,i\,] + p'(H_n)\varepsilon_{mn|n}[\,i\,]\Big)\\ &+ p'(H_m)\varepsilon_{mn|m}^2[\,i\,] + p'(H_n)\varepsilon_{mn|n}^2[\,i\,].\end{aligned}$$

According to the Cauchy-Schwartz inequality, we must have

$$\big|\,\boldsymbol{b}^\top \boldsymbol{\chi}_{mn}[\,i\,]\,\big| \le \|\,\boldsymbol{b}\,\|_2 \cdot \|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2\,;$$

thus,

$$\begin{aligned}\mathrm{E}\Big(E_{mn,k}^2[\,i\,]\Big) \le {}& \|\,\boldsymbol{b}\,\|_2^2 \cdot \|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2^2 + \mathrm{Var}\big(E_{mn,k}[\,i\,]\big)\\ &+ 2\,\|\,\boldsymbol{b}\,\|_2 \cdot \|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2 \cdot \big|\,p'(H_m)\varepsilon_{mn|m}[\,i\,] + p'(H_n)\varepsilon_{mn|n}[\,i\,]\,\big|\\ &+ p'(H_m)\varepsilon_{mn|m}^2[\,i\,] + p'(H_n)\varepsilon_{mn|n}^2[\,i\,].\end{aligned}$$

Let us define the function $J_{mn}(x)$ for $x \in \mathbb{R}$ as follows:

$$\begin{aligned}J_{mn}(x) = {}& \|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2^2\, x^2 + \mathrm{Var}\big(E_{mn,k}[\,i\,]\big)\\ &+ 2\,\|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2 \cdot \big|\,p'(H_m)\varepsilon_{mn|m}[\,i\,] + p'(H_n)\varepsilon_{mn|n}[\,i\,]\,\big|\, x\\ &+ p'(H_m)\varepsilon_{mn|m}^2[\,i\,] + p'(H_n)\varepsilon_{mn|n}^2[\,i\,].\end{aligned}$$

Since $\frac{d^2}{dx^2} J_{mn}(x) = 2\,\|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2^2 > 0$, $J_{mn}(x)$ has a minimum value which can be found by solving $\frac{d}{dx} J_{mn}(x) = 0$: thus,

$$\begin{aligned}&2\,\|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2^2\; x + 2\,\|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2 \cdot \big|\,p'(H_m)\varepsilon_{mn|m}[\,i\,] + p'(H_n)\varepsilon_{mn|n}[\,i\,]\,\big| = 0\\ &\Rightarrow x = -\frac{\big|\,p'(H_m)\varepsilon_{mn|m}[\,i\,] + p'(H_n)\varepsilon_{mn|n}[\,i\,]\,\big|}{\|\,\boldsymbol{\chi}_{mn}[\,i\,]\,\|_2} = x_{\min}.\end{aligned}$$

Hence, $J_{mn}(x)$ is non-decreasing for $x > x_{\min}$; furthermore, since $x_{\min} < 0$, $J_{mn}(x)$ is non-decreasing for $x > 0$. Thus, making $x = \|\,\boldsymbol{b}\,\|_2$, the new function $G_{mn}(\,\boldsymbol{b}\,) = J_{mn}(\|\,\boldsymbol{b}\,\|_2)$ can be minimized as $\boldsymbol{b}$ is chosen such that $\|\,\boldsymbol{b}\,\|_2$ is minimum. Therefore, by finding a minimum $\|\,\boldsymbol{b}\,\|_2$ we are also minimizing the maximum value of $\mathrm{E}\Big(E_{mn,k}^2[\,i\,]\Big)$, since

$$\mathrm{E}\Big(E_{mn,k}^2[\,i\,]\Big) \le G_{mn}(\,\boldsymbol{b}\,).$$

Finally, since this result does not depend on which pair of hypotheses we choose, by choosing a $\boldsymbol{b}$ such that its norm is minimum reduces the maximum mean square error of the low-complexity test in Equation (5.10). Let us state this result as a new proposition.

**Proposition 13.** *When using the low-complexity dLMS test defined in Equation (5.10), in order to minimize the maximum mean square detection error $\mathrm{E}\Big(E_{mn,k}^2[\,i\,]\Big)$ in Equation (5.18), committed due to the approximations when comparing any pair of statistics $\{t''_{n,k}[\,i\,], t''_{m,k}[\,i\,]\}$ when $H_n$ or $H_m$ is the true hypothesis, we must choose a value of the initial estimate $\boldsymbol{b}$ such that*

$$\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}} \|\,\boldsymbol{b}\,\|_2\,,$$
$$\text{subject to: } (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}) = (\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{b}),\ \forall m \ne n.$$

A very special case happens when $N = 2$ (i.e., a binary detection problem) and $p(H_1) = p(H_2)$[3]: we can show that $\boldsymbol{b} = \bar{\boldsymbol{\theta}}_{12} = \frac{1}{2}(\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2)$ in fact minimizes the mean square detection error directly, not just its maximum value. In other words,

$$\begin{aligned} \bar{\boldsymbol{\theta}}_{12} &= \arg\min_{\boldsymbol{b}} \mathrm{E}\Big(E^2_{21,k}[\,i\,]\Big), \\ \text{subject to: } (\boldsymbol{\theta}_1 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_1 - \boldsymbol{b}) &= (\boldsymbol{\theta}_2 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_2 - \boldsymbol{b}). \end{aligned}$$

It is trivial to see that $\bar{\boldsymbol{\theta}}_{12}$ is a solution of Equation (5.12):

$$\begin{aligned} &(\boldsymbol{\theta}_1 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_1 - \boldsymbol{b}) = (\boldsymbol{\theta}_2 - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_2 - \boldsymbol{b}) \\ \Rightarrow\ &(\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2 - 2\boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) = 0. \end{aligned}$$

Clearly, any vector $\boldsymbol{b} = \bar{\boldsymbol{\theta}}_{12} + \alpha\boldsymbol{n}$ such that $\boldsymbol{n}^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) = 0$, $\forall \alpha \in \mathbb{R}$, is also a solution. But we are going to see that $\alpha = 0$ in fact minimizes the mean square detection error as defined in Equation (5.18). From Equation (5.20), we have, for $N = 2$ and making $\boldsymbol{b} = \bar{\boldsymbol{\theta}}_{12} + \alpha\boldsymbol{n}$, the following:

$$\begin{aligned} \mathrm{E}\big(E_{21,k}[\,i\,]\,|\,H_\nu\big) &= (\boldsymbol{\theta}_\nu - \bar{\boldsymbol{\theta}}_{12} - \alpha\boldsymbol{n})^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) + \alpha\boldsymbol{n}^\top \widetilde{\boldsymbol{R}}_k[\,i\,](\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \\ &= (\boldsymbol{\theta}_\nu - \bar{\boldsymbol{\theta}}_{12})^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \\ &\quad + \alpha\boldsymbol{n}^\top\big(\widetilde{\boldsymbol{R}}_k[\,i\,] - \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,]\big)(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1). \end{aligned}$$

Call $\zeta_{\nu,k} = (\boldsymbol{\theta}_\nu - \bar{\boldsymbol{\theta}}_{12})^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)$ and $\boldsymbol{\kappa}_{21,k} = \big(\widetilde{\boldsymbol{R}}_k[\,i\,] - \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,]\big)(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)$. Thus, we have

$$\Big(\mathrm{E}\big(E_{21,k}[\,i\,]\,|\,H_\nu\big)\Big)^2 = (\alpha\boldsymbol{n}^\top\boldsymbol{\kappa}_{21,k})^2 + 2\alpha\boldsymbol{n}^\top\boldsymbol{\kappa}_{21,k}\zeta_{\nu,k} + \zeta^2_{\nu,k};$$

from Equation (5.18) and the fact that $p(H_1) = p(H_2)$, we have

$$\mathrm{E}\Big(E^2_{21,k}[\,i\,]\Big) = (\alpha\boldsymbol{n}^\top\boldsymbol{\kappa}_{21,k})^2 + 2\alpha\boldsymbol{n}^\top\boldsymbol{\kappa}_{21,k}(\zeta_{1,k} + \zeta_{2,k}) + \zeta^2_{1,k} + \zeta^2_{2,k} + \mathrm{Var}(E_{21,k}[\,i\,]).$$

Note that $\zeta_{2,k} = -\zeta_{1,k} = (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)^\top \boldsymbol{P}_k^\top[\,i\,]\widetilde{\boldsymbol{Q}}_k[\,i\,](\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)$; thus, $\zeta_{1,k} + \zeta_{2,k} = 0$ and

$$\mathrm{E}\Big(E^2_{21,k}[\,i\,]\Big) = (\alpha\boldsymbol{n}^\top\boldsymbol{\kappa}_{21,k})^2 + \zeta^2_{1,k} + \zeta^2_{2,k} + \mathrm{Var}(E_{21,k}[\,i\,]).$$

Therefore, the minimum value of the mean square detection error is reached for $\alpha = 0$, as we wanted to demonstrate.

### 5.2.3 Case III: $N - 1 > D$

In this case, we just choose a $\boldsymbol{b}$ that minimizes Equation (5.3), since a unique solution to the system in Equation (5.13) depends on the very particular condition where the vector on the right-hand side of Equation (5.13) is in the range space of the matrix multiplying $\boldsymbol{b}$ in the left-hand side.

[3] In this case, the MAP detector defined in Chapter 2 reduces to what is called the *Maximum Likelihood* (ML) detector, which was studied in [60].

## 5.3 Stopping time

### 5.3.1 *Off-line* estimation of the stopping time

Prior to an eventual deployment, we can estimate the number of iterations of the detection algorithm to reach a desirable probability of error $\varepsilon$—in other words, the *stopping time*. In fact, this can be done without having to simulate a distributed network in its entirety: we need only to calculate the distribution of the statistics at each time $i$. To do so, we use recursive versions of Equations (3.20) and (4.19) to calculate the matrices $\mathcal{P}[\,i\,]$ and $\mathcal{Z}[\,i\,]$, which are

$$\mathcal{P}[\,i\,] = \mathcal{A}(\mathcal{Y}[\,i\,]\mathcal{P}[\,i-1\,] + \boldsymbol{H}[\,i\,]), \tag{5.23}$$

$$\mathcal{Z}[\,i\,] = \mathcal{A}(\mathcal{Y}[\,i\,]\mathcal{Z}[\,i-1\,]\mathcal{Y}^{\top}[\,i\,] + \mathcal{D}[\,i\,])\mathcal{A}^{\top}, \tag{5.24}$$

recalling $\mathcal{Y}[\,i\,] = \operatorname{diag}\{\boldsymbol{Y}_1[\,i\,], \ldots, \boldsymbol{Y}_K[\,i\,]\}$ in (3.14), $\boldsymbol{H}[\,i\,] = \mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{U}[\,j\,]$ in (4.14) and $\mathcal{D}[\,i\,] = \mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{U}^{\top}[\,i\,]\mathcal{M}^{\top}$ in (4.19). The recursive expression of $\mathcal{P}[\,i\,]$ in (5.23) was obtained in Appendix B, Equation (B.4). To obtain the recursive expression of $\mathcal{Z}[\,i\,]$ in (5.24), we use the definitions $\mathcal{Z}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}^{\top}[\,i\,]$ and $\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}} = \boldsymbol{I}_{i+1} \otimes \boldsymbol{\Sigma}_{\boldsymbol{w}}$. Thus, we have

$$\begin{aligned}
\mathcal{Z}[\,i\,] &= \begin{bmatrix}\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,] & \mathcal{A}\,\mathcal{Y}[\,i\,]\mathcal{L}[\,i-1\,]\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{w}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i-1}}\end{bmatrix} \begin{bmatrix}(\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,])^{\top} \\ (\mathcal{A}\,\mathcal{Y}[\,i\,]\mathcal{L}[\,i-1\,])^{\top}\end{bmatrix} \\
&= \mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{U}^{\top}[\,i\,]\mathcal{M}^{\top}\mathcal{A}^{\top} + \mathcal{A}\,\mathcal{Y}[\,i\,]\mathcal{L}[\,i-1\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i-1}}\mathcal{L}^{\top}[\,i-1\,]\mathcal{Y}^{\top}[\,i\,]\mathcal{A}^{\top} \\
&= \mathcal{A}(\mathcal{D}[\,i\,] + \mathcal{Y}[\,i\,]\mathcal{Z}[\,i-1\,]\mathcal{Y}^{\top}[\,i\,])\mathcal{A}^{\top}.
\end{aligned}$$

Next, we estimate the local probabilities of error $\xi_k[\,i\,]$ by performing a Monte Carlo simulation with $L$ realizations[4] by which we generate samples of a pseudo-random vector whose $N$ entries represent the $N$ local statistics $t''[\,i\,]$, $1 \leq n \leq N$, at node $k$. Supposing $H_n$ true, this pseudo-random vector has a distribution given as

$$\underset{\sim}{\boldsymbol{t}}{}_k''[\,i\,] \mid H_n \sim \mathcal{N}(\widehat{\boldsymbol{Q}}^{\top}\boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b}), \widehat{\boldsymbol{Q}}^{\top}\boldsymbol{S}_k[i]\widehat{\boldsymbol{Q}}), \tag{5.25}$$

where we define $\widehat{\boldsymbol{Q}} = [\,\boldsymbol{q}_1 \;\ldots\; \boldsymbol{q}_N\,]$, where the $\boldsymbol{q}_n$ were define in Equation (5.9). To verify that the distribution of $\underset{\sim}{\boldsymbol{t}}{}_k''[\,i\,]$ is correct, we start noting that, from (3.28) and (5.10), we have

$$\begin{aligned}
\mathrm{E}\Big(t''_{m,k}[\,i\,] \,|\, H_n\Big) &= \boldsymbol{q}_m^{\top}\boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b}), \\
\mathrm{Var}\Big(t''_{m,k}[\,i\,]\Big) &= \boldsymbol{q}_m^{\top}\boldsymbol{S}_k[\,i\,]\boldsymbol{q}_m, \\
\mathrm{Cov}\Big(t''_{m,k}[\,i\,], t''_{\nu,k}[\,i\,]\Big) &= \boldsymbol{q}_m^{\top}\boldsymbol{S}_k[\,i\,]\boldsymbol{q}_{\nu}.
\end{aligned}$$

Define $\boldsymbol{t}_k''[\,i\,] = \operatorname{col}\{t''_{1,k}[\,i\,], \ldots, t''_{N,k}[\,i\,]\}$. Therefore, we have

$$\begin{aligned}
\mathrm{E}(\boldsymbol{t}_k'[\,i\,] \,|\, H_n) &= \widehat{\boldsymbol{Q}}^{\top}\boldsymbol{P}_k[\,i\,](\boldsymbol{\theta}_n - \boldsymbol{b}), \\
\mathrm{Cov}(\boldsymbol{t}_k[\,i\,]) &= \widehat{\boldsymbol{Q}}^{\top}\boldsymbol{S}_k[\,i\,]\widehat{\boldsymbol{Q}},
\end{aligned}$$

[4] For a binary hypothesis test, one can obtain the probabilities of error directly from a standard normal table since in this case the cumulative distribution in (4.4) becomes univariate, depending only on $T_{12,k}[\,i\,]$.

which are, as a matter of course, the expected value and variance of $\underset{\sim}{\boldsymbol{t}}''_k[\,i\,]$ in (5.25).

With the samples of $\underset{\sim}{\boldsymbol{t}}''_k[\,i\,]$ in each Monte Carlo trial, we can simulate the hypothesis tests at each node, and then estimate the local probabilities of error $\xi_k[\,i\,]$ from the relative frequency of incorrect detection in the total number of trials. We then compare, for each iteration $i$, the maximum of the $K$ probabilities $\xi_k[\,i\,]$ with the desirable maximum probability of error $\varepsilon$, and repeat the process if that maximum is greater than $\varepsilon$. This process can end, for example, when $\max\{\xi_1[\,i\,], \dots, \xi_K[\,i\,]\} < \varepsilon$ for the first time,[5] and the total number of iterations until will be the estimate of the stopping time. The Algorithm 1 below shows in pseudo-code the process just described.

**Algorithm 1** – Off-line estimation of the stopping time.

```
Inputs: $\mu_k$, $\sigma^2_{v_k}$, $\boldsymbol{u}_k[\,i\,]$, $\{\mathbb{P}(H_n)\}_{n=1}^{N}$, $\boldsymbol{\theta}_n$, $\widehat{\boldsymbol{Q}}$, $\boldsymbol{b}$, $\boldsymbol{\Sigma}_{\boldsymbol{w}}$, $\mathcal{A}$, $\varepsilon$, $i_{\max}$, $L$.
Initialize $i = 0$, $i_{\text{stop}} = 0$, all matrices and vectors with zero entries.
while $i < i_{\max}$ do
    $\boldsymbol{H}_k \leftarrow \mu_k \boldsymbol{u}_k[\,i\,]\boldsymbol{u}_k^{\top}[\,i\,]$ for $k = 1, \dots, K$.
    $\boldsymbol{Y}_k \leftarrow \boldsymbol{I}_D - \boldsymbol{H}_k$ for $k = 1, \dots, K$.
    $\boldsymbol{D}_k \leftarrow \mu_k \sigma^2_{v_k} \boldsymbol{H}_k$ for $k = 1, \dots, K$.
    $\boldsymbol{H} \leftarrow \text{col}\{\boldsymbol{H}_1, \dots, \boldsymbol{H}_K\}$, $\mathcal{Y} \leftarrow \text{diag}\{\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_K\}$, $\mathcal{D} \leftarrow \text{diag}\{\boldsymbol{D}_1, \dots, \boldsymbol{D}_K\}$.
    $\mathcal{P} \leftarrow \mathcal{A}(\boldsymbol{H} + \mathcal{Y}\mathcal{P})$.            (5.23)
    $\mathcal{Z} \leftarrow \mathcal{A}(\mathcal{Y}\mathcal{Z}\mathcal{Y} + \mathcal{D})\mathcal{A}^{\top}$.      (5.24)
    $\mathcal{S} \leftarrow \mathcal{P}\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{P}^{\top} + \mathcal{Z}$.            (3.23)
    $\boldsymbol{\alpha}_{n,k} \leftarrow \boldsymbol{P}_k(\boldsymbol{\theta}_n - \boldsymbol{b})$ for $k = 1, \dots, K; n = 1, \dots, N$
    $\xi_k \leftarrow \text{MC}\big(\widehat{\boldsymbol{Q}}, \boldsymbol{S}_k, \{\boldsymbol{\alpha}_{n,k}\}_{n=1}^{N}, \{\mathbb{P}(H_n)\}_{n=1}^{N}, L\big)$ for $k = 1, \dots, K$
    if $\max\{\xi_1, \dots, \xi_K\} < \varepsilon$ then
        $i_{\text{stop}} \leftarrow i$
        break
    end if
    $i \leftarrow i + 1$
end while
$I_{\text{stop}}$ ←i
```

It is important to highlight that the desired maximum probability of error $\varepsilon$ cannot be, evidently, smaller than the *lower bound* of the probability of error of a certain network setting; an appropriate description of how this lower bound can be estimated is given in Appendix E.

