UNIVERSIDAD CARLOS III DE MADRID
Departamento de Teoría de la Señal y Comunicaciones

DOCTORAL THESIS

ALGORITHMS FOR ENERGY-EFFICIENT ADAPTIVE WIRELESS SENSOR NETWORKS

Author: JESÚS FERNÁNDEZ BES

Supervised by: JESÚS CID SUEIRO
JERÓNIMO ARENAS GARCÍA

JUNE 2015

Tesis Doctoral:	ALGORITHMS FOR ENERGY-EFFICIENT ADAPTIVE WIRELESS SENSOR NETWORKS
Autor:	Jesús Fernández Bes
Directores:	Jesús Cid Sueiro Jerónimo Arenas García
Fecha:	18 de Junio 2015

Tribunal

Presidente: Santiago Zazo Bello

Vocal: Alexander Bertrand

Secretario: Miguel Lazaro Gredilla

Al Abuelo Bes
Por haberme atraído desde muy pequeño al mundo del cacharreo

This work was partially supported by the "Formación de Profesorado Universitario" fellowship from the Spanish Ministry of Education (FPU AP2010-5225).

Agradecimientos

Wahrlich es ist nicht das Wissen, sondern das Lernen, nicht das Besitzen sondern das Erwerben, nicht das Da-Seyn, sondern das Hinkommen, was den grössten Genuss gewährt.

No es el conocimiento, sino el acto de aprendizaje; y no la posesión, sino el acto de llegar a ella, lo que concede el mayor disfrute.

Karl Friedrich Gauß

Me resulta muy difícil agradecer en unas pocas líneas a todo el mundo que de una forma directa o indirecta ha contribuido a esta tesis. Pero vamos a ello.

En primer lugar tengo que agradecer a mis dos tutores Jesús y Jerónimo por su inmensa contribución a esta tesis. Aunque suene a tópico, la mayor parte de los aciertos de este trabajo se les debe atribuir a ellos y la mayor parte de los errores a mi mismo.

Casi al mismo nivel que a mis tutores tengo que poner a Rocío, Magno, Antonio y especialmente a Luis. Si este trabajo tiene algo de calidad es debido a su capacidad de trabajo y a lo involucrados que han estado en él. Espero que sigamos colaborando.

I would also like to thank Professor Ali Sayed and his team: Zaid, Jianshu, Xiaochuan, Chung-Kai, and Shang-Kee for their warm hospitality and the valuable discussions we had in the Lab whiteboard.

Al resto de la gente del departamento de Teoría de la Señal de la UC3M y muy especialmente a los miembros del G2PI y GTS (gracias chicos por haberme dejado "infiltrarme" tan amablemente). Lo siento mucho pero no os voy a nombrar porque me juego olvidarme de alguien. Si voy a nombrar especialmente a Pablo "El bueno", Fran, Isa, Wilton y Alex, por aguantarme tantas horas en el día a día en el 4.3.A.03, y a Víctor por aguantarme (pocas horas al día) en el salón de casa. Compañero nos queda mucho por recorrer.

Y como *la ironía es la forma más alta de sinceridad*, también tengo que agradecer al Ministerio de Educación, a la Oficina de Posgrado de la universidad y demás organismos públicos con los que he tenido la suerte de tratar durante estos años. Su voluntad de servicio, atención y eficiencia ha sido una ayuda impagable para la consecución de esta tesis.

Me voy metiendo poco a poco en lo personal y en este momento no puedo evitar acordarme de gente que ha estado cerca de mi desde hace muchos años. A los pocos compañeros de Teleco que aún considero amigos: Juan, Carlos y Darío (espero que nos sigamos viendo por los más diversos lugares de Europa y que Juan siga teniendo aventuras que contar). A mis niñas del Erasmus: Almu, Moni y Cris; ya me tenéis que llamar Dr. Je.