### 5.3.2 Tracking changes in the state of nature

As discussed in Chapter 1, this thesis does not cover the problem of detecting, as quickly as possible, a transition between states. However, it was mentioned that a simple periodic reset strategy could enable the tracking of this changes in state adequately for the purposes of this

[5] The criterion to stop the algorithm can be different if we want, for example, that the maximum probability of error is *consistently* inferior than $\varepsilon$. For instance, we could estimate the stopping time as the iteration $i$ such that it is the first time that $\max\{\xi_1[\,i\,], \dots, \xi_K[\,i\,]\} < \varepsilon$ uninterruptedly in a sequence of ten iterations.

work. Conveniently, with the estimate $I_{\text{stop}}$, we can devise an strategy that allows the algorithm to track eventual changes on the state of the environment. The proposed detector, as showed, has better performance as the step size $\mu$ is smaller; therefore, we would naturally favor slower adaptation. However, if a change occurs during the process of diffusion, higher values of the step size would be needed to quickly respond to such a change. Instead of falling back to the trade-off we discussed in the Chapter 1, we can reset the algorithm periodically, whenever it reaches the number $I_{\text{stop}}$ of iterations. This guarantees that any change in state is correctly tracked as the algorithm resets and we can keep the step size as small as we desire.[6] A scenario with a change in state during the operation of the proposed detector is simulated and the results are presented in Chapter 6, Figure 13. Algorithm 2 below encapsulates all we have been discussing so far.

**Algorithm 2** – Low-complexity dLMS detection algorithm with stopping time and resetting.

```
Inputs: $\mu_k, d_k[i], \boldsymbol{u}_k[i], \gamma_{mn}, \boldsymbol{\theta}_n, \boldsymbol{b}, \boldsymbol{\Sigma}_{\boldsymbol{w}}, \boldsymbol{A}, \boldsymbol{q}_n$ (5.9), $I_{\text{stop}}$
For each sensor node $k = 1, \dots, K$ do:
while TRUE do
    $i \leftarrow 0, \hat{\boldsymbol{w}}_k \leftarrow \boldsymbol{b}$ (see Section 5.2)
    while $i < I_{\text{stop}}$ do
        $\hat{\boldsymbol{w}}_k \leftarrow \text{dLMS}(\hat{\boldsymbol{w}}_k, \mu_k, d_k[i], \boldsymbol{u}_k[i], \boldsymbol{A})$
        $t''_{k,n} \leftarrow (\hat{\boldsymbol{w}}_k - \boldsymbol{b})^{\top}\boldsymbol{q}_n$ for $n = 1, \dots, N$    (5.10)
        Decide $H_n$ such that $t''_{k,n} > t''_{k,m} + (i+1)\mu\gamma_{mn}\ \forall m, n$
    end while
end while
```

[6] To properly track the change in state, the time between two consecutive changes must be larger than $I_{\text{stop}}$ by a sufficient amount.

# 6 SIMULATIONS

In this chapter, we see the simulated performance of the low-complexity detector of Chapter 5 (called “LCD” for short in this chapter), comparing it with the theoretical detector from Chapter 3 (called “HCD” for “high-complexity detector”) and with the optimal detector under the same conditions. The performances are tested under the worst-case connectivity (an open ring) to show the real potential of the proposed detector. Additional simulations are also presented, such as the effect of the number of Monte Carlo trials on the bias of the maximum probability of error, and a sudden change of state and the application of the resetting strategy, discussed in Subsection 5.3.2. We also see simulations showing the effect of varying the initial estimate and a discussion about an alternative choice to the one recommended in Proposition 13 that can improve the detection performance. Finally, the chapter concludes with simulations showing the effect of the conditioning number of the covariance matrix of the state vector $\boldsymbol{w}$.

## 6.1 Simulation 1: network with 10 nodes

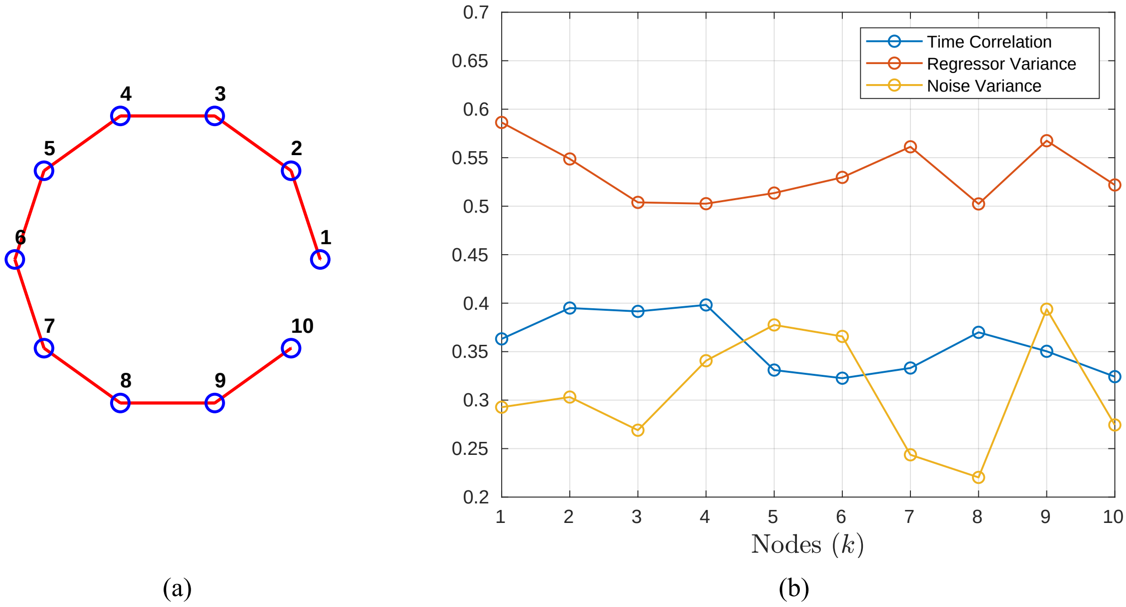


Figure 6 – Topology and network parameters used in Simulation 1.

*Note*: (a) Open ring topology. (b) Values of network parameters for each node.

In order to illustrate the results in this thesis, first a distributed detector is simulated in a network composed of $K = 10$ nodes, connected in a form of a open ring (see Figure 6a); i.e., the least connected topology possible, chosen as such to simulate a worst case scenario. Each node must decide among $N = 6$ hypotheses about the environment state; thus, we have the following possibilities for the vector mean $\boldsymbol{\theta}_n \in \mathbb{R}^5$, arranged column-wise:

$$\begin{bmatrix}\boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \boldsymbol{\theta}_3 & \boldsymbol{\theta}_4 & \boldsymbol{\theta}_5 & \boldsymbol{\theta}_6\end{bmatrix} =$$

$$\begin{bmatrix} 0.16317 & 1.22914 & 0.95917 & 0.76734 & -0.33855 & 0.45338 \\ 0.62918 & -0.36314 & -0.20530 & 0.54310 & 1.84345 & 2.31907 \\ 1.31436 & 0.95557 & 2.05600 & 1.25648 & 1.04744 & 0.90886 \\ 0.31290 & 0.68000 & 0.43832 & 0.95900 & 0.58317 & -0.29710 \\ 0.27186 & 1.73930 & 1.73293 & 1.58458 & 0.95732 & 1.05079 \end{bmatrix}.$$

To obtain a positive definite covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$, we generated a matrix $\boldsymbol{M}$ whose elements were each randomly selected form a normal random variable in the interval with zero mean and standard deviation 1, and calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = \sigma_{\boldsymbol{w}}^2(\boldsymbol{M}\boldsymbol{M}^\top + D\boldsymbol{I}_D),\ \boldsymbol{M} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D) \tag{6.1}$$

where $\sigma_{\boldsymbol{w}}^2 = 10^{-2}$ is a constant. The term $D\boldsymbol{I}_D$ guarantees a covariance matrix with a low conditioning number (we discuss this issue further in Section 6.4). The following prior probabilities were also randomly selected, using a uniform distribution in the interval $[0, 1]$ and appropriately normalized to sum to one:

$$\left\{\mathbb{P}(H_1) \quad \mathbb{P}(H_2) \quad \mathbb{P}(H_3) \quad \mathbb{P}(H_4) \quad \mathbb{P}(H_5) \quad \mathbb{P}(H_6)\right\} =$$
$$\left\{0.04753 \quad 0.18643 \quad 0.24122 \quad 0.26087 \quad 0.03105 \quad 0.23289\right\}.$$

To obtain regressors $\boldsymbol{u}_k[\,i\,] \in \mathbb{R}^5$, modeled as realizations of $\underset{\sim}{\boldsymbol{u}}_k[\,i\,] \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}_k}\big)$ with full covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{u}_k}$, we generate $K = 10$ signals using first order Markov processes with power $\sigma_{\boldsymbol{u}_k}^2$ and time correlation $\alpha_k$. The expression for the first-order auto-regressive filter, for each node $k$, is given as[1]

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

The noise in the measurements was modeled as white Gaussian noise with power $\sigma_{v_k}^2$. The values of $\sigma_{\boldsymbol{u}_k}^2$, $\alpha_k$ and $\sigma_{v_k}^2$ were obtained randomly and can be seen in the graph in Figure 6b. The weights $a_{\ell k}$ in Equation (3.3) are obtained using the Metropolis rule, which is given as [103]

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

in which $n_k$ and $n_\ell$ are the number of connections of nodes $k$ and $\ell$. This composition results in $\boldsymbol{A} = (a_{\ell k})$ being a double stochastic matrix, although this is not a necessary condition. The initial estimate $\boldsymbol{b} = \boldsymbol{w}[-1]$ was obtained by solving the corresponding system in Equation (5.13) with $N - 1 = 5$ equations and $D = 5$ unknowns.

[1] First-order auto-regressive models are usually used, among other applications, to model certain communication channels and to generate signals with known correlation properties in simulations of adaptive filtering algorithms [98, 102]. We used them in order to avoid overly simple uncorrelated regressors.

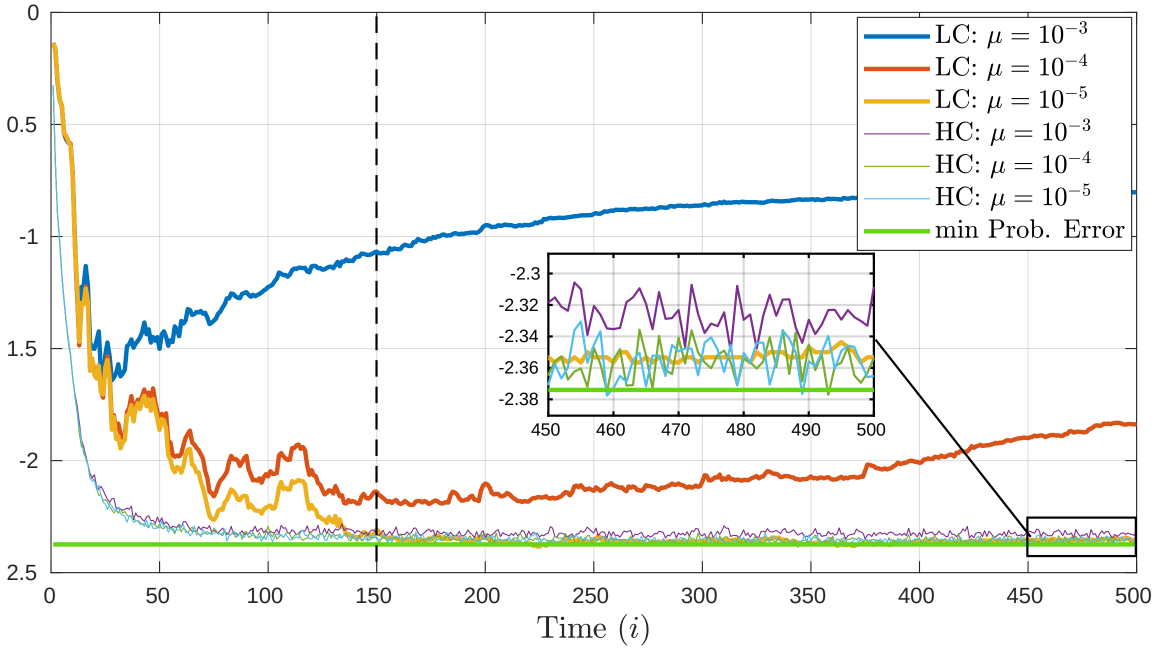


Figure 7 – $\log_{10}(\xi_{\text{max}}[\,i\,])$ for different values of the step size for Simulation 1.

*Note*: The figure shows the log-scaled maximum probabilities of error of the dLMS Low-Complexity Detector (LCD, thick curves), and the ideal High-Complexity Detector (HCD, thin curves). We can see the performance for three different values of $\mu$: $10^{-3}$, $10^{-4}$ and $10^{-5}$, and also the asymptotic performance of the optimal detector described in Equation (4.37) (green curve).

In Figure 7, we can see the log of the maximum probability of error among the $K = 10$ nodes in the network for two different detectors: the Low Complexity Detector (LCD, thick curves) which is the detector we developed in Chapter 5, Equation (5.10), and what we call High Complexity Detector (HCD, thin curves), which simulates the performance of an ideal detector for the dLMS as presented in Chapter 3 and as given in Equation (5.1); i.e., it is calculated without taking any approximation. For both detectors, we estimated the probabilities of error using Monte Carlo simulations with $10^4$ realizations. We simulated these two detectors for 3 different values for $\mu$:

$$\mu \in \{10^{-3},\, 10^{-4},\, 10^{-5}\}.$$

As we can see from the results in Figure 7, for $\mu = 10^{-3}$, the LCD (blue curve) does not show a good performance, since it has the highest probability of error of all the other possible configurations. However, as we reduce $\mu$ tenfold, the performance improves (red curve), as expected from our previous discussion about how a smaller value of $\mu$ leads to better performance, since in this way we better approximate the performance of the theoretical HCD. Nevertheless, the LCD does not reach the performance of the HCD, since the former does not attain as low probabilities of error as the latter; also, as time $i$ increases, the performance worsens: this is due to the effect of high order terms of $\mathscr{P}[\,i\,]$ and $\mathscr{Z}[\,i\,]$ (see equations (4.13) and (4.20)). In more precise words, the value of $\mu$ does not provide a $I_{\text{sup}}$ sufficiently large for the approximations of Section 4.4 to be valid. But, for $\mu = 10^{-5}$ (yellow curve), we can see that the performance of the

LCD becomes virtually the same as the HCD performance for sufficient large $i$. Therefore, as we said earlier, as we reduce the value of the step size, we extend the time period in which our approximations are valid, and simultaneously we obtain a performance that approaches that of the HCD. Note that the LCD still lags the HCD in the initial iterations ($i <\sim 150$) as an effect of approximating matrices $\boldsymbol{Q}_k[\,i\,]$ and $\boldsymbol{R}_k[\,i\,]$ in Equation (5.8) and also $[\boldsymbol{A}^i]_{\ell k} \approx \boldsymbol{\pi}_\ell$, since both require a sufficiently large $i$ (more precisely, $i > I_{\text{inf}}$), as argued in sections 4.4 and 5.1. However, for $i > 150$, the performance is close to the minimum probability of error possible in this network setting (green line), calculated by simulating the asymptotic performance of an optimal detector (see Appendix E). This fact confirms what we argued in Section 4.4: the HCD approximates the optimal detector given in Equation (4.37) as $\mu$ is chosen smaller, despite the fact that the network is poorly connected, which indicates the robustness of our approximations. Furthermore, since the LCD matches the HCD, and despite the small values of $\mu$, the low-complexity detector can also closely match the optimal detector.

## 6.2 Simulation 2: network with 20 nodes

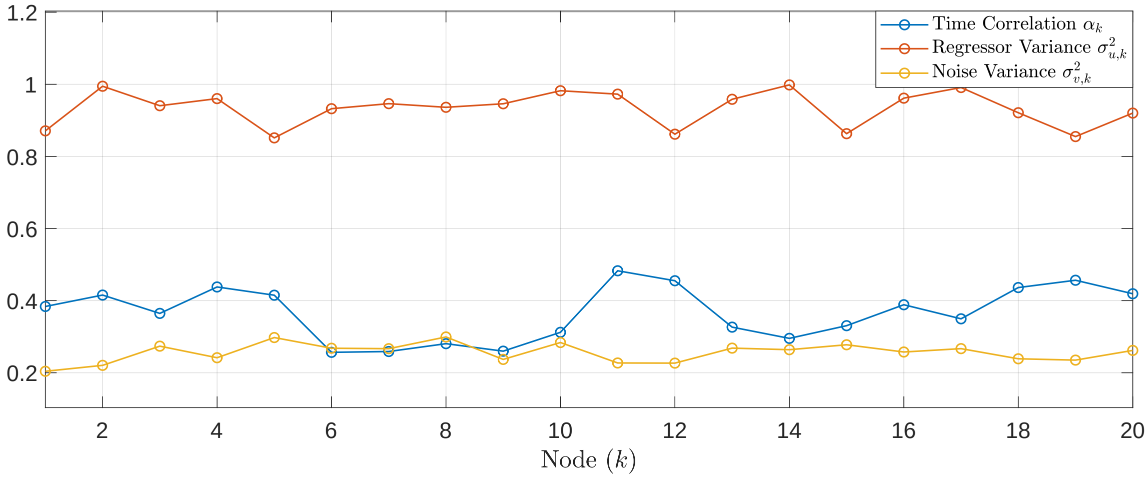


Figure 8 – Values of network parameters for Simulation 2.