En este momento no puedo evitar acordarme, en orden no demasiado coherente, de: Víctor, Marta, Germán, Santi, Almu, Gerard, Pablo, Gema, Antonio, David, Jose, Rodrigo, Paco, Virgina, Jorge y Pepi. Después de tantos años algo habréis influido en esta tesis (probablemente para mal) porque mucho de lo que soy tiene que ver con vosotros y con lo que hemos vivido desde el Amor de Dios "campo de concentración" al día de hoy.

Llegando al *top ten* no puedo olvidarme de mis primos: Beto y Tania (+ Axel), Ángel, Kike, Miguel y Pablo con los que tengo algunos de los recuerdos más dulces de mi vida. Ni de mi tía María José con la que se que siempre podré contar.

La única forma posible de cerrar este estas líneas es agradeciendo a mis hermanos: Alfonso y Carlos y a mis padres. Gracias por estar allí, aunque no lo suelo decir os quiero. Y por último, Paula, el tiempo que he pasado trabajando en esta tesis va en paralelo con tiempo que he pasado junto a ti. Me has hecho muy feliz estos años y esto es sólo el principio de una vida juntos.

Abstract

In this thesis we focus on the development of *energy-efficient adaptive algorithms for Wireless Sensor Networks*. Its contributions can be arranged in two main lines.

Firstly, we focus on the efficient management of energy resources in WSNs equipped with finite-size batteries and energy-harvesting devices. To that end, we propose a censoring scheme by which the nodes are able to decide if a message transmission is worthy or not given their energetic condition. In order to do so, we model the system using a Markov Decision Process and use this model to derive optimal policies. Later, these policies are analyzed in simplified scenarios in order to get insights of their features. Finally, using Stochastic Approximation, we develop low-complexity censoring algorithms that approximate the optimal policy, with less computational complexity and faster convergence speed than other approaches such as Q-learning.

Secondly, we propose a novel diffusion scheme for adaptive distributed estimation in WSNs. This strategy, which we call Decoupled Adapt-then-Combine (D-ATC), is based on keeping an estimate that each node adapts using purely local information and then combines with the diffused estimations by other nodes in its neighborhood. Our strategy, which is specially suitable for heterogeneous networks, is theoretically analyzed using two different techniques: the classical procedure for transient analysis of adaptive systems and the energy conservation method. Later, as using different combination rules in the transient and steady-state regime is needed to obtain the best performance, we propose two adaptive rules to learn the combination coefficients that are useful for our diffusion strategy. Several experiments simulating both stationary estimation and tracking problems show that our method outperforms state-of-the-art techniques in relevant scenarios. Some of these simulations reveal the robustness of our scheme under node failures.

Finally, we show that both approaches can be combined in a common setup: a WSN composed of harvesting nodes aiming to solve an adaptive distributed estimation problem. As a result, a censoring scheme is added on top of D-ATC. We show how our censoring approach helps to improve both steady-state and convergence performance of the diffusion scheme.

Resumen

La presente tesis se centra en el desarrollo de *algoritmos adaptativos energéticamente eficientes para redes de sensores inalámbricos.* Sus contribuciones se pueden englobar en dos líneas principales.

Por un lado, estudiamos el problema de la gestión eficiente de recursos energéticos en redes de sensores equipadas con dispositivos de captación de energía y baterías finitas. Para ello, proponemos un esquema de censura mediante el cual, en un momento dado, un nodo es capaz de decidir si la transmisión de un mensaje merece la pena en las condiciones energéticas actuales. El sistema se modela mediante un Proceso de Decisión de Markov (Markov Decision Process, MDP) de horizonte infinito y dicho modelo nos sirve para derivar políticas óptimas de censura bajo ciertos supuestos. Después, analizamos estas políticas óptimas en escenarios simplificados para extraer intuiciones sobre las mismas. Por último, mediante técnicas de Aproximación Estocástica, desarrollamos algoritmos de censura de menor complejidad que aproximan estas políticas óptimas. Las numerosas simulaciones realizadas muestran que estas aproximaciones son competitivas, obteniendo una mayor tasa de convergencia y mejores prestaciones que otras técnicas del estado del arte como las basadas en *Q-learning.*

Por otro lado, proponemos un nuevo esquema de difusión para estimación distribuida adaptativa. Esta estrategia, que denominamos *Decoupled Adapt-then-Combine* (D-ATC), se basa en mantener una estimación que cada nodo adapta con información puramente local y que posteriormente combina con las estimaciones difundidas por los demás nodos de la vecindad. Analizamos teóricamente nuestra estrategia, que es especialmente útil en redes heterogéneas, usando dos métodos diferentes: el método clásico para el análisis de régimen transitorio en sistemas adaptativos y el método de conservación de la energía. Posteriormente, y dado que para obtener el mejor rendimiento es necesario utilizar reglas de combinación diferentes en el transitorio y en régimen permanente, proponemos dos reglas adaptativas para el aprendizaje de los pesos de combinación para nuestra estrategia de difusión. La primera de ellas está

basada en una aproximación de mínimos cuadrados (*least-squares*, LS); mientras que la segunda se basa en el algoritmo de proyecciones afines (*Affine Projection Algorithm*, APA). Se han realizado numerosos experimentos tanto en escenarios estacionarios como de seguimiento que muestran cómo nuestra estrategia supera en prestaciones a otras aproximaciones del estado del arte. Algunas de estas simulaciones revelan además la robustez de nuestra estrategia ante errores en los nodos de la red.

Por último, mostramos que estas dos aproximaciones son complementarias y las combinamos en mismo escenario: una red de sensores inalámbricos compuesta de nodos equipados con dispositivos de captación energética cuyo objetivo es resolver de manera distribuida y adaptativa un problema de estimación. Para ello, añadimos la capacidad de censurar mensajes a nuestro esquema D-ATC. Nuestras simulaciones muestran que la censura puede ser beneficiosa para mejorar tanto el rendimiento en régimen permanente como la tasa de convergencia en escenarios relevantes de estimación basada en difusión.

Contents

Foreword **vii**

List of Figures **xvi**

List of Tables **xix**

1 Introduction **1**

1.1 Background 1

1.1.1 Wireless sensor networks 2

1.1.2 Applications of wireless sensor networks 5

1.1.3 Wireless sensor network constraints 7

1.2 Motivation 9

1.2.1 Extending the network lifetime 9

1.2.2 Censoring schemes in WSNs 10

1.2.3 Harvesting devices 11

1.2.4 Distributed signal processing 11

1.3 Objectives and Main contributions 12

1.4 Thesis organization 15

2 MDP models for censoring in harvesting sensor networks **17**

2.1 State-of-the-art in energy management in harvesting sensor networks . 18

2.2 System model 21

2.2.1 Notation 21

2.2.2 State vector 21

2.2.3 Actions and policies 24

2.2.4 State dynamics . . . 24
2.2.5 Rewards . . . 25
2.2.6 Problem formulation . . . 25
2.3 Optimal stationary policy . . . 26
2.4 Analysis of the optimal policy . . . 30
2.4.1 Optimal policies for particular cases . . . 31
2.4.2 Asymptotic behavior: steady-state distributions . . . 33
2.5 Stochastic approximate schemes . . . 38
2.5.1 A stochastic approximation to the optimal policy . . . 38
2.5.2 Q-learning . . . 41
2.5.3 Adaptive balanced transmitter . . . 42
2.6 Simulation results . . . 44
2.6.1 Single-hop network . . . 45
2.6.2 Multi-hop networks . . . 50
2.7 Summary . . . 51

3 Decoupled diffusion for adaptive distributed estimation in WSN 53
3.1 State-of-the-art in adaptive networks . . . 54
3.2 Decoupled Adapt-then-Combine diffusion scheme . . . 56
3.2.1 Notation . . . 56
3.2.2 Description of the diffusion strategy . . . 56
3.2.3 Comparison between ATC and D-ATC . . . 60
3.3 Theoretical analysis of D-ATC . . . 62
3.3.1 Data model and definitions . . . 63
3.3.2 Statistical transient analysis . . . 64
3.3.3 Energy conservation analysis . . . 67
3.4 Adaptive combiners for D-ATC scheme . . . 74
3.4.1 Affine projection algorithm . . . 75
3.4.2 Least-Squares algorithm . . . 76
3.5 Simulation results . . . 79
3.5.1 Validation of the theoretical analysis for D-ATC . . . 80
3.5.2 Stationary performance of D-ATC with adaptive combiners . . . 82
3.5.3 Tracking performance of D-ATC with adaptive combiners . . . 84