A new simulation was conducted using $K = 20$ nodes in the open ring topology as in Simulation 1. In this new setting the value of $\sigma^2_{\boldsymbol{w}}$ in Equation (6.1) was changed to $1.9 \cdot 10^{-4}$, and also those of $\sigma^2_{v_k}$, $\sigma^2_{\boldsymbol{u}_k}$ and $\alpha_k$ (see Figure 8). This simulation used $10^5$ Monte Carlo trials to estimate the probabilities of error for both HCD and LCD, using $4$ different values for $\mu$:

$$\mu \in \{10^{-3},\, 3 \cdot 10^{-4},\, 3 \cdot 10^{-5},\, 3 \cdot 10^{-6}\}.$$

Also, it includes the performance of the optimal detector for each $i$—details of how it can be estimated can be seen in Appendix E. This time, we have the following $N = 6$ possibilities for the vector mean $\boldsymbol{\theta}_n \in \mathbb{R}^5$, arranged column-wise:

$$\begin{bmatrix}\boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \boldsymbol{\theta}_3 & \boldsymbol{\theta}_4 & \boldsymbol{\theta}_5 & \boldsymbol{\theta}_6\end{bmatrix} =$$

$$\begin{bmatrix} 0.41025 & 0.38319 & 0.91008 & 0.51673 & 0.55084 & 0.57463 \\ 1.09786 & 0.90515 & 1.13891 & 1.17009 & 1.22394 & 1.16104 \\ 0.30546 & 0.26871 & 0.45478 & 0.63032 & 0.32843 & 0.04282 \\ -0.07999 & 0.06362 & -0.43254 & -0.15304 & 0.14816 & 0.28056 \\ 0.09365 & 0.13203 & -0.27248 & -0.28267 & -0.46275 & -0.51267 \end{bmatrix}.$$

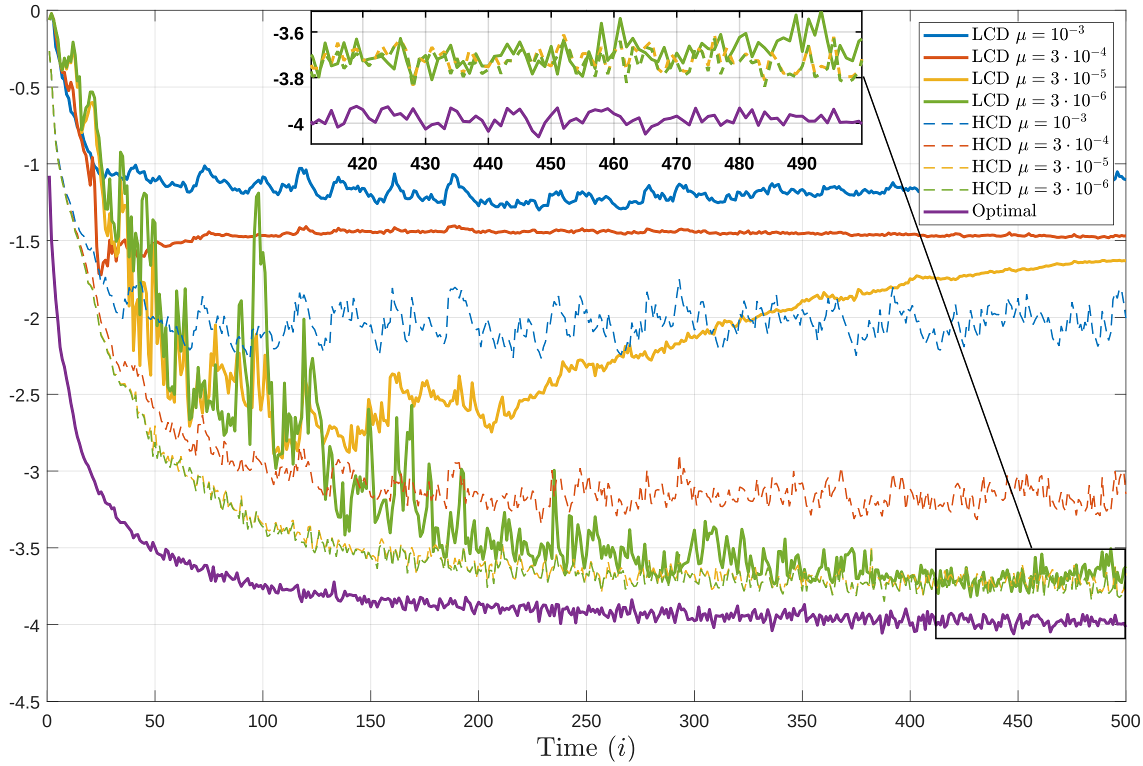


Figure 9 – $\log_{10}(\xi_{\max}[\,i\,])$ for different values of the step size for Simulation 2.

*Note*: The figure shows the log-scaled maximum probabilities of error of LCD (thick curves) and HCD (thin dashed curves) in a 20-nodes open ring topology for $\mu \in \{10^{-3}, 3 \cdot 10^{-4}, 3 \cdot 10^{-5}, 3 \cdot 10^{-6}\}$, and also the performance of the optimal detector described in Equation (4.37) (purple curve).

As we examine the curves in Figure 9, we observe that the HCD and the LCD exhibit similar behavior in terms of the maximum probability of error. Some noteworthy observations can be made from these results. Firstly, for relatively large values of $\mu$, the HCD reaches a certain level of probability of error and seems stationary. This happens because large values of $\mu$ allow the dLMS filters to reach the steady state in estimation. As we discussed earlier, the HCD approximates the optimal performance when kept in a transient state by means of small values of $\mu$. This phenomenon was not apparent in Simulation 1 due the relatively high value of $\sigma_{\boldsymbol{w}}^2$, since it sets the general level of "uncertainty" about the state vector $\boldsymbol{w}$ (see Equation (6.1)). As shown in Appendix E, the asymptotic probability of the optimal detector significantly depends on

$\boldsymbol{\Sigma}_{\boldsymbol{w}}$, and larger entries in $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ lead to worse asymptotic performance due to a more challenging discrimination among the possible $\boldsymbol{w}$. This implies that as $\sigma_{\boldsymbol{w}}^2$ increases, the asymptotic error probability also increases, and if this lower bound is sufficiently high, we may not observe the steady state effect of the dLMS estimator on the HCD probability of error. However, the use of substantially smaller $\sigma_{\boldsymbol{w}}^2$ in this simulation allowed us to observe this phenomenon and also note that the steady-state probability of error becomes smaller with smaller values of $\mu$.

Additionally, comparing with the results in Figure 7, the maximum probabilities of error in Figure 9 appear to reach a state where the difference with respect to the optimal performance is notably larger. Two reasons contribute to this observation. Firstly, the open ring topology poses challenges in the dissemination of information among nodes as the number of sensors increases. Recall that a condition for the HCD to approximate the optimal performance is that the network is not significantly different from a totally connected topology after several iterations of the dLMS. However, in an open ring topology, a node with the worst performance among all receives less information from other nodes, which could improve its decision quality. Consequently, the maximum probability of error tends to increase in this scenario. However, Figure 10 reveals that the average probability of error

$$\bar{\xi}[i] = \frac{1}{K}\sum_{k=1}^{K} \xi_k[i]$$

is not significantly different from the optimal detector, suggesting that, on average, the open ring can still be approximated as a totally connected network.

Another factor contributing to the greater difference between the maximum probability of error and the optimal performance for larger open rings is the bias introduced by the finite number of Monte Carlo trials. As the number of nodes increases, the likelihood of a node performing exceptionally poorly in the trials increases. Consequently, when selecting the maxima for each iteration, these extreme performances contribute to the increased discrepancy. To validate this observation, we repeated Simulation 2 for the LCD with $\mu = 3 \cdot 10^{-6}$ using $L = 10^6$ realizations of Monte Carlo trials, a tenfold increase in the number of realizations. As shown in Figure 11, the performance appears to improve for lower probabilities with more realizations. On the other hand, Figure 12, comparing the mean probability of error for the same simulations, demonstrates that the performance does not change significantly in the same region of lower probabilities. This leads us to conclude that the bias incurred by the number of Monte Carlo trials impacts the maximum probabilities more significantly than the mean probabilities.

Finally, to demonstrate the effectiveness of the reset strategy discussed in Section 5.3, a change in the active hypotheses (state of nature) was insert during the simulation at iteration $i = 200$. Figure 13 displays the performance curves of the LCD, showing a significant impact on the performance of the detector after the change, with the maximum probability of error remaining high as time progresses. On the other hand, by applying the reset strategy at a specific point (in the simulation, at $i = I_{\text{stop}} = 250$), the performance of the detector behaves similarly as the initial iterations, exhibiting the characteristic decay in the maximum probability of error and

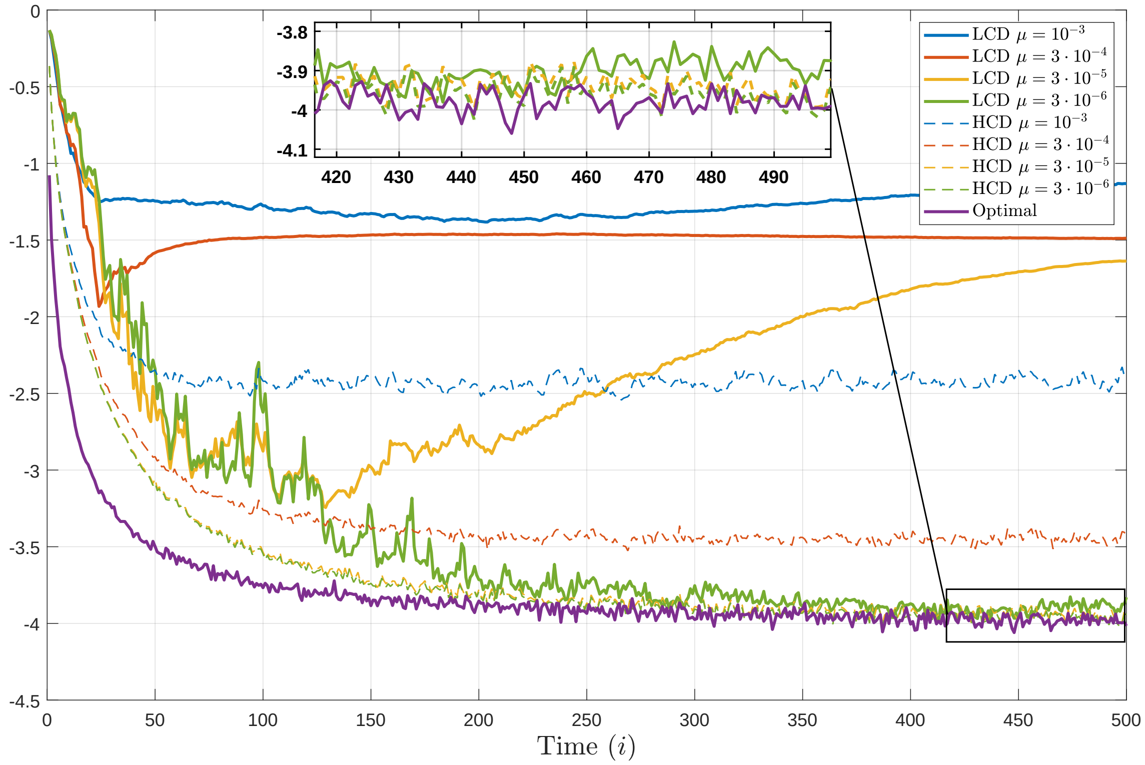


Figure 10 – $\log_{10}(\bar{\xi}[\,i\,])$ for different values of the step size for Simulation 2.

*Note*: The figure shows the log-scaled average probabilities of error of LCD (thick curves) and HCD (thin dashed curves) in a 20-nodes open ring topology for $\mu \in \{10^{-3},\ 3 \cdot 10^{-4},\ 3 \cdot 10^{-5},\ 3 \cdot 10^{-6}\}$, and also the performance of the optimal detector described in Equation (4.37) (purple curve).

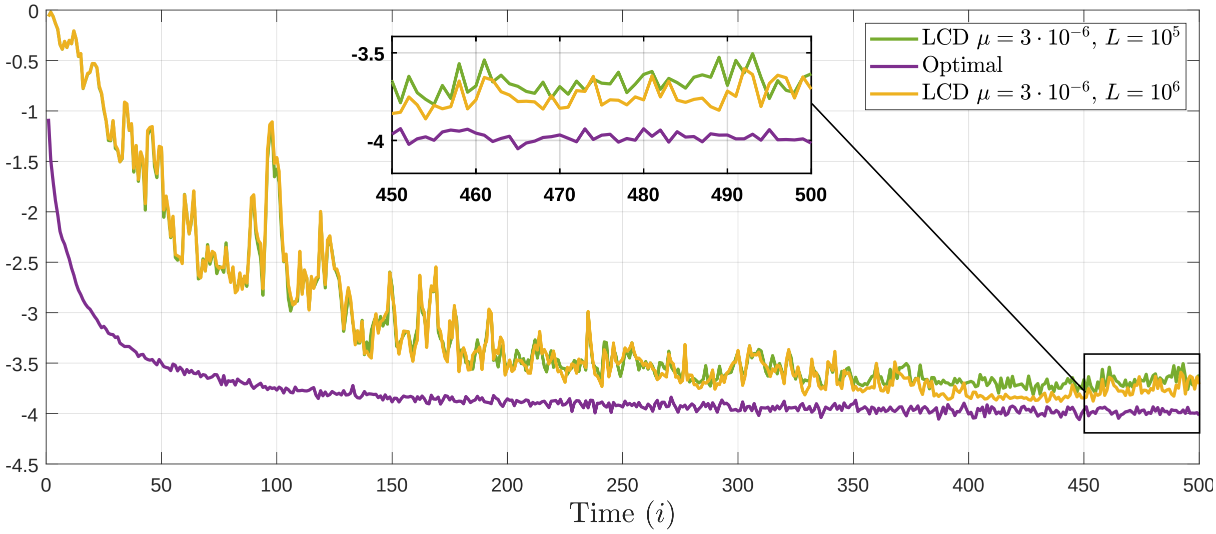


Figure 11 – $\log_{10}(\xi_{\max}[\,i\,])$ varying the number of Monte Carlo trials.

*Note*: The figure shows the log-scaled maximum probabilities of error of LCD (thick curves) for $\mu = 3 \cdot 10^{-6}\}$ for $10^{5}$ and $10^{6}$ realizations of Monte Carlo trials to estimate the probabilities, and also the performance of the optimal detector described in Equation (4.37) (purple curve).

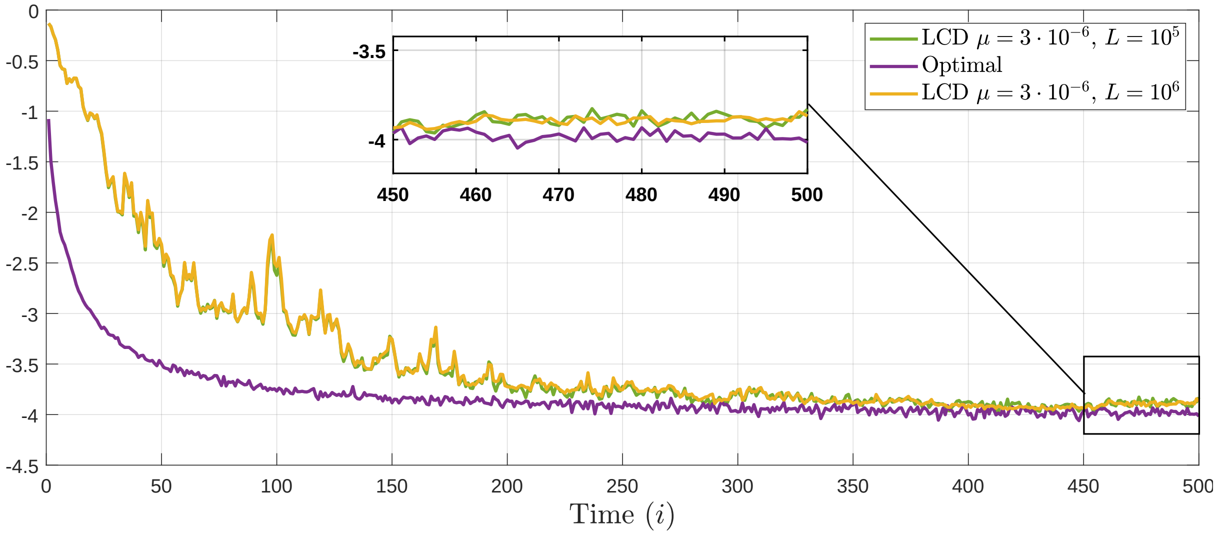


Figure 12 – $\log_{10}(\bar{\xi}[\,i\,])$ varying the number of Monte Carlo trials.

*Note*: The figure shows the log-scaled average probabilities of error of LCD (thick curves) for $\mu = 3 \cdot 10^{-6}\}$ for $10^5$ and $10^6$ realizations of Monte Carlo trials to estimate the probabilities, and also the performance of the optimal detector described in Equation (4.37) (purple curve).

regaining the previous level of performance. Therefore, this indicates that a simple reset strategy for the detection algorithm can effectively track eventual changes in the state of nature.

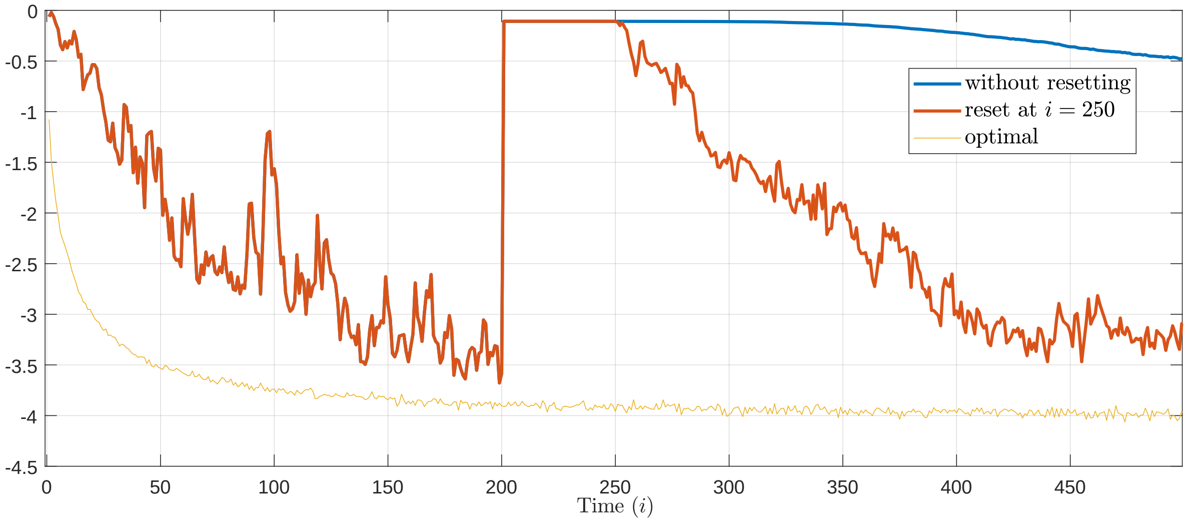


Figure 13 – $\log_{10}(\xi_{\max}[\,i\,])$ when resetting the detector after a change in the state of nature

*Note*: The figure shows the log-scaled maximum probabilities of error of LCD, showing the performance when the detection algorithm is reset ($i = I_{\text{stop}} = 250$) after a change in the state of nature at $i = 200$ and when there is no reset.