3.5.4 Performance under node failures 85
3.6 Summary 86

4 Censoring in adaptive diffusion networks 89
4.1 Signal and energy Model 90
4.2 Assignment of importance 91
4.3 Censoring algorithm 92
4.4 Simulation Results 94
4.5 Summary 96

5 Conclusions and future work 97
5.1 Summary of contributions 97
5.2 Discussion 99
5.3 Future work 100

A Proofs and Derivations of Chapter 2 105
A.1 Proof of Theorem 1 105
A.2 Derivation of Transition probability matrix 106
A.3 Derivation of the stochastic algorithm 107

B Proofs and Derivations of Chapter 3 109

C Acronyms and abbreviations 115

D List of Publications 117

Bibliography 121

List of Figures

1.1 Sensor node architecture 2
1.2 Examples of sensor nodes 3
1.3 Typical WSN topology 4
1.4 Example of applications of WSNs 7

2.1 Data and energy operation model 23
2.2 Graphical model relating the variables of the MDP. 27
2.3 Optimal thresholds and value function for different $\bar{b}_0$ 34
2.4 Optimal thresholds and value function for different B 35
2.5 Expected Performance for a scenario with high refill rate 37
2.6 Expected Performance for a scenario with low refill rate 38
2.7 Comparison between the SAP, ABT, and Q-learning algorithms 46
2.8 Performance of a single-hop stationary scenario. 47
2.9 Theoretical and estimated thresholds of single-hop scenario 48
2.10 Battery evolution in a non-stationary environment 49
2.11 Example of network routing topologies 50
2.12 Performance for multi-hop scenarios 52

3.1 Example of diffusion network. 58
3.2 Block diagram of Decoupled ATC diffusion stategy. 59
3.3 Steady-state Network MSD for 2 nodes 62
3.4 Network MSD convergence rate for 2 nodes 62
3.5 Network topology and parameters for simulation experiments 79

3.6 Comparison between theoretical transient model and empirical performance . . . 81
3.7 Comparison between energy conservation theoretical model and empirical performance . . . 82
3.8 Comparison between theoretical and empirical performance in a tracking scenario . . . 83
3.9 Influence of the parameters of the adaptive combiner algorithms. . . . 84
3.10 Network MSD performance for a stationary estimation problem. . . . 86
3.11 Network MSD performance for a tracking problem. . . . 86
3.12 Network MSD for the complex network when nodes are subject to random failures at every iteration, with failure rate: (a) 30%, (b) 50%, and (c) 80%. . . . 87

4.1 Network topology and noise variance for C-DATC simulation experiments 94
4.2 Network MSD for two different harvesting scenarios . . . 95
4.3 Steady-state MSD for two different harvesting scenarios . . . 95
4.4 Censoring threshold evolution . . . 96

List of Tables

1.1 Sensor node platforms and characteristics 3

2.1 Summary of the notation used in Chapter 2 22
2.2 Stochastic approximate policy (SAP) algorithm 41
2.3 Q-learning algorithm 43
2.4 Adaptive Balanced Transmitter (ABT) algorithm 44
2.5 Performance in a non-stationary environment. 49

3.1 Summary of the notation used in Chapter 3 57
3.2 Settings of the scenarios simulated in Fig. 3.7 81
3.3 Parameters of the adaptive combiners algorithms. 85

4.1 Censoring D-ATC scheme 93

1
Introduction

In this chapter, we start providing an overview of Wireless Sensor Network (WSN) technologies and applications. Later, we focus on their limitations and constraints as a motivation for the presented work. Then, the main contributions of the thesis dissertation are introduced. Finally, we include the organization of the dissertation as a point of reference for readers.