## 6.3 Simulation 3: varying the initial estimate $\boldsymbol{b}$

As we discussed in Section 5.2, when the dimension $D$ of the state vector $\boldsymbol{w}_n$ and the number of hypotheses $N$ are such that $N-1 < D$, there can be infinite choices for the initial estimate $\boldsymbol{b}$ that solves the linear system in Equation (5.13). As stated in Proposition 13, a reasonable choice for $\boldsymbol{b}$ is

$$\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}} \|\,\boldsymbol{b}\,\|_2\,,$$
$$\text{subject to: } (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{b}) = (\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\theta}_m - \boldsymbol{b}),\ \forall m \neq n. \tag{6.2}$$

To show the effect of choosing different values of $\boldsymbol{b}$ with different norms, the same open ring network with $K = 20$ nodes and the same network parameters given in Figure 8 was simulated, with $N = D = 6$ and the following values of the parameters:

$$\begin{bmatrix}\boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \boldsymbol{\theta}_3 & \boldsymbol{\theta}_4 & \boldsymbol{\theta}_5 & \boldsymbol{\theta}_6\end{bmatrix} =$$
$$\begin{bmatrix} -0.11949 & 0.11899 & -0.06096 & -0.08742 & 0.08657 & -0.47078 \\ -0.50881 & -0.04039 & -0.13411 & 0.15146 & -0.53324 & -0.29274 \\ 0.80704 & 0.95676 & 0.73766 & 0.64996 & 1.11694 & 1.14088 \\ 0.37673 & 0.16697 & 0.33704 & 0.34582 & 0.53080 & 0.53003 \\ 0.82207 & 0.62328 & 0.57793 & 0.78833 & 0.36003 & 0.89798 \\ 0.90127 & 0.42503 & 0.89978 & 0.27165 & 0.59004 & 0.24073 \end{bmatrix}.$$

The step size $\mu$ was set at $3 \cdot 10^{-6}$; all the other parameter settings were kept as presented in Section 6.2. The initial estimate $\boldsymbol{b}$ was chosen as follows:

$$\boldsymbol{b} = \boldsymbol{b}^* + \alpha\boldsymbol{n} \text{ such that } \begin{bmatrix}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_n)^\top \\ \vdots \\ (\boldsymbol{\theta}_N - \boldsymbol{\theta}_n)^\top\end{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\boldsymbol{n} = \boldsymbol{0}, \alpha \in \mathbb{R}. \tag{6.3}$$

in other words, $\boldsymbol{n}$ is in the null space (kernel) of linear system in Equation (5.13), whose basis is given as

$$\left\{\boldsymbol{n} \,\middle|\, \boldsymbol{n} = \begin{bmatrix}-0.38005 & 0.32875 & 0.28043 & 0.64025 & -0.46097 & 0.21547\end{bmatrix}^\top\right\}$$

(note that the null space has dimension 1 since $N = D$). And the initial estimate with least norm is

$$\boldsymbol{b}^* = \begin{bmatrix}-0.61067 & -0.36597 & 0.74963 & 0.09063 & 0.27763 & 0.33521\end{bmatrix}^\top.$$

We can see in Figure 14 the different detection performances of the LCD in terms of the maximum probability of error in the network for $\alpha \in \{-1, -0.2, -0.1, 0, 0.1, 1\}$. The corresponding resulting norms of $\boldsymbol{b}^* + \alpha\boldsymbol{n}$ are

$$\left\{1.50548,\ \ 1.14301,\ \ 1.12981,\ \ 1.12537,\ \ 1.12981,\ \ 1.50548\right\}.$$

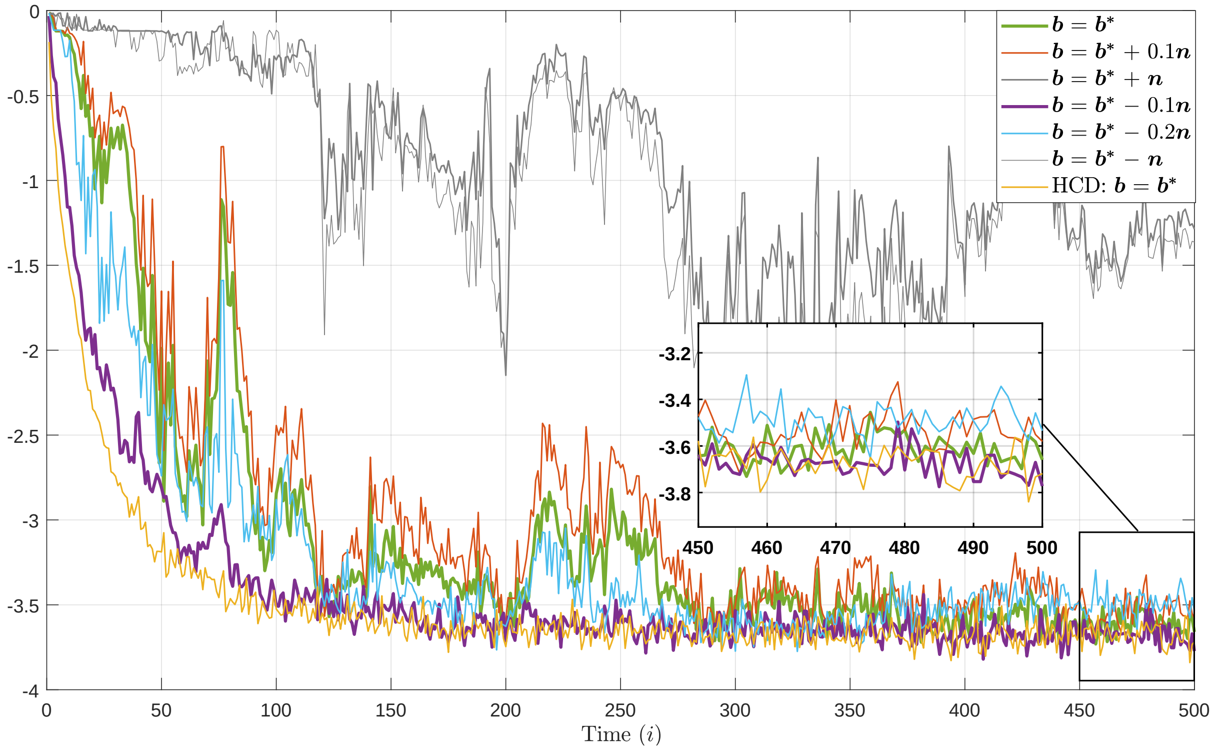


Figure 14 – $\log_{10}(\xi_{\max}[\,i\,])$ for different initial estimates for Simulation 3.

*Note*: The figure shows the log-scaled maximum probabilities of error of LCD in a 20-nodes open ring topology for $\mu = 3 \cdot 10^{-6}$, using different values for the initial estimate $\boldsymbol{b}$ as described in Equation (6.3). The performance of the HCD with $\boldsymbol{b} = \boldsymbol{b}^*$ was included for reference.

The performance of the HCD with $\boldsymbol{b} = \boldsymbol{b}^*$ was included for reference, but the reader must recall that the HCD does not have a performance that changes with the choice of $\boldsymbol{b}$.

As we examine the performance curves in Figure 15, it is evident that the worst detector performances correspond to $\boldsymbol{b}$ with larger norms (thinner gray curves), while the best performances correspond to smaller norms, consistent with our discussion in Section 5.2. However, interestingly, the best performance is not achieved by the detector with $\boldsymbol{b} = \boldsymbol{b}^*$ (thick green curve) having the least norm, but rather by choosing $\boldsymbol{b} = \boldsymbol{b}^* - 0.1\boldsymbol{n}$ (thick purple curve). This phenomenon is not surprising, as the choice of $\boldsymbol{b}$ that minimizes the *maximum* quadratic detection error due to approximations, as defined in Equation (5.16), may not necessarily minimize the quadratic error *per se*. In fact, finding the best initial value for the detection problem in this thesis remains an open matter, except for the special case when $N = 2$ and $p(H_1) = p(H_2)$, as shown in Section 5.2. In this case, $\bar{\boldsymbol{\theta}}_{12} = \frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2}{2}$ effectively minimizes the quadratic detection error due to approximations.

To explore this further, we considered the strategy of choosing an initial estimate $\boldsymbol{b}$ that minimizes the distance of $\boldsymbol{b}$ from the *mean value* of the expected vector means values, i.e.

$$\bar{\boldsymbol{\theta}} = \frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2 + \cdots + \boldsymbol{\theta}_N}{N}.$$

This choice was motivated by the observation that sometimes the overall detection performance

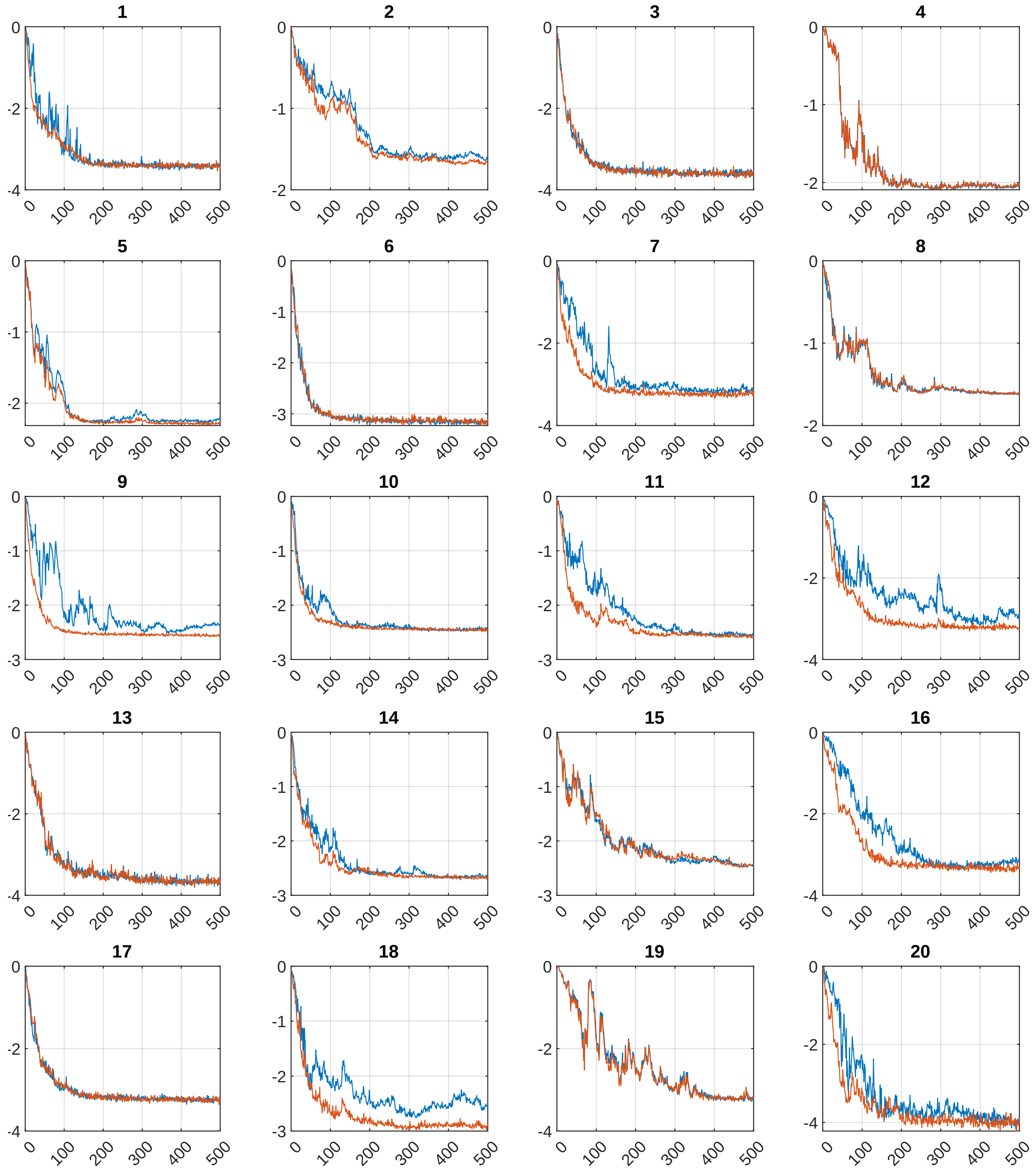


Figure 15 – $\log_{10}(\xi_{\max}[\,i\,])$ for two different strategies for choosing the initial estimate

*Note*: The figure shows a sequence of 20 different simulations comparing the maximum probability of error of two initialization strategies: *least norm* Equation (6.2) and *closest to the mean* Equation (6.4), and varying $\sigma^2_{v_k}$, $\sigma^2_{\boldsymbol{u}_k}$, $\alpha_k$, $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$, and $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ each time, keeping the $K = 20$ nodes in an open ring topology and all the other network settings of the simulation in Section 6.2. In each simulation, the blue curve represents the performance of the LCD for the strategy in Equation (6.2) (least norm), and the red one represents the performance for the strategy in Equation (6.4) (closest to the mean).

can be improved by selecting an initial estimate that is closer to the mean value. To formalize this approach, we choose $\boldsymbol{b}^*$ as the solution to the following optimization problem:

$$\begin{gathered}\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}} \left\| \boldsymbol{b} - \bar{\boldsymbol{\theta}} \right\|_2, \\ \text{subject to: } (\boldsymbol{\theta}_n - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_n - \boldsymbol{b}) = (\boldsymbol{\theta}_m - \boldsymbol{b})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} (\boldsymbol{\theta}_m - \boldsymbol{b}),\ \forall m \neq n. \end{gathered} \tag{6.4}$$

To evaluate the effectiveness of this strategy, we conducted a sequence of 20 different simulations, comparing the performance in terms of the maximum probability of error, of the *least norm strategy* in Equation (6.2) and the *closest to the mean strategy* in Equation (6.4). Each simulation varied the network parameters $\sigma^2_{v_k}$, $\sigma^2_{\boldsymbol{u}_k}$, and $\alpha_k$, the vector means $\{\boldsymbol{\theta}_n\}_{n=1}^N$, and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$, while maintaining the $K = 20$ nodes in an open ring topology and all other network settings as in Simulation 2. In each simulation result, the blue curve represents the performance of the LCD for $\boldsymbol{b}$ chosen with the least norm, and the red curve represents the performance for $\boldsymbol{b}$ chosen as the closest to the mean.

The results in Figure 15 demonstrate that there are instances where the *closest to the mean strategy* outperforms the *least norm strategy* for the initial iterations of the algorithm (results 1, 2, 5, 7, 9, 10, 11, 12, 14, 16, 18, and 20) In other instances, there is no significant difference in performance between the two strategies. This intriguing finding suggests that there may be an interplay between these two constraints on the choice of $\boldsymbol{b}$, warranting further investigation in future research.

The choice of $\boldsymbol{b}$ with the least norm was initially intuitive, as argued in Section 5.2. On the other hand, the *closest to the mean strategy* has an intuitive interpretation, as it forces the initial estimate to be close to the potential “answers” to the detection problem (i.e., the expected values of the current $\boldsymbol{w}_n$). This approach helps prevent any of the norms $\| \boldsymbol{b} - \boldsymbol{\theta}_n \|_2$, $1 \leq n \leq N$, from being excessively large, which could otherwise amplify the detection error due to approximations. While there remains a need for further investigation, these findings provide valuable insights into the impact of different strategies for initializing the detection algorithm.

## 6.4 Simulation 4: the effect of the conditioning number of $\boldsymbol{\Sigma}_{\boldsymbol{w}}$

A high conditioning number[2] of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ can impact significantly the performance of the LCD. To investigate this effect, let us vary the term that controls the conditioning number in Equation (6.1); i.e., let us vary $\alpha$ in

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = \sigma^2_{\boldsymbol{w}} (\boldsymbol{M}\boldsymbol{M}^\top + \alpha \boldsymbol{I}_D),\ \alpha \in \mathbb{R}.$$

For $\alpha = 0,\ 1,\ \dots$ , 5, we have the following conditioning numbers:

$$\left\{ 4.662,\quad 2.968,\quad 2.345,\quad 2.022,\quad 1.824 \right\}.$$

[2] The conditioning number of a matrix is defined as the ratio of the largest to the smallest eigenvalue of the matrix, and provides a measure of its numerical stability.

All the other parameters were kept as used in Simulation 2, with $\mu = 3 \cdot 10^{-6}$ and $\boldsymbol{b} = \boldsymbol{b}^*$ (least norm). As shown in Figure 16, lower values of $\alpha$ (corresponding to higher conditioning numbers) caused the performance of the LCD to deteriorate significantly. This phenomenon can be attributed to the fact that matrices with high conditioning numbers are numerically unstable, in the sense that even tiny deviations resulting from the approximations made in Chapter 5 can be greatly amplified by such matrices, leading to a worsening of performance.

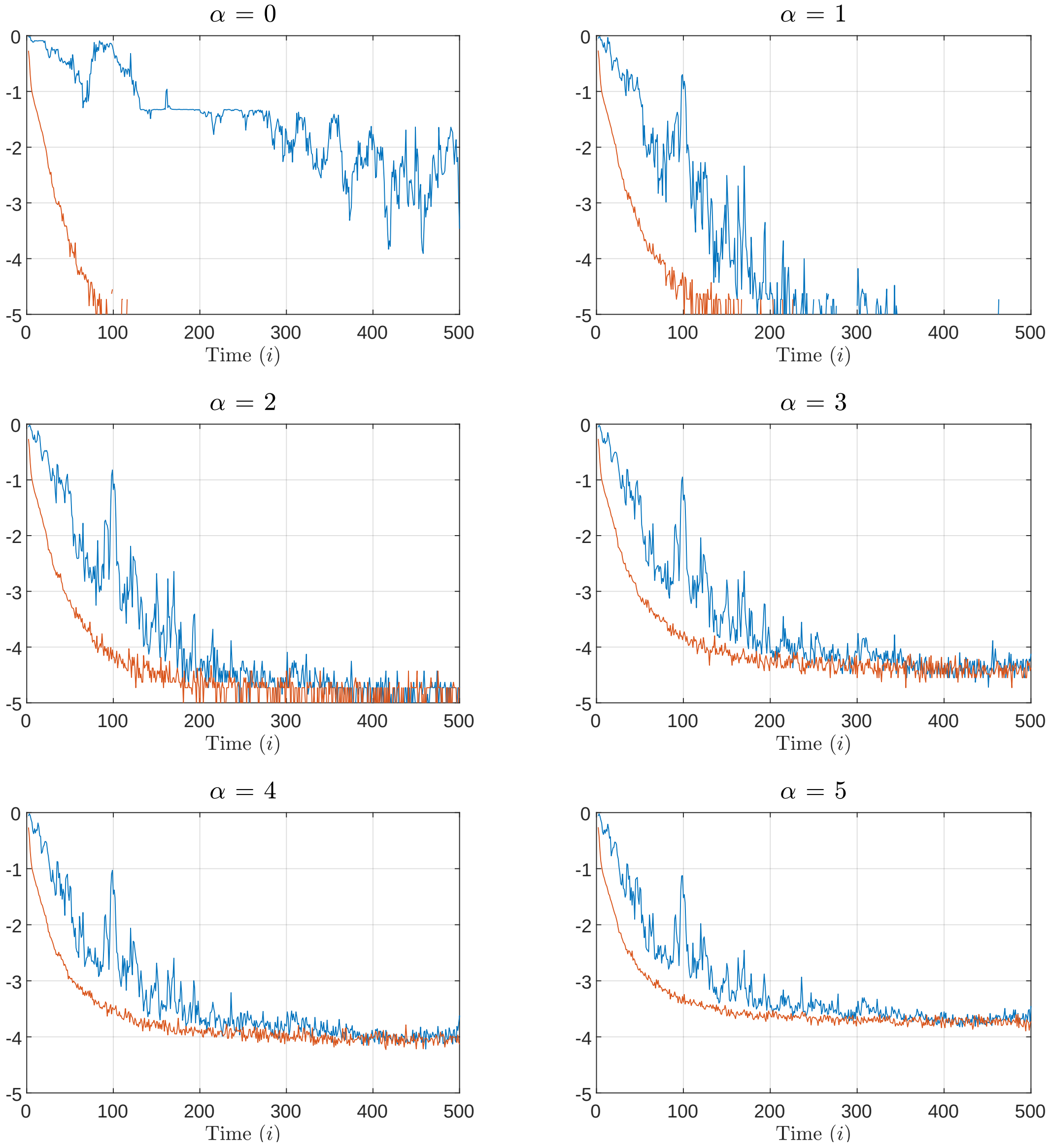


Figure 16 – $\log_{10}(\xi_{\max}[i])$ for different conditioning numbers of $\boldsymbol{\Sigma_w}$ for Simulation 4.