1.1 Background

In 2003, the Massachusetts Institute of Technology classified Wireless Sensor Networks (WSNs) as one of the top ten emerging technologies that would change the world [87]. It is their capability to provide distributed, real-time interaction with the physical world which has attracted attention from a wide range of disciplines. Since then, WSNs are slowly becoming an integral part of our lives and paradigms such as Internet of Things (IoT) or Smart Cities, which are technically supported by WSNs, have been classified as strategic development areas by the European Union [38, 39].

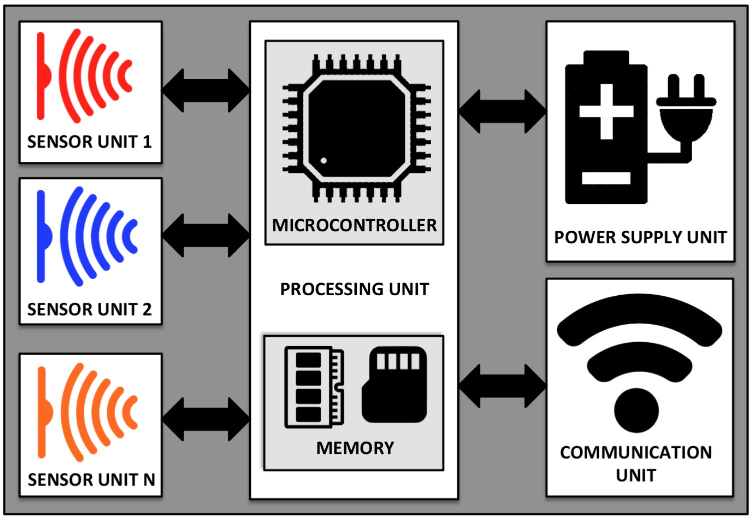
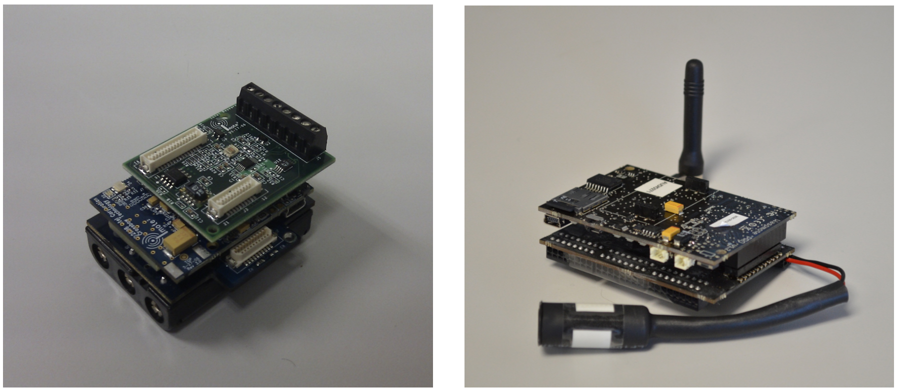


Figure 1.1: Sensor node architecture. A sensor node comprises 4 subsystems: A processing unit, several kinds of sensor units to collect data, a communication unit and a power supply, usually a battery.

Nevertheless, the realization of existing and potential applications of WSNs requires the development of technologies at least in three different research areas: sensing, communication and computation (including hardware, software, and processing algorithms). Moreover, the combined advancement of these three areas is fundamental in order to fully exploit this technology in a variety of application domains.