*Note*: The figure shows a sequence of simulations to observe the effect of the conditioning number of the covariance matrix $\boldsymbol{\Sigma_w}$ of the state vector. The blue curves are the maximum probability of error of the LCD, and the red ones that of the HCD. The network setting is the same as Simulation 2, with $\mu = 3 \cdot 10^{-6}$ and $\boldsymbol{b} = \boldsymbol{b}^*$.

Although this effect may appear to impose limitations on the use of the LCD, it is essential to consider that a poorly conditioned $\boldsymbol{\Sigma_w}$ indicates the presence of an eigenvalue close to zero. Consequently, there exists a subspace of $\mathbb{R}^D$ given by the corresponding eigenvector where the state vector process $\underset{\sim}{\boldsymbol{w}}_n$ is nearly deterministic. Therefore, it is possible to incorporate this understanding into the problem modeling and apply some form of dimension reduction to formulate the detection problem in $\mathbb{R}^{D-1}$. By removing the eigenvalue closest to zero from the original covariance matrix, the new matrix in $\mathbb{R}^{D-1}$ will have a lower conditioning number, potentially leading to improved numerical stability and overall performance.

# 7 CONCLUSION, REMARKS AND NEXT STEPS

This doctoral thesis fulfilled its main objective, which was the development of a low-complexity detection algorithm for wireless networks with a detailed theoretical analysis encompassing its most outstanding feature, namely a performance in terms of probability of error that better approximates an optimal detector by means of a slower learning rate of the estimation algorithm at the core of the distributed routine. A similar result for a much simpler setting was achieved during my master's degree, as mentioned in the Introduction, which served as the motivation to this doctoral thesis. Therefore, this result is also an extension to a much more general situation.

The detector's low-complexity, discussed in detail in Chapter 5, makes it suitable for distributed smart sensor networks, such as those employed in IoT networks, where low energy consumption is fundamental to its feasibility and practical use. The estimation algorithm used is the dLMS, which allows for the exploitation of its distributed nature to share information across the network, resulting in a more robust detection routine.

Alongside the steps for the formulation of a feasible and low-complexity detector (LCD), an ideal high complexity dLMS detector (HCD) that does not consider any physical limitations such as energy, time, computational power, etc., was also developed to make performance comparisons and better explain the theoretical results. The detector shows great generality by encompassing detection problems with multiple possible outcomes with any values of the prior probabilities, also considering uncertainties in the state vector which represents the different hypotheses.

The main contribution of this thesis is proving theoretically that this low-complexity detector has a performance that can be made very close to the performance of both the ideal HCD and the optimal detectors by means of smaller choices of the LMS step size, without damaging or penalizing the decay rate of the probability of error even in settings with poorly connected networks. As shown, the performance of the detector $(i)$ is independent of the particular value of the step size when it is sufficiently small and $(ii)$ better approximates the asymptotic performance of the optimal detector during a finite time interval, which can be made as long as desired by smaller choices of the step size; therefore, by slowing the learning speed of the estimation algorithm. In other words, this thesis showed that, at least in this so unexpected a scenario, the Tortoise really beats the Hare as Aesop so aptly ventured to say in his classic fable.

As discussed, this thesis provides some interesting potential extensions for future work, mainly a further investigation on the effect of the initial estimate on the first iterations of the detection algorithm. In addition to that, other interesting extensions would be $(i)$ the expansion of the results presented here to the classic Neyman-Pearson detection formulation, $(ii)$ the estimation of the probability of error for a class of regressors rather than a specific observed one, $(iii)$ strategies for scenarios where it is not possible to obtain the network statistics such as the noise powers, and possibly $(iv)$ a formulation for non-Gaussian signals. Other possible extensions are $(v)$ the use of algorithm other than the dLMS, $(vi)$ the use of different sets of states of nature for different subsets of the network, and $(vii)$ a study on the effect of varying the length in bits of the data.

# REFERENCES

1 POTTIE, G. J.; KAISER, W. J. Wireless integrated network sensors. **Communications of the ACM**, ACM New York, NY, USA, v. 43, n. 5, p. 51–58, 2000.

2 AKYILDIZ, I. F. et al. A survey on sensor networks. **IEEE Communications Magazine**, v. 40, n. 8, p. 102–114, Aug. 2002. ISSN 1558-1896. DOI: 10.1109/MCOM.2002.1024422.

3 BULT, K. et al. Low power systems for wireless microsensors. In: PROCEEDINGS of 1996 International Symposium on Low Power Electronics and Design. 1996. P. 17–21. DOI: 10.1109/LPE.1996.542724.

4 ASADA, G. et al. Wireless integrated network sensors: Low power systems on a chip. In: PROCEEDINGS of the 24th European Solid-State Circuits Conference. 1998. P. 9–16. DOI: 10.1109/ESSCIR.1998.186200.

5 ESTRIN, D. et al. Instrumenting the world with wireless sensor networks. In: PROC. 2001 IEEE Int Conf. Acoust., Speech Signal Process. (ICASSP). 2001. v. 4, p. 2033–2036. DOI: 10.1109/ICASSP.2001.940390.

6 RASHID, B.; REHMANI, M. H. Applications of wireless sensor networks for urban areas: A survey. **Journal of Network and Computer Applications**, v. 60, p. 192–219, 2016. ISSN 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2015.09.008.

7 JINO RAMSON, S. R.; MONI, D. J. Applications of wireless sensor networks — A survey. In: 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT). 2017. P. 325–329. DOI: 10.1109/ICIEEIMT.2017.8116858.

8 PULE, M.; YAHYA, A.; CHUMA, J. Wireless sensor networks: A survey on monitoring water quality. **Journal of Applied Research and Technology**, v. 15, n. 6, p. 562–570, 2017. ISSN 1665-6423. DOI: 10.1016/j.jart.2017.07.004.

9 K. KHEDO, K.; PERSEEDOSS, R.; MUNGUR, A. A Wireless Sensor Network Air Pollution Monitoring System. **International Journal of Wireless and Mobile Networks**, Academy and Industry Research Collaboration Center (AIRCC), v. 2, n. 2, p. 31–45, May 2010. ISSN 0975-4679. DOI: 10.5121/ijwmn.2010.2203.

10 VIKRAM, R. et al. EEFFL: energy efficient data forwarding for forest fire detection using localization technique in wireless sensor network. **Wireless Networks**, v. 26, n. 7, p. 5177–5205, Oct. 2020. ISSN 1572-8196. DOI: 10.1007/s11276-020-02393-1.

11 SALHI, L. et al. Early Detection System for Gas Leakage and Fire in Smart Home Using Machine Learning. In: 2019 IEEE International Conference on Consumer Electronics (ICCE). 2019. P. 1–6. DOI: 10.1109/ICCE.2019.8661990.

12 MUDULI, L.; MISHRA, D. P.; JANA, P. K. Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review. **Journal of Network and Computer Applications**, v. 106, p. 48–67, 2018. ISSN 1084–8045. DOI: 10.1016/j.jnca.2017.12.022.

13 MOURTZIS, D.; VLACHOU, E. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. **Journal of Manufacturing Systems**, v. 47, p. 179–198, 2018. ISSN 0278-6125. DOI: 10.1016/j.jmsy.2018.05.008.

14 HUANG, Q.; MAO, C. Occupancy Estimation in Smart Building Using Hybrid $CO_2$/light Wireless Sensor Network. **Journal of Applied Sciences and Arts**, v. 1, n. 2, 2016. Available from: <https://opensiuc.lib.siu.edu/jasa/vol1/iss2/5>.

15 LU, W. et al. Collaborative Energy and Information Transfer in Green Wireless Sensor Networks for Smart Cities. **IEEE Transactions on Industrial Informatics**, v. 14, n. 4, p. 1585–1593, 2018. DOI: 10.1109/TII.2017.2777846.

16 DEY, N. et al. Developing residential wireless sensor networks for ECG healthcare monitoring. **IEEE Transactions on Consumer Electronics**, v. 63, n. 4, p. 442–449, 2017. DOI: 10.1109/TCE.2017.015063.

17 KADIRAVAN, D. et al. Metaheuristic Clustering Protocol for Healthcare Data Collection in Mobile Wireless Multimedia Sensor Networks. **Computers, Materials and Continua**, v. 66, p. 3215–3231, Jan. 2021. DOI: 10.32604/cmc.2021.013034.

18 GHOSH, K. et al. Intrusion Detection at International Borders and Large Military Barracks with Multi-sink Wireless Sensor Networks: An Energy Efficient Solution. **Wireless Pers Commun**, v. 98, p. 1083–1101, 2018. ISSN 0929-6212. DOI: 10.1007/s11277-017-4909-5.

19 KUMAR, S. A.; ILANGO, P. The Impact of Wireless Sensor Network in the Field of Precision Agriculture: A Review. **Wireless Personal Communications**, v. 98, n. 1, p. 685–698, 2018. ISSN 1572–834X. DOI: 10.1007/s11277-017-4890-z.

20 MAHBUB, M. A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. **Internet of Things**, v. 9, p. 100161, 2020. ISSN 2542–6605. DOI: 10.1016/j.iot.2020.100161.

21 RAHAMAN, M. M.; AZHARUDDIN, M. Wireless sensor networks in agriculture through machine learning: A survey. **Computers and Electronics in Agriculture**, v. 197, p. 106928, 2022. ISSN 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.106928. Available from: <https://www.sciencedirect.com/science/article/pii/S0168169922002459>.

22 PASCALE, A. et al. Wireless sensor networks for traffic management and road safety. **IET Intelligent Transport Systems**, v. 6, n. 1, p. 67–77, 2012. DOI: 10.1049/iet-its.2010.0129.

23 ABDELHAQ, M. F.; SALMAN, A. Wireless Sensor Network for Traffic Monitoring. In: 2020 International Conference on Promising Electronic Technologies (ICPET). 2020. P. 16–21. DOI: 10.1109/ICPET51420.2020.00012.

24 TALAMPAS, M. C. R. Wireless sensor networks for intelligent transportation applications: A survey. **Industrial Wireless Sensor Networks**, CRC Press, p. 47–77, 2017.

25 HOLFELD, B. et al. Wireless Communication for Factory Automation: an opportunity for LTE and 5G systems. **IEEE Communications Magazine**, v. 54, n. 6, p. 36–43, 2016. DOI: 10.1109/MCOM.2016.7497764.

26 CANDELL, R.; KASHEF, M. Industrial wireless: Problem space, success considerations, technologies, and future direction. In: 2017 Resilience Week (RWS). 2017. P. 133–139. DOI: 10.1109/RWEEK.2017.8088661.

27 VERA-AMARO, R.; ANGELES, M. E. R.; LUVIANO-JUAREZ, A. Design and Analysis of Wireless Sensor Networks for Animal Tracking in Large Monitoring Polar Regions Using Phase-Type Distributions and Single Sensor Model. **IEEE Access**, v. 7, p. 45911–45929, 2019. DOI: 10.1109/ACCESS.2019.2908308.

28 KIM DONG-SEONG AND TRAN-DANG, H. Wireless Sensor Networks for Industrial Applications. In: INDUSTRIAL Sensors and Controls in Communication Networks: From Wired Technologies to Cloud Computing and the Internet of Things. Cham: Springer International Publishing, 2019. P. 127–140. ISBN 978-3-030-04927-0. DOI: 10.1007/978-3-030-04927-0_10.

29 TENNEY, R. R.; SANDELL, N. R. Detection with Distributed Sensors. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE, AES-17, p. 501–510, 4 July 1981. ISSN 2371-9877. DOI: 10.1109/TAES.1981.309178.

30 WERNER-ALLEN, G. et al. Monitoring volcanic eruptions with a wireless sensor network. In: PROCEEEDINGS of the Second European Workshop on Wireless Sensor Networks, 2005. 2005. P. 108–120. DOI: 10.1109/EWSN.2005.1462003.

31 CHAIR, Z.; VARSHNEY, P. Distributed Bayesian hypothesis testing with distributed data fusion. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 18, n. 5, p. 695–699, 1988. DOI: 10.1109/21.21597.

32 TSITSIKLIS, J. N. Decentralized detection by a large number of sensors. **Mathematics of Control, Signals and Systems**, v. 1, n. 2, p. 167–182, 1988. DOI: https://doi.org/10.1007/BF02551407.

33 ________. Decentralized detection. **Advs. Statistical Signal Processing**, v. 2, p. 297–344, 1993.

34 VARSHNEY, P. K. **Distributed detection and data fusion**. New York, NY, USA: Springer-Verlag, 1997. ISBN 978-1-4612-7333-2. DOI: 10.1007/978-1-4612-1904-0.

35 VISWANATHAN, R.; VARSHNEY, P. Distributed detection with multiple sensors: Part I—fundamentals. **Proceedings of the IEEE**, IEEE, v. 85, n. 1, p. 54–63, 1 Jan. 1997. ISSN 1558-2256. DOI: 10.1109/5.554208.

36 BLUM, R.; KASSAM, S.; POOR, H. Distributed detection with multiple sensors: Part II—Advanced topics. **Proceedings of the IEEE**, v. 85, n. 1, p. 64–79, Jan. 1997. DOI: 10.1109/5.554209.

37 WILLETT, P.; SWASZEK, P.; BLUM, R. The good, bad and ugly: distributed detection of a known signal in dependent Gaussian noise. **IEEE Transactions on Signal Processing**, v. 48, n. 12, p. 3266–3279, 2000. DOI: 10.1109/78.886990.

38 CHAMBERLAND, J.-F.; VEERAVALLI, V. V. Decentralized Detection in Sensor Networks. **IEEE Transactions on Signal Processing**, v. 51, n. 2, p. 407–416, 2003. DOI: 10.1109/TSP.2002.806982.

39 ________. Asymptotic Results for Decentralized Detection in Power Constrained Wireless Sensor Networks. **IEEE Journal on Selected Areas in Communications**, v. 22, n. 6, p. 1007–1015, 2004. DOI: 10.1109/JSAC.2004.830894.

40 HASHEMI, H.; RHODES, I. Decentralized sequential detection. **IEEE Transactions on Information Theory**, v. 35, n. 3, p. 509–520, 1989. DOI: 10.1109/18.30973.

41 CHAMBERLAND, J.-F.; VEERAVALLI, V. V. Wireless Sensors in Distributed Detection Applications. **IEEE Signal Processing Magazine**, IEEE, v. 24, p. 16–25, 3 May 2007. ISSN 1558-0792. DOI: 10.1109/MSP.2007.361598.

42 CHEN, B.; TONG, L.; VARSHNEY, P. Channel-aware distributed detection in wireless sensor networks. **IEEE Signal Processing Magazine**, v. 23, n. 4, p. 16–26, 2006. DOI: 10.1109/MSP.2006.1657814.

43 NIU, R.; CHEN, B.; VARSHNEY, P. Fusion of decisions transmitted over Rayleigh fading channels in wireless sensor networks. **IEEE Transactions on Signal Processing**, v. 54, n. 3, p. 1018–1027, 2006. DOI: 10.1109/TSP.2005.863033.

44 CIUONZO, D.; ROSSI, P. S.; VARSHNEY, P. K. Distributed Detection in Wireless Sensor Networks Under Multiplicative Fading via Generalized Score Tests. **IEEE Internet of Things Journal**, v. 8, n. 11, p. 9059–9071, 2021. DOI: 10.1109/JIOT.2021.3056325.

45 BISWAS, N.; DAS, G.; RAY, P. Buffer-Aware User Selection and Resource Allocation for an Opportunistic Cognitive Radio Network: A Cross-Layer Approach. **IEEE/ACM Transactions on Networking**, v. 30, n. 5, p. 1940–1954, 2022. DOI: 10.1109/TNET.2022.3159819.

46 CHENG, X. et al. Multi-Bit & Sequential Decentralized Detection of a Noncooperative Moving Target Through a Generalized Rao Test. **IEEE Transactions on Signal and Information Processing over Networks**, v. 7, p. 740–753, 2021. DOI: 10.1109/TSIPN.2021.3126930.

47 ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787–2805, 2010. ISSN 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

48 KOCAKULAK, M.; BUTUN, I. An overview of Wireless Sensor Networks towards internet of things. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC). 2017. P. 1–6. DOI: 10.1109/CCWC.2017.7868374.

49 KRUGER, C. P.; HANCKE, G. P. Implementing the Internet of Things vision in industrial wireless sensor networks. In: 2014 12th IEEE International Conference on Industrial Informatics (INDIN). 2014. P. 627–632. DOI: 10.1109/INDIN.2014.6945586.

50 KIM, S. et al. Ambient RF Energy-Harvesting Technologies for Self-Sustainable Standalone Wireless Sensor Platforms. **Proceedings of the IEEE**, v. 102, n. 11, p. 1649–1666, 2014. DOI: 10.1109/JPROC.2014.2357031.

51 SHAIKH, F. K.; ZEADALLY, S. Energy harvesting in wireless sensor networks: A comprehensive review. **Renewable and Sustainable Energy Reviews**, v. 55, p. 1041–1054, 2016. ISSN 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.11.010.

52 ADU-MANU, K. S. et al. Energy-Harvesting Wireless Sensor Networks (EH-WSNs): A Review. **ACM Trans. Sen. Netw.**, Association for Computing Machinery, New York, NY, USA, v. 14, n. 2, Apr. 2018. ISSN 1550-4859. DOI: 10.1145/3183338. Available from: <https://doi.org/10.1145/3183338>.

53 SAH, D. K.; AMGOTH, T. Renewable energy harvesting schemes in wireless sensor networks: A Survey. **Information Fusion**, v. 63, p. 223–247, 2020. ISSN 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.07.005. Available from: <https://www.sciencedirect.com/science/article/pii/S156625352030316X>.

54 ARDESHIRI, G.; VOSOUGHI, A. On Adaptive Transmission for Distributed Detection in Energy Harvesting Wireless Sensor Networks With Limited Fusion Center Feedback. **IEEE Transactions on Green Communications and Networking**, v. 6, n. 3, p. 1764–1779, 2022. DOI: 10.1109/TGCN.2022.3146868.

55 FERRI, G. et al. Cooperative robotic networks for underwater surveillance: an overview. **IET Radar, Sonar & Navigation**, IET, v. 11, n. 12, p. 1740–1761, 2017.

56 MERINO, L. et al. Cooperative Fire Detection using Unmanned Aerial Vehicles. In: PROCEEDINGS of the 2005 IEEE International Conference on Robotics and Automation. 2005. P. 1884–1889. DOI: 10.1109/ROBOT.2005.1570388.

57 SUDHAKAR, S. et al. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. **Computer Communications**, v. 149, p. 1–16, 2020. ISSN 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.10.007. Available from: <https://www.sciencedirect.com/science/article/pii/S0140366419308655>.

58 FARRAR, C. R.; WORDEN, K. **Structural health monitoring: a machine learning perspective**. John Wiley & Sons, 2012.

59 FEITOSA, A. E. **Classification techniques for adaptive distributed networks and aeronautical structures.** 2018. MA thesis – Universidade de São Paulo.

60 FEITOSA, A. E.; NASCIMENTO, V. H.; LOPES, C. G. Adaptive detection in distributed networks using maximum likelihood detector. **IEEE Signal Processing Letters**, v. 25, n. 7, p. 974–978, July 2018. ISSN 1070-9908. DOI: 10.1109/LSP.2018.2832029.

61 OLFATI-SABER, R.; FAX, J. A.; MURRAY, R. M. Consensus and Cooperation in Networked Multi-Agent Systems. **Proceedings of the IEEE**, v. 95, n. 1, p. 215–233, 2007. DOI: 10.1109/JPROC.2006.887293.

62 KAR, S.; ALDOSARI, S.; MOURA, J. M. F. Topology for Distributed Inference on Graphs. **IEEE Transactions on Signal Processing**, v. 56, n. 6, p. 2609–2613, 2008. DOI: 10.1109/TSP.2008.923536.

63 SCHIZAS, I. D.; MATEOS, G.; GIANNAKIS, G. B. Distributed LMS for Consensus-Based In-Network Adaptive Processing. **IEEE Transactions on Signal Processing**, v. 57, n. 6, p. 2365–2382, 2009. DOI: 10.1109/TSP.2009.2016226.

64 DIMAKIS, A. G. et al. Gossip Algorithms for Distributed Signal Processing. **Proceedings of the IEEE**, v. 98, n. 11, p. 1847–1864, 2010. DOI: 10.1109/JPROC.2010.2052531.

65 BRACA, P.; MARANO, S.; MATTA, V. Enforcing Consensus While Monitoring the Environment in Wireless Sensor Networks. **IEEE Transactions on Signal Processing**, v. 56, n. 7, p. 3375–3380, 2008. DOI: 10.1109/TSP.2008.917855.

66 BRACA, P. et al. Asymptotic Optimality of Running Consensus in Testing Binary Hypotheses. **IEEE Transactions on Signal Processing**, v. 58, n. 2, p. 814–825, 2010. DOI: 10.1109/TSP.2009.2030610.

67 BAJOVIĆ, D. et al. Distributed Detection via Gaussian Running Consensus: Large Deviations Asymptotic Analysis. **IEEE Transactions on Signal Processing**, v. 59, n. 9, p. 4381–4396, 2011. DOI: 10.1109/TSP.2011.2157147.

68 BAJOVIĆ, D. et al. Large deviations performance of consensus+innovations distributed detection with non-Gaussian observations. **IEEE Transactions on Signal Processing**, v. 60, n. 11, p. 5987–6002, 2012. DOI: 10.1109/TSP.2012.2210885.

69 JAKOVETIĆ, D.; MOURA, J. M. F.; XAVIER, J. Distributed detection over noisy networks: large deviations analysis. **IEEE Transactions on Signal Processing**, v. 60, n. 8, p. 4306–4320, 2012. DOI: 10.1109/TSP.2012.2197395.

70 SAHU, A. K.; KAR, S. Recursive distributed detection for composite hypothesis testing: nonlinear observation models in additive Gaussian noise. **IEEE Transactions on Information Theory**, v. 63, n. 8, p. 4797–4828, 2017. DOI: 10.1109/TIT.2017.2686435.

71 LEONARD, M. R.; ZOUBIR, A. M. Robust sequential detection in distributed sensor networks. **IEEE Transactions on Signal Processing**, v. 66, n. 21, p. 5648–5662, Nov. 2018. DOI: 10.1109/TSP.2018.2869128.

72 LOPES, C. G.; SAYED, A. H. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. **IEEE Transactions on Signal Processing**, v. 56, n. 7, p. 3122–3136, 2008. DOI: 10.1109/TSP.2008.917383.

73 SAYED, A. H. et al. Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior. **IEEE Signal Processing Magazine**, v. 30, n. 3, p. 155–171, May 2013. ISSN 1053-5888. DOI: 10.1109/MSP.2012.2231991.

74 SAYED, A. H. Adaptive networks. **Proceedings of the IEEE**, v. 102, n. 4, p. 460–497, Apr. 2014. ISSN 0018-9219. DOI: 10.1109/JPROC.2014.2306253.

75 MATTA, V. et al. Diffusion-based adaptive distributed detection: steady-state performance in the slow adaptation regime. **IEEE Transactions on Information Theory**, v. 62, n. 8, p. 4710–4732, 2016. DOI: 10.1109/TIT.2016.2580665.

76 ________. Distributed detection over adaptive networks: refined asymptotics and the role of connectivity. **IEEE Transactions on Signal and Information Processing over Networks**, v. 2, n. 4, p. 442–460, 2016. DOI: 10.1109/TSIPN.2016.2613682.

77 CATTIVELLI, F. S.; SAYED, A. H. Diffusion LMS-based distributed detection over adaptive networks. In: PROC. 2009 Conf. Signals, Syst. Comput. (Asilomar). 2009. P. 171–175. DOI: 10.1109/ACSSC.2009.5470136.

78 ________. Distributed detection over adaptive networks using diffusion adaptation. **IEEE Transactions on Signal Processing**, v. 59, n. 5, p. 1917–1932, 2011. DOI: 10.1109/TSP.2011.2107902.

79 LEE, J.-W. et al. Spatio-temporal diffusion strategies for estimation and detection over networks. **IEEE Transactions on Signal Processing**, v. 60, n. 8, p. 4017–4034, 2012. DOI: 10.1109/TSP.2012.2197205.

80 AL-SAYED, S.; ZOUBIR, A. M.; SAYED, A. H. Robust distributed detection over adaptive diffusion networks. In: PROC. 2014 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP). 2014. P. 7233–7237. DOI: 10.1109/ICASSP.2014.6855004.

81 AL-SAYED, S. et al. Node-specific diffusion LMS-based distributed detection over adaptive networks. **IEEE Transactions on Signal Processing**, v. 66, n. 3, p. 682–697, Feb. 2018. ISSN 1053-587X. DOI: 10.1109/TSP.2017.2771731.

82 MARANO, S.; SAYED, A. H. Detection under one-bit messaging over adaptive networks. **IEEE Transactions on Information Theory**, v. 65, n. 10, p. 6519–6538, Oct. 2019. DOI: 10.1109/TIT.2019.2916845.

83 CATTIVELLI, F. S.; LOPES, C. G.; SAYED, A. H. Diffusion recursive least-squares for distributed estimation over adaptive networks. **IEEE Transactions on Signal Processing**, v. 56, n. 5, p. 1865–1877, 2008. DOI: 10.1109/TSP.2007.913164.

84 CATTIVELLI, F. S.; SAYED, A. H. Diffusion strategies for distributed Kalman filtering and smoothing. **IEEE Transactions on Automatic Control**, v. 55, n. 9, p. 2069–2084, Sept. 2010. ISSN 2334-3303. DOI: 10.1109/TAC.2010.2042987.

85 TU, S.-Y.; SAYED, A. H. Diffusion Strategies Outperform Consensus Strategies for Distributed Estimation Over Adaptive Networks. **IEEE Transactions on Signal Processing**, v. 60, n. 12, p. 6217–6234, 2012. DOI: 10.1109/TSP.2012.2217338.

86 KAY, S. M. **Fundamentals of statistical signal processing: Detection theory, vol. 2**. Prentice Hall Upper Saddle River, NJ, USA, 1998.

87 CIOFFI, J. M. Signal processing and detection. Class notes, page 18. Available from: <https://cioffi-group.stanford.edu/doc/book/chap1.pdf>.

88 FEITOSA, A. E.; NASCIMENTO, V. H.; LOPES, C. G. A low-complexity map detector for distributed networks. In: PROC. 2020 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP). IEEE, May 2020. DOI: 10.1109/icassp40776.2020.9054197.

89 SHAHRAMPOUR, S.; RAKHLIN, A.; JADBABAIE, A. Distributed Detection: Finite-Time Analysis and Impact of Network Topology. **IEEE Transactions on Automatic Control**, v. 61, n. 11, p. 3256–3268, 2016. DOI: 10.1109/TAC.2015.2506903.

90 LI, S.; WANG, X. Fully Distributed Sequential Hypothesis Testing: Algorithms and Asymptotic Analyses. **IEEE Transactions on Information Theory**, v. 64, n. 4, p. 2742–2758, 2018. DOI: 10.1109/TIT.2018.2806961.

91 VEERAVALLI, V. Decentralized quickest change detection. **IEEE Transactions on Information Theory**, v. 47, n. 4, p. 1657–1665, 2001. DOI: 10.1109/18.923755.

92 XIE, L. et al. Sequential (Quickest) Change Detection: Classical Results and New Directions. **IEEE Journal on Selected Areas in Information Theory**, v. 2, n. 2, p. 494–514, 2021. DOI: 10.1109/JSAIT.2021.3072962.

93 BRACA, P. et al. Quickest Detection of COVID-19 Pandemic Onset. **IEEE Signal Processing Letters**, v. 28, p. 683–687, 2021. DOI: 10.1109/LSP.2021.3068072.

94 MARANO, S.; SAYED, A. H. Decision Learning and Adaptation Over Multi-Task Networks. **IEEE Transactions on Signal Processing**, v. 69, p. 2873–2887, 2021. DOI: 10.1109/TSP.2021.3077804.

95 LALITHA, A.; JAVIDI, T.; SARWATE, A. D. Social Learning and Distributed Hypothesis Testing. **IEEE Transactions on Information Theory**, v. 64, n. 9, p. 6161–6179, 2018. DOI: 10.1109/TIT.2018.2837050.

96 BORDIGNON, V.; MATTA, V.; SAYED, A. H. Partial Information Sharing Over Social Learning Networks. **IEEE Transactions on Information Theory**, v. 69, n. 3, p. 2033–2058, 2023. DOI: 10.1109/TIT.2022.3227587.

97 RANGI, A.; FRANCESCHETTI, M.; MARANO, S. Distributed Chernoff Test: Optimal Decision Systems Over Networks. **IEEE Transactions on Information Theory**, v. 67, n. 4, p. 2399–2425, 2021. DOI: 10.1109/TIT.2020.3046191.

98 SAYED, A. H. **Adaptive filters**. John Wiley & Sons, 2011.

99 NASCIMENTO, V. H.; SILVA, M. T. M. Adaptive Filters. In: **Academic press library in signal processing: Signal processing theory and machine learning**. Ed. by Sergios Theodoridis Rama Chellappa. Academic Press, 2014. v. 1, p. 619–761.

100 KAY, S. M. **Fundamentals of statistical signal processing: Estimation theory, vol. 1**. Prentice Hall PTR, 1993.

101 PILLAI, S.; SUEL, T.; CHA, S. The Perron-Frobenius theorem: some of its applications. **IEEE Signal Processing Magazine**, v. 22, n. 2, p. 62–75, 2005. DOI: 10.1109/MSP.2005.1406483.

102 HAYKIN, S. O. **Adaptive filter theory**. Pearson Higher Ed, 2013.

103 XIAO, L.; BOYD, S.; LALL, S. A scheme for robust distributed sensor fusion based on average consensus. In: PROC. Fourth Int. Symp. Inf. Process. in Sensor Networks (IPSN). Apr. 2005. P. 63–70. DOI: 10.1109/IPSN.2005.1440896.

# Appendix

# APPENDIX A – THE COVARIANCE MATRIX OF THE dLMS

We deduce here the expressions for the covariance matrix of the diffusion-LMS estimator. By Equation (3.21) and Equation (3.22), we have that

$$\underset{\sim}{\hat{\mathrm{w}}}[\,i\,] - \mathrm{E}(\underset{\sim}{\hat{\mathrm{w}}}[\,i\,] \mid H_n) = \mathcal{P}[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n) + \mathcal{L}[\,i\,]\boldsymbol{v}_{0:i}.$$

Call $\underset{\sim}{\boldsymbol{\alpha}}[\,i\,] = \underset{\sim}{\hat{\mathrm{w}}}[\,i\,] - \mathrm{E}(\underset{\sim}{\hat{\mathrm{w}}}[\,i\,] \mid H_n)$; then, we have that

$$\begin{aligned}\underset{\sim}{\boldsymbol{\alpha}}[\,i\,]\underset{\sim}{\boldsymbol{\alpha}}^{\top}[\,i\,] = {} & \mathcal{P}[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)(\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)^{\top}\mathcal{P}^{\top}[\,i\,] \\ & + \mathcal{P}[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)\boldsymbol{v}_{0:i}^{\top}\mathcal{L}^{\top}[\,i\,] \\ & + \mathcal{L}[\,i\,]\boldsymbol{v}_{0:i}(\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)^{\top}\mathcal{P}^{\top}[\,i\,] \\ & + \mathcal{L}[\,i\,]\boldsymbol{v}_{0:i}\boldsymbol{v}_{0:i}^{\top}\mathcal{L}^{\top}[\,i\,].\end{aligned}$$

Recall that matrices $\mathcal{L}[\,i\,]$ and $\mathcal{P}[\,i\,]$ are deterministic, since the regressors $\boldsymbol{u}_k[\,i\,]$ are also defined as such. Moreover, the state vector process $\underset{\sim}{\boldsymbol{w}}_n$ and the noise $\tilde{v}_k[\,i\,]$ are independent for all $k$ and $i$, and $\mathrm{E}(()v_k[\,i\,]) = 0$. Therefore, the covariance of $\underset{\sim}{\hat{\mathrm{w}}}[\,i\,]$, given as $\mathcal{S}[\,i\,] = \mathrm{E}(\underset{\sim}{\boldsymbol{\alpha}}[\,i\,]\underset{\sim}{\boldsymbol{\alpha}}^{\top}[\,i\,])$, is

$$\mathcal{S}[\,i\,] = \mathcal{P}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{P}^{\top}[\,i\,] + \mathcal{L}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}^{\top}[\,i\,],$$

which proves Equation (3.23) by defining $\mathcal{Z}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\mathcal{L}^{\top}[\,i\,]$. The first term in Equation (3.23) can be expanded as

$$\begin{aligned}\mathcal{P}[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{P}^{\top}[\,i\,] &= \begin{bmatrix}\boldsymbol{P}_1[\,i\,] \\ \vdots \\ \boldsymbol{P}_k[\,i\,]\end{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{w}} \begin{bmatrix}\boldsymbol{P}_1^{\top}[\,i\,] & \cdots & \boldsymbol{P}_K^{\top}[\,i\,]\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{P}_1[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_1^{\top}[\,i\,] & \cdots & \boldsymbol{P}_1[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_K^{\top}[\,i\,] \\ \vdots & \ddots & \vdots \\ \boldsymbol{P}_K[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_1^{\top}[\,i\,] & \cdots & \boldsymbol{P}_K[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_K^{\top}[\,i\,]\end{bmatrix}.\end{aligned}$$

Since $\boldsymbol{Z}_k[\,i\,]$ is the $k$-th $M \times M$ diagonal block of matrix $\mathcal{Z}[\,i\,]$, as a consequence we have $\boldsymbol{S}_k[\,i\,] = \boldsymbol{P}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,] + \boldsymbol{Z}_k[\,i\,]$, which is the expression presented in Equation (3.27).

Lastly, we prove that $\boldsymbol{Z}_k[\,i\,] = \mathrm{E}(\underset{\sim}{\boldsymbol{z}}_k[\,i\,]\underset{\sim}{\boldsymbol{z}}_k^{\top}[\,i\,])$. Note that, from Equation (2.3) and Equation (4.8), we have

$$\underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,] - \mathrm{E}(\underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,] \mid H_n) = \boldsymbol{P}_k[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n) + \underset{\sim}{\boldsymbol{z}}_k[\,i\,],$$

where $\underset{\sim}{\boldsymbol{z}}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{v}_{0:i}$ just as defined in Equation (4.30). Next, call

$$\underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,] - \mathrm{E}(\underset{\sim}{\hat{\boldsymbol{w}}}_k[\,i\,] \mid H_n),$$

and expand

$$\begin{aligned}\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}^{\top} = \boldsymbol{P}_k[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)(\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)^{\top}\boldsymbol{P}_k^{\top}[\,i\,] + \boldsymbol{P}_k[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)\underset{\sim}{\boldsymbol{z}}_k^{\top}[\,i\,] \\ + \underset{\sim}{\boldsymbol{z}}_k[\,i\,](\underset{\sim}{\boldsymbol{w}}_n - \boldsymbol{\theta}_n)^{\top} + \underset{\sim}{\boldsymbol{z}}_k[\,i\,]\underset{\sim}{\boldsymbol{z}}_k^{\top}[\,i\,].\end{aligned}$$

Under the same conditions assumed so far, the covariance matrix of $\underset{\sim}{\widehat{\boldsymbol{w}}}_k[\,i\,]$ can also be expressed as $\boldsymbol{S}_k[\,i\,] = \mathrm{E}(\,\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}^{\top})$ by definition. Therefore,

$$\boldsymbol{S}_k[\,i\,] = \boldsymbol{P}_k[\,i\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{P}_k^{\top}[\,i\,] + \mathrm{E}\big(\boldsymbol{z}_k[\,i\,]\boldsymbol{z}_k^{\top}[\,i\,]\big);$$

comparing with Equation (3.27), we must have $\boldsymbol{Z}_k[\,i\,] = \mathrm{E}\big(\boldsymbol{z}_k[\,i\,]\boldsymbol{z}_k^{\top}[\,i\,]\big)$.

# APPENDIX B – DEFINITION OF MATRIX $\boldsymbol{L}_k[\,i\,]$

In Chapter 4 we defined the matrix $\boldsymbol{L}_k[\,i\,] \in \mathbb{R}^{D\times(i+1)K}$ such that $\boldsymbol{P}_k[\,i\,] = \boldsymbol{L}_k[\,i\,]\boldsymbol{U}_{0:i}$. In fact, $\boldsymbol{L}_k[\,i\,]$ is well defined in this way if

$$\boldsymbol{L}_k[\,0\,] = \sum_{\ell\in\vartheta_k} a_{\ell k}\mu_\ell \boldsymbol{W}_\ell[\,0\,], \tag{B.1}$$

$$\boldsymbol{L}_k[\,i\,] = \sum_{\ell\in\vartheta_k} a_{\ell k}[\,\mu_\ell \boldsymbol{W}_\ell[\,i\,] \quad \mathbf{Y}_\ell[\,i\,]\boldsymbol{L}_\ell[\,i-1\,]\,],\ i \geq 1, \tag{B.2}$$

where we define $\boldsymbol{W}_\ell[\,i\,] = [\,\mathbf{0}\ \ldots\ \boldsymbol{u}_\ell[\,i\,]\ \ldots\ \mathbf{0}\,]$ whose $\ell$th column is the only nonzero column. To show that the expressions in Equation (B.1) and Equation (B.2) are adequate, we need to develop a few new expressions first. From Equation (3.9) and Equation (3.16), we can rewrite $\hat{\mathrm{w}}[\,i\,]$ as

$$\begin{aligned}\hat{\mathrm{w}}[\,i\,] &= \mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{d}[\,i\,] + \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{L}[i-1]\boldsymbol{d}_{0:i-1} + \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{F}[\,i-1,0\,]\\ &= [\,\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\ \ \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{L}[i-1]\,]\begin{bmatrix}\boldsymbol{d}[\,i\,]\\ \boldsymbol{d}_{0:i-1}\end{bmatrix} + \mathcal{F}[\,i,0\,];\end{aligned}$$

therefore, by inspection we conclude, from Equation (3.16), that $\mathcal{L}[\,i\,]$ can be written as the following recursive expression:

$$\mathcal{L}[\,i\,] = [\,\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\ \ \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{L}[i{-}1]\,]. \tag{B.3}$$

We then apply the definition $\mathcal{P}[\,i\,] = \mathcal{L}[\,i\,]\boldsymbol{U}_{0:i}$ to obtain

$$\begin{aligned}\mathcal{P}[\,i\,] &= [\,\mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\ \ \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{L}[i-1]\,]\boldsymbol{U}_{0:i}\\ &= \mathcal{A}\mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{U}[\,i\,] + \mathcal{A}\mathcal{Y}[\,i\,]\mathcal{L}[i-1]\boldsymbol{U}_{0:i-1}\\ &= \mathcal{A}(\mathcal{M}\mathcal{U}[\,i\,]\boldsymbol{U}[\,i\,] + \mathcal{Y}[\,i\,]\mathcal{P}[i-1]).\end{aligned} \tag{B.4}$$

From the definitions of $\mathcal{M}$, $\mathcal{U}[\,i\,]$ and $\mathcal{Y}[\,i\,]$ in Equation (3.7), of $\boldsymbol{U}[\,i\,]$ in Equation (3.13), and that of $\boldsymbol{P}_k[\,i\,]$ in Equation (3.26), we have

$$\mathcal{P}[\,i\,] = \mathcal{A}\left(\begin{bmatrix}\mu_1\boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,]\\ \vdots\\ \mu_K\boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,]\end{bmatrix} + \begin{bmatrix}\mathbf{Y}_1[\,i\,]\boldsymbol{P}_1[\,i-1\,]\\ \vdots\\ \mathbf{Y}_K[\,i\,]\boldsymbol{P}_K[\,i-1\,]\end{bmatrix}\right).$$

Therefore, from $\mathcal{A} = \boldsymbol{A}^\top \otimes \boldsymbol{I}_D$, we have

$$\boldsymbol{P}_k[\,i\,] = \sum_{\ell=1}^{K} a_{\ell k}(\mu_\ell\boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,] + \mathbf{Y}_\ell[\,i\,]\boldsymbol{P}_\ell[i{-}1]). \tag{B.5}$$

For $i = 0$, and setting $\boldsymbol{P}_k[-1] = \mathbf{0}\ \forall k$, we have

$$\boldsymbol{P}_k[\,0\,] = \sum_{\ell=1}^{K} a_{\ell k}\mu_\ell\boldsymbol{u}_\ell[\,0\,]\boldsymbol{u}_\ell^\top[\,0\,] = \sum_{\ell=1}^{K} a_{\ell k}\mu_\ell\boldsymbol{W}_\ell[\,0\,]\boldsymbol{U}[\,0\,],$$

and since $\boldsymbol{U}[\,0\,] = \boldsymbol{U}_{0:0}$, we have $\boldsymbol{P}_k[\,0\,] = \boldsymbol{L}_k[\,0\,]\boldsymbol{U}_{0:0}$ if we define $\boldsymbol{L}_k[\,0\,]$ according to Equation (B.1), which shows that it is in fact an adequate expression for $\boldsymbol{L}_k[\,i\,]$ for $i = 0$. For $i \geq 1$, suppose that $\boldsymbol{P}_k[\,i-1\,] = \boldsymbol{L}_k[i-1]\boldsymbol{U}_{0:i-1}$ is a true statement; thus, we can rewrite Equation (B.5) as

$$\boldsymbol{P}_k[\,i\,] = \sum_{\ell=1}^{K} a_{\ell k}[\,\mu_\ell \boldsymbol{W}_\ell[\,i\,] \;\; \boldsymbol{Y}_\ell[\,i\,]\boldsymbol{L}_\ell[\,i-1\,]\,] \begin{bmatrix} \boldsymbol{U}[\,i\,] \\ \boldsymbol{U}_{0:i-1} \end{bmatrix} = \boldsymbol{L}_k[\,i\,]\boldsymbol{U}_{0:i}\,.$$

Therefore, by a mathematical induction process, we can conclude that the expression in Equation (B.2) is also adequate for $\boldsymbol{L}_k[\,i\,]$ to be well defined.

# APPENDIX C – THE COEFFICIENTS OF $\mathcal{P}[\,i\,]$ AND $\mathcal{Z}[\,i\,]$

In Section 4.3, we were presented to the coefficients $\mathcal{B}_\kappa[\,i\,]$ of the polynomial expansion of $\mathcal{P}[\,i\,]$ in Equation (4.13). In order to obtain such coefficients, we evaluate at $\mu = 0$

$$\mathcal{B}_\kappa[\,i\,] = \frac{1}{\kappa!}\left(\frac{d^\kappa}{d\mu^\kappa}\mathcal{P}[\,i\,]\right)\Bigg|_{\mu=0}, \ \kappa \geq 0. \tag{C.1}$$

From Equation (3.10) we obtain

$$\frac{d}{d\mu}\mathcal{F}[\,i,j\,] = -\sum_{\iota=j}^{i}\mathcal{F}[\,i,\iota+1\,]\mathcal{A}\mathcal{H}[\,\iota\,]\mathcal{F}[\,\iota-1,j\,], \tag{C.2}$$

where $\mathcal{H}[\,\iota\,] = \mathcal{M}'\mathcal{U}[\,j\,]\mathcal{U}[\,j\,]$. Thus, rewriting $\mathcal{P}[\,i\,]$ in Equation (3.20) as

$$\mathcal{P}[\,i\,] = \mu\sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\boldsymbol{H}[\,j\,],$$

we have, for $\kappa = 1$ and $\kappa = 2$, the following:

$$\begin{aligned}
\frac{d}{d\mu}\mathcal{P}[\,i\,] &= \sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}\boldsymbol{H}[\,\iota\,] + \mu\sum_{j=0}^{i}\frac{d}{d\mu}\mathcal{F}[\,i,j+1\,]\mathcal{A}\boldsymbol{H}[\,\iota\,]\\
\frac{d^2}{d\mu^2}\mathcal{P}[\,i\,] &= 2\sum_{j=0}^{i}\frac{d}{d\mu}\mathcal{F}[\,i,j+1\,]\mathcal{A}\boldsymbol{H}[\,\iota\,] + \mu\sum_{j=0}^{i}\frac{d^2}{d\mu^2}\mathcal{F}[\,i,j+1\,]\mathcal{A}\boldsymbol{H}[\,\iota\,].
\end{aligned}$$

Finally, for $0 \leq \kappa \leq 2$ we evaluate $\mathcal{B}_\kappa[\,i\,]$ in Equation (C.1) using the following identities, which result from Equation (3.10) and Equation (C.2):

$$\begin{aligned}
\mathcal{F}[\,i,j+1\,]\big|_{\mu=0} &= \mathcal{A}^{i-j},\\
\frac{d}{d\mu}\mathcal{F}[\,i,j+1\,]\Bigg|_{\mu=0} &= \sum_{\iota=j}^{i}\mathcal{A}^{i-\iota+1}\mathcal{H}[\,\iota\,]\mathcal{A}^{\iota-j}.
\end{aligned}$$

We evaluate the coefficients $\mathcal{C}_\kappa[\,i\,]$ of the polynomial expansion of $\mathcal{Z}[\,i\,]$ in Equation (4.19) in the same way. For $\kappa = 0$ and $\kappa = 1$, its trivial that $\mathcal{C}_\kappa[\,i\,] = \mathbf{0}$. Let us rewrite $\mathcal{Z}[\,i\,]$ in Equation (4.19) as

$$\mathcal{Z}[\,i\,] = \mu^2\sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}^\top\mathcal{D}'[\,j\,]\mathcal{A}\mathcal{F}^\top[\,i,j+1\,],$$

where $\mathcal{D}'[\,j\,] = \mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{\Sigma}_{\boldsymbol{w}}\mathcal{U}^\top[\,j\,](\mathcal{M}')^\top$. Then we have

$$\begin{aligned}
\frac{d^2}{d\mu^2}\mathcal{Z}[\,i\,] = {} & 2\sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}^\top\mathcal{D}'[\,j\,]\mathcal{A}\mathcal{F}^\top[\,i,j+1\,]\\
& + 2\mu\frac{d}{d\mu}\left(\sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}^\top\mathcal{D}'[\,j\,]\mathcal{A}\mathcal{F}^\top[\,i,j+1\,]\right)\\
& + \mu^2\frac{d^2}{d\mu^2}\left(\sum_{j=0}^{i}\mathcal{F}[\,i,j+1\,]\mathcal{A}^\top\mathcal{D}'[\,j\,]\mathcal{A}\mathcal{F}^\top[\,i,j+1\,]\right).
\end{aligned}$$

Finally, one needs only to evaluate $\mathcal{C}_2[\,i\,] = \frac{1}{2}\frac{d^2}{d\mu^2}\mathcal{Z}[\,i\,]|_{\mu=0}$.

# APPENDIX D – APPROXIMATIONS OF $\boldsymbol{P}_k[\,i\,]$ AND $\boldsymbol{Z}_k[\,i\,]$

First, let us demonstrate the approximation of $\boldsymbol{P}_k[\,i\,]$ in Equation (4.18). Notice that

$$\boldsymbol{A}^i \otimes \boldsymbol{I}_D = \begin{bmatrix} (\boldsymbol{A}^i)_{11}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{12}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^i)_{1K}\boldsymbol{I}_D \\ (\boldsymbol{A}^i)_{21}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{22\boldsymbol{I}_D} & \cdots & (\boldsymbol{A}^i)_{2K}\boldsymbol{I}_D \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{A}^i)_{K1}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{K2}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^i)_{KK}\boldsymbol{I}_D \end{bmatrix}.$$

From Proposition 8, we have that $\mathcal{A}^i = (\boldsymbol{A}^\top)^i \otimes \boldsymbol{I}_D$. Note that

$$(\boldsymbol{A}^\top)^i = \underbrace{\boldsymbol{A}^\top \cdot \boldsymbol{A}^\top \cdot\ \dots\ \cdot \boldsymbol{A}^\top}_{i \text{ times}} = (\underbrace{\boldsymbol{A} \cdot \boldsymbol{A} \cdot\ \dots\ \cdot \boldsymbol{A}}_{i \text{ times}})^\top = (\boldsymbol{A}^i)^\top;$$

thus, from the property $(\boldsymbol{M}_1 \otimes \boldsymbol{M}_2)^\top = \boldsymbol{M}_1^\top \otimes \boldsymbol{M}_2^\top$ of the Kronecker product,

$$\mathcal{A}^i = (\boldsymbol{A}^\top)^i \otimes \boldsymbol{I}_D = (\boldsymbol{A}^i)^\top \otimes \boldsymbol{I}_D = (\boldsymbol{A}^i \otimes \boldsymbol{I}_D)^\top = \begin{bmatrix} (\boldsymbol{A}^i)_{11}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{21}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^i)_{K1}\boldsymbol{I}_D \\ (\boldsymbol{A}^i)_{12}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{22}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^i)_{K2}\boldsymbol{I}_D \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{A}^i)_{1K}\boldsymbol{I}_D & (\boldsymbol{A}^i)_{2K}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^i)_{KK}\boldsymbol{I}_D \end{bmatrix}.$$

From the definition of $\boldsymbol{H}[\,i\,]$ in Equation (4.15), we have

$$\begin{aligned} \boldsymbol{H}[\,i\,] &= \mathcal{M}'\,\mathcal{U}[\,j\,]\boldsymbol{U}[\,j\,] \\ &= \begin{bmatrix} \mu_1'\boldsymbol{I}_D & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \mu_2'\boldsymbol{I}_D & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \mu_K'\boldsymbol{I}_D \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u}_1[\,i\,] & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{u}_2[\,i\,] & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{u}_K[\,i\,] \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u}_1^\top[\,i\,] \\ \boldsymbol{u}_2^\top[\,i\,] \\ \vdots \\ \boldsymbol{u}_K^\top[\,i\,] \end{bmatrix} \\ &= \begin{bmatrix} \mu_1'\boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] \\ \mu_2'\boldsymbol{u}_2[\,i\,]\boldsymbol{u}_2^\top[\,i\,] \\ \vdots \\ \mu_K'\boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix} \end{aligned}$$

Thus, from the approximation of $\mathcal{P}[\,i\,]$ in Equation (4.16), we have

$$\begin{aligned} \frac{\mathcal{P}[\,i\,]}{\mu} &\approx \sum_{j=0}^{i} \mathcal{A}^{i-j+1}\boldsymbol{H}[\,i\,]\mathcal{A}^{i-j+1}\boldsymbol{H}[\,i\,] \\ &= \sum_{j=0}^{i} ((\boldsymbol{A}^\top)^{i-1+1} \otimes \boldsymbol{I}_D)\boldsymbol{H}[\,i\,] \\ &= \sum_{j=0}^{i} \begin{bmatrix} (\boldsymbol{A}^{i-j+1})_{11}\boldsymbol{I}_D & (\boldsymbol{A}^{i-j+1})_{21}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{K1}\boldsymbol{I}_D \\ (\boldsymbol{A}^{i-j+1})_{12}\boldsymbol{I}_D & (\boldsymbol{A}^{i-j+1})_{22}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{K2}\boldsymbol{I}_D \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{A}^{i-j+1})_{1K}\boldsymbol{I}_D & (\boldsymbol{A}^{i-j+1})_{2K}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{KK} \end{bmatrix} \cdot \begin{bmatrix} \mu_1'\boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] \\ \mu_2'\boldsymbol{u}_2[\,i\,]\boldsymbol{u}_2^\top[\,i\,] \\ \vdots \\ \mu_K'\boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix} \end{aligned}$$

$$= \sum_{j=0}^{i} \begin{bmatrix} \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell 1}\, \mu_1' \boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] \\ \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell 2}\, \mu_2' \boldsymbol{u}_2[\,i\,]\boldsymbol{u}_2^\top[\,i\,] \\ \vdots \\ \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell K}\, \mu_K' \boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix}.$$

Therefore, seeing $\mathcal{P}[\,i\,]$ divided into block matrices $\boldsymbol{P}_k[\,i\,]$ as in Equation (3.26), we have

$$\boldsymbol{P}_k[\,i\,] \approx \mu \sum_{j=0}^{i} \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell k}\, \mu_\ell' \boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,],$$

as we wanted to demonstrate.

We now demonstrate the approximation of $\boldsymbol{Z}_k[\,i\,]$ in Equation (4.21). From the definitions of $\mathcal{U}[\,i\,]$ in Equation (3.7), $\boldsymbol{\Sigma}_v$ in Equation (3.24) and $\mathcal{M}'$ in Equation (4.12), and the fact that all these matrices are diagonal block matrices, the following product is a diagonal matrix as well:

$$\mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{\Sigma}_v\mathcal{U}^\top[\,j\,]\mathcal{M}' = \begin{bmatrix} (\mu_1')^2\sigma_{v_1}^2 \boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & (\mu_K')^2\sigma_{v_K}^2 \boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix}.$$

Hence, we have

$$\begin{aligned}
\frac{\mathcal{Z}[\,i\,]}{\mu^2} &\approx \sum_{j=0}^{i} \mathcal{A}^{i-j+1}\mathcal{M}'\mathcal{U}[\,j\,]\boldsymbol{\Sigma}_v\mathcal{U}^\top[\,j\,]\mathcal{M}'(\mathcal{A}^\top)^{i-j+1} \\
&= \sum_{j=0}^{i} \begin{bmatrix} (\boldsymbol{A}^{i-j+1})_{11}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{K1}\boldsymbol{I}_D \\ \vdots & \ddots & \vdots \\ (\boldsymbol{A}^{i-j+1})_{1K}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{KK} \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} (\mu_1')^2\sigma_{v_1}^2 \boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & (\mu_K')^2\sigma_{v_K}^2 \boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} (\boldsymbol{A}^{i-j+1})_{11}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{1K}\boldsymbol{I}_D \\ \vdots & \ddots & \vdots \\ (\boldsymbol{A}^{i-j+1})_{K1}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{KK} \end{bmatrix} \\
&= \sum_{j=0}^{i} \begin{bmatrix} (\boldsymbol{A}^{i-j+1})_{11}(\mu_1')^2\sigma_{v_1}^2 \boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] & \cdots & (\boldsymbol{A}^{i-j+1})_{K1}(\mu_K')^2\sigma_{v_K}^2 \boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K[\,i\,] \\ \vdots & \ddots & \vdots \\ (\boldsymbol{A}^{i-j+1})_{1K}(\mu_1')^2\sigma_{v_1}^2 \boldsymbol{u}_1[\,i\,]\boldsymbol{u}_1^\top[\,i\,] & \cdots & (\boldsymbol{A}^{i-j+1})_{2K}(\mu_K')^2\sigma_{v_K}^2 \boldsymbol{u}_K[\,i\,]\boldsymbol{u}_K^\top[\,i\,] \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} (\boldsymbol{A}^{i-j+1})_{11}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{1K}\boldsymbol{I}_D \\ \vdots & \ddots & \vdots \\ (\boldsymbol{A}^{i-j+1})_{K1}\boldsymbol{I}_D & \cdots & (\boldsymbol{A}^{i-j+1})_{KK} \end{bmatrix} \\
&= \sum_{j=0}^{i} \begin{bmatrix} \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell 1}^2 (\mu_\ell')^2\sigma_{v_\ell}^2 \boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,] & \cdots & \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell 1}(\boldsymbol{A}^{i-j+1})_{\ell K}(\mu_\ell')^2\sigma_{v_\ell}^2 \boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,] \\ \vdots & \ddots & \vdots \\ \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell 1}(\boldsymbol{A}^{i-j+1})_{\ell K}(\mu_\ell')^2\sigma_{v_\ell}^2 \boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,] & \cdots & \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})_{\ell K}^2 (\mu_\ell')^2\sigma_{v_\ell}^2 \boldsymbol{u}_\ell[\,i\,]\boldsymbol{u}_\ell^\top[\,i\,] \end{bmatrix}.
\end{aligned}$$

Therefore,

$$\boldsymbol{Z}_k[\,i\,] \approx \mu^2 \sum_{j=0}^{i} \sum_{\ell=1}^{K} (\boldsymbol{A}^{i-j+1})^2_{\ell k} (\mu'_\ell)^2 \sigma^2_{v_\ell} \boldsymbol{u}_\ell[\,i\,] \boldsymbol{u}_\ell^\top[\,i\,],$$

as we wanted to demonstrate.

# APPENDIX E – PERFORMANCE OF THE OPTIMAL DETECTOR

## Performance at each time instant $i$

In order to obtain the theoretical performance of an optimal detector, we need, as usual, the expectations and the covariances of its statistics. We start by first expanding Equation (4.37) and remove the term which is the same for all $H_n$, and define a new test statistic given as

$$o'_n[\,i\,] = (\boldsymbol{d}_{0:i} - \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n)^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{d}_{0:i}} \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n - \frac{1}{2}\boldsymbol{\theta}_n^\top \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}} \boldsymbol{U}_{0:i}\boldsymbol{\theta}_n. \tag{E.1}$$

For simplicity, define $\boldsymbol{C}_{0:i} = \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}} \boldsymbol{U}_{0:i}$. We then obtain:

$$\begin{aligned}
\mathrm{E}(o'_m[\,i\,] \mid H_n) &= \boldsymbol{\theta}_m^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_n - \frac{1}{2}\boldsymbol{\theta}_m^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_m,\\
\mathrm{Var}(o'_m[\,i\,]) &= \boldsymbol{\theta}_m^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_m,\\
\mathrm{Cov}(o'_m[\,i\,], o'_\nu[\,i\,]) &= \boldsymbol{\theta}_m^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_\nu.
\end{aligned}$$

Define $\boldsymbol{o}'[\,i\,] = \mathrm{col}\{o'_1[\,i\,], \,\dots\,, o'_N[\,i\,]\}$ and $\boldsymbol{\Theta} = [\,\boldsymbol{\theta}_1 \,\dots\, \boldsymbol{\theta}_N\,]$. Therefore, we have

$$\begin{aligned}
\mathrm{E}(\boldsymbol{o}'[\,i\,] \mid H_n) &= \boldsymbol{\Theta}^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_n - \frac{1}{2}\,\mathrm{col}\{\boldsymbol{\theta}_1^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_1, \,\dots\,, \boldsymbol{\theta}_N^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_N\},\\
\mathrm{Cov}(\boldsymbol{o}'[\,i\,]) &= \boldsymbol{\Theta}^\top \boldsymbol{C}_{0:i}\boldsymbol{\Theta}.
\end{aligned} \tag{E.2}$$

Note that the terms $\{\boldsymbol{\theta}_1^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_1, \,\dots\,, \boldsymbol{\theta}_N^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_N\}$ form the main diagonal of Cov$(()\boldsymbol{o}'[\,i\,])$, and similarly, that $\boldsymbol{\Theta}^\top \boldsymbol{C}_{0:i}\boldsymbol{\theta}_n$ is the $n$th column of Cov$(\boldsymbol{o}'[\,i\,])$.

Next, using the matrix inversion lemma, we expand $\boldsymbol{C}_{0:i}$ to obtain

$$\begin{aligned}
\boldsymbol{C}_{0:i} &= \boldsymbol{U}_{0:i}^\top \big(\boldsymbol{U}_{0:i}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{U}_{0:i}^\top + \boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}\big)^{-1}\boldsymbol{U}_{0:i}\\
&= \boldsymbol{U}_{0:i}^\top \big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}} - \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}\big)^{-1}\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\big)\\
&= \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i} - \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}\big)^{-1}\boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}.
\end{aligned}$$

Again for simplicity, let us call $\boldsymbol{D}_{0:i} = \boldsymbol{U}_{0:i}^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{v}_{0:i}}\boldsymbol{U}_{0:i}$. Thereby, let us develop the last expression above as follows:

$$\begin{aligned}
\boldsymbol{C}_{0:i} &= \boldsymbol{D}_{0:i} - \boldsymbol{D}_{0:i}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}\\
&= \boldsymbol{D}_{0:i} - \boldsymbol{D}_{0:i}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i} - \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}\\
&\qquad + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}\\
&= \boldsymbol{D}_{0:i} - \underbrace{(\boldsymbol{D}_{0:i} + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}})(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i})^{-1}}_{\boldsymbol{I}_{i+1}}\boldsymbol{D}_{0:i} + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}\\
&= \boldsymbol{D}_{0:i} - \boldsymbol{D}_{0:i} + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}\\
&= \boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}}\big(\boldsymbol{\Sigma}^{-1}_{\boldsymbol{w}} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i}.
\end{aligned}$$

We can develop this expression even further:

$$\begin{aligned}\boldsymbol{C}_{0:i} &= \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{D}_{0:i} + \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \\ &\qquad\qquad - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \\ &= \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\underbrace{\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\big(\boldsymbol{D}_{0:i} + \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big)}_{\boldsymbol{I}_{i+1}} - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \\ &= \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}. \end{aligned} \tag{E.3}$$

Thus, we have

$$\boldsymbol{C}_{0:i} = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\Big(\boldsymbol{\Sigma}_{\boldsymbol{w}} - \big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}\Big)\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}. \tag{E.4}$$

In the next steps, we analyze $\boldsymbol{D}_{0:i} = \boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}$. We saw in Equation (4.28) that it can be rewritten as the double sum; i.e.,

$$\boldsymbol{D}_{0:i} = \sum_{j=0}^{i}\sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,j\,]\,\boldsymbol{u}_k^{\top}[\,j\,]}{\sigma_{v_k}^2},$$

which can be rewritten again as follows:

$$\boldsymbol{D}_{0:i} = \sum_{j=0}^{i-1}\sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,j\,]\,\boldsymbol{u}_k^{\top}[\,j\,]}{\sigma_{v_k}^2} + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2} = \boldsymbol{D}_{0:i-1} + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2}.$$

To continue the analysis, define $\boldsymbol{M}[\,i\,] = \big(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\big)^{-1}$. Thereby, note that

$$\begin{aligned}\boldsymbol{M}[\,i\,] &= \left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i-1} + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2}\right)^{-1} \\ &= \left(\boldsymbol{M}^{-1}[\,i-1\,] + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2}\right)^{-1}. \end{aligned} \tag{E.5}$$

Thus, we have

$$\boldsymbol{M}^{-1}[\,i\,] = \boldsymbol{M}^{-1}[\,i-1\,] + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2}. \tag{E.6}$$

Let us also define the matrices $\boldsymbol{M}_k[\,i\,]$ such that

$$\boldsymbol{M}_k^{-1}[\,i\,] = \boldsymbol{M}^{-1}[\,i-1\,] + \sum_{\ell=1}^{k}\frac{\boldsymbol{u}_\ell[\,i\,]\,\boldsymbol{u}_\ell^{\top}[\,i\,]}{\sigma_{v_\ell}^2},\ 1 \leq k \leq K. \tag{E.7}$$

Therefore, from Equation (E.5), we have

$$\begin{aligned}\boldsymbol{M}[\,i\,] &= \left(\boldsymbol{M}^{-1}[\,i-1\,] + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_k}^2}\right)^{-1} \\ &= \left(\boldsymbol{M}^{-1}[\,i-1\,] + \sum_{k=1}^{K-1}\frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_1}^2} + \frac{\boldsymbol{u}_K[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_K}^2}\right)^{-1} \\ &= \left(\boldsymbol{M}_{K-1}^{-1}[\,i\,] + \frac{\boldsymbol{u}_K[\,i\,]\,\boldsymbol{u}_k^{\top}[\,i\,]}{\sigma_{v_K}^2}\right)^{-1}. \end{aligned}$$

Using the Matrix Inversion Lemma, we obtain

$$\begin{aligned}\boldsymbol{M}[\,i\,] &= \boldsymbol{M}_{K-1}[\,i\,] - \boldsymbol{M}_{K-1}[\,i\,]\boldsymbol{u}_K[\,i\,]\big(\sigma^2_{v_K} + \boldsymbol{u}_k^\top[\,i\,]\boldsymbol{M}_{K-1}[\,i\,]\boldsymbol{u}_K[\,i\,]\big)^{-1}\boldsymbol{u}_k^\top[\,i\,]\boldsymbol{M}_{K-1}[\,i\,] \\ &= \left(\boldsymbol{I}_D - \frac{\boldsymbol{M}_{K-1}[\,i\,]\boldsymbol{u}_K[\,i\,]\boldsymbol{u}_k^\top[\,i\,]}{\sigma^2_{v_K} + \boldsymbol{u}_k^\top[\,i\,]\boldsymbol{M}_{K-1}[\,i\,]\boldsymbol{u}_K[\,i\,]}\right)\boldsymbol{M}_{K-1}[\,i\,]. \end{aligned} \tag{E.8}$$

From Equation (E.7), we have for $k = K-1, K-2, \ldots, 2$ that

$$\boldsymbol{M}_k[\,i\,] = \left(\boldsymbol{M}_{k-1}^{-1}[\,i\,] + \frac{\boldsymbol{u}_k[\,i\,]\,\boldsymbol{u}_k^\top[\,i\,]}{\sigma^2_{v_k}}\right)^{-1},$$

and for $k = 1$, we have

$$\boldsymbol{M}_1[\,i\,] = \left(\boldsymbol{M}[\,i-1\,] + \frac{\boldsymbol{u}_1[\,i\,]\,\boldsymbol{u}_1^\top[\,i\,]}{\sigma^2_{v_1}}\right)^{-1}.$$

Thus, we can do the same process that resulted in Equation (E.8) to find each $\boldsymbol{M}_k[\,i\,]$, which results in

$$\boldsymbol{M}_k[\,i\,] = \left(\boldsymbol{I}_D - \frac{\boldsymbol{M}_{k-1}[\,i\,]\boldsymbol{u}_k[\,i\,]\boldsymbol{u}_k^\top[\,i\,]}{\sigma^2_{v_k} + \boldsymbol{u}_k^\top[\,i\,]\boldsymbol{M}_{k-1}[\,i\,]\boldsymbol{u}_k[\,i\,]}\right)^{-1}\boldsymbol{M}_{k-1}[\,i\,],$$

which is valid for $k = 1, \ldots, K$ if noting, as a consequence of Equation (E.7), that we have

$$\boldsymbol{M}^{-1}[\,i\,] = \boldsymbol{M}_K^{-1}[\,i\,] \Longrightarrow \boldsymbol{M}[\,i\,] = \boldsymbol{M}_K[\,i\,],$$

and also by following the convention that

$$\boldsymbol{M}_0[\,i\,] = \boldsymbol{M}[\,i-1\,].$$

Now, what remains is to find a proper initial value $\boldsymbol{M}[-1]$. From Equation (E.6), we have

$$\boldsymbol{M}^{-1}[\,0\,] = \boldsymbol{M}^{-1}[-1] + \sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,0\,]\,\boldsymbol{u}_k^\top[\,0\,]}{\sigma^2_{v_k}} \Longrightarrow \boldsymbol{M}[\,0\,] = \left(\boldsymbol{M}^{-1}[-1] + \boldsymbol{D}_{0:0}\right)^{-1};$$

Therefore, comparing with the definition $\boldsymbol{M}[\,i\,] = \left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\right)^{-1}$, we must have

$$\boldsymbol{M}^{-1}[-1] = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} \Longrightarrow \boldsymbol{M}[-1] = \boldsymbol{\Sigma}_{\boldsymbol{w}}.$$

Returning to Equation (E.4), we can thus write

$$\boldsymbol{C}_{0:i} = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{M}[\,i\,])\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1},$$

and defining $\boldsymbol{\Theta}' = \boldsymbol{\Theta}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}$ (recall that $\boldsymbol{\Theta} = [\,\boldsymbol{\theta}_1\ \ldots\ \boldsymbol{\theta}_N\,]$), we can rewrite Equation (E.2) as

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{o}'[\,i\,]) &= (\boldsymbol{\Theta}')^\top(\boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{M}[\,i\,])\,\boldsymbol{\Theta}', \\ \mathrm{E}(\boldsymbol{o}'[\,i\,] \mid H_n) &= \big(\mathrm{Cov}(\boldsymbol{o}'[\,i\,])\big)_{mn,\,1\le m\le N} - \frac{1}{2}\big(\mathrm{Cov}(\boldsymbol{o}'[\,i\,])\big)_{mm,\,1\le m\le N}, \end{aligned} \tag{E.9}$$

where $\big(\mathrm{Cov}(\boldsymbol{o}'[\,i\,])\big)_{mn,\,1\le m\le N}$ denotes the entries of the $n$th column of $\mathrm{Cov}(\boldsymbol{o}'[\,i\,])$, and similarly $\big(\mathrm{Cov}(\boldsymbol{o}'[\,i\,])\big)_{mm,\,1\le m\le N}$ denotes the entries of the main diagonal of $\mathrm{Cov}(\boldsymbol{o}'[\,i\,])$. Therefore, all we need to estimate the performance of the optimal detector at each time instant $i$ is its covariance matrix $\mathrm{Cov}(\boldsymbol{o}'[\,i\,])$, which can be done by numerical Monte Carlo (MC) trials with a number $L$ of realizations.

We synthesize all the last steps in Algorithm 3 below.

**Algorithm 3** – Estimation of Optimal detector performance.

```
Inputs: $\sigma^2_{v_k}$, $\boldsymbol{u}_k[\,i\,]$, $\{P(H_n)\}_{n=1}^{N}$, $\boldsymbol{\Theta}'$, $\boldsymbol{\Sigma}_{\boldsymbol{w}}$, $L$, $i_{\max}$.
Initialize $i = 0$, $\boldsymbol{M} = \boldsymbol{\Sigma}_{\boldsymbol{w}}$.
while $i < i_{\max}$ do
    for $k = 1, \dots, K$ do                                        Equation (E.8)
        $\boldsymbol{m}_k \leftarrow \boldsymbol{M}\boldsymbol{u}_k[\,i\,]$.
        $\boldsymbol{m}'_k \leftarrow \boldsymbol{m}_k/(\sigma^2_{v_k} + \boldsymbol{u}_k^{\top}[\,i\,]\boldsymbol{m}_k)$.
        $\boldsymbol{M} \leftarrow (\boldsymbol{I}_D - \boldsymbol{m}'_k\boldsymbol{u}_k^{\top}[\,i\,])\boldsymbol{M}$.
    end for
    $\mathrm{Cov}(\boldsymbol{o}') \leftarrow (\boldsymbol{\Theta}')^{\top}(\boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{M})\,\boldsymbol{\Theta}'$.      Equation (E.9)
    $\xi \leftarrow \mathrm{MC}(\mathrm{Cov}(\boldsymbol{o}'),\ \{P(H_n)\}_{n=1}^{N},\ L)$.
    $i \leftarrow i + 1$
end while
```

## Lower Bound of the probability of error

The minimum probability of error that is achievable by a distributed detector in a given setting is the asymptotic probability of error of the optimal detector. Therefore, in order to estimate it we need the asymptotic distribution of the optimal statistic $o_n[\,i\,]$, given in Equation (4.37).

In the previous section it was shown that the performance of the optimal detector depends basically on the covariance matrix of the equivalent optimal statistic $o'_n[\,i\,]$, defined in Equation (E.1). From Equation (E.2), we have that $\mathrm{Cov}(\boldsymbol{o}'[\,i\,]) = \boldsymbol{\Theta}^{\top}\boldsymbol{C}_{0:i}\boldsymbol{\Theta}$, where $\boldsymbol{\Theta} = [\,\boldsymbol{\theta}_1\ \dots\ \boldsymbol{\theta}_N]$ and $\boldsymbol{C}_{0:i} = \boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}$. It was shown in Equation (E.3) that

$$\boldsymbol{C}_{0:i} = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1},$$

where $\boldsymbol{D}_{0:i} = \boldsymbol{U}_{0:i}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{v}_{0:i}}^{-1}\boldsymbol{U}_{0:i}$, which can be written as the double sum as showed in Equation (4.28) in Chapter 5. Thereby, we can rewrite Equation (4.28) as

$$\sum_{j=0}^{i}\sum_{k=1}^{K}\frac{\boldsymbol{u}_k[\,j\,]\,\boldsymbol{u}_k^{\top}[\,j\,]}{\sigma^2_{v_k}} = (i+1)\sum_{k=1}^{K}\frac{1}{\sigma^2_{v_k}}\sum_{j=0}^{i}\frac{\boldsymbol{u}_k[\,j\,]\,\boldsymbol{u}_k^{\top}[\,j\,]}{i+1},$$

Since the innermost sum above converges to $\boldsymbol{\Sigma}_{\boldsymbol{u}_k}$ as $i$ increases, because we assume also in Chapter 5 that $\underset{\sim}{\boldsymbol{u}}_k[\,i\,] \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}_k}\big)$, the outermost sum converges to a nonsingular matrix as $i$ increases, as it is a sum of positive definite matrices. Therefore, $\boldsymbol{D}_{0:i}$ is nonsingular and its terms become larger as $i$ increases. Thus, as $i \to \infty$, $\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \boldsymbol{D}_{0:i}\right)^{-1} \to \boldsymbol{0}$, and consequently $\boldsymbol{C}_{0:i} \to \boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}$. Hence, we can use this result to estimate, via a Monte Carlo simulation, the asymptotic probability of error of the optimal detector.