1.1.1 Wireless sensor networks

A sensor node is an autonomous electronic device with embedded sensing, data processing and communication capabilities. The typical architecture of a sensor node, shown in Fig. 1.1, comprises four subsystems: a processing unit, a communication unit, a power supply unit —usually a battery— and one or more sensing units. The diversity of available sensors for this kind of platforms is huge: temperature, humidity, pressure, light, sound, vibration, motion, radiation, chemicals [2, 71]; enabling a significant number of potential applications. Nodes work as information sources which interact with the physical environment and sense, measure, or gather detailed information from some physical entities of interest, performing, when required, simple



(a) Crossbow Imote2 (b) Libelium Waspmote

Figure 1.2: Examples of sensor nodes. (a) Crossbow Imote with a battery module; (b) Libellium Waspmote with acoustic sensor.

processing on the extracted data, and transmitting it to remote locations. Table 1.1, which has been adapted from [2], displays a summary of the characteristics of some popular sensor platforms. The communication units of all platforms are based on the IEEE 802.15.4 standard [12, 64] for low-power wireless personal area networks, some notable exceptions being *Mica2*, which appeared before the standard, and *Shimmer3*, based on Bluetooth. Figure 1.2 displays some examples of node platforms.

Table 1.1: Sensor node platforms and characteristics

Mote Type	**Manufacturer**	**CPU**	**RAM**	**Radio Freq.**	**(kbps)**
Mica2	Crossbow	16 MHz	4 kB	433/868/916 MHz	38.4 kbps
MicaZ	Crossbow	16 MHz	4 kB	2.4 GHz	250 kbps
Tmote	Sentilla	16 MHz	10 kB	2.4 GHz	250 kbps
Imote2	Crossbow	13-416 MHz	256 kB	2.4 GHz	250 kbps
Shimmer3 [111]	Shimmer	24 MHz	16 kB	BT/2.4 GHz	1 Mbps
WaspMote [70]	Libelium	14 MHz	8 kB	2.4 GHz	250 kbps

Hence, a Wireless Sensor Network is the collection of a variable amount of interconnected sensor nodes, ranging from hundreds to thousands, deployed either directly inside the phenomenon of interest or close to it [2]. Such sensors can be scattered on the ground, underground, in the air, under water, in vehicles, inside human bodies or embedded in structures or buildings. Once sensor nodes are deployed, they are

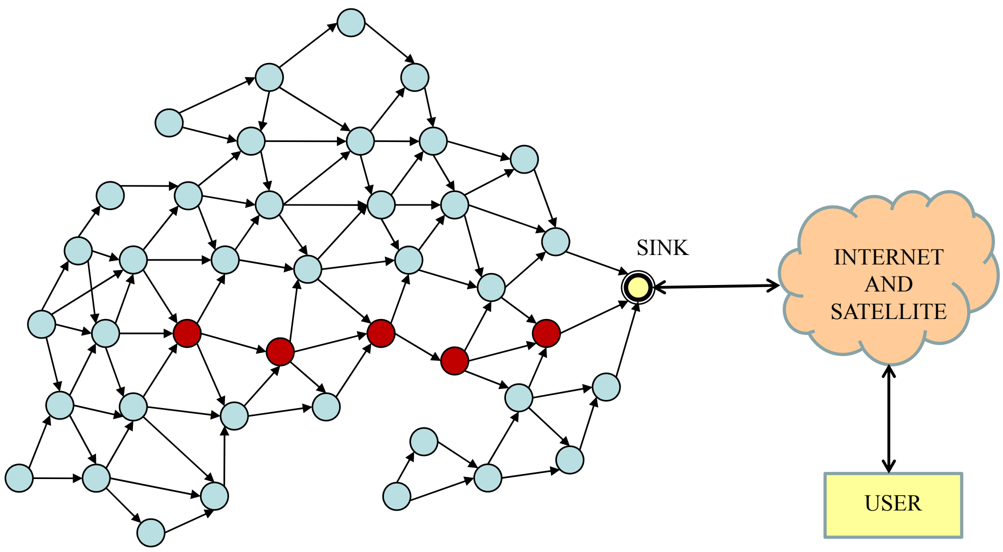


Figure 1.3: Typical WSN topology. In this kind of topology there is one or various sink nodes to which all the other nodes send their data in a multi-hop manner.

supposed to require minimal support for their functioning. This minimal support forces sensor network protocols and algorithms to possess self-organizing capabilities. Nodes should be capable of organizing themselves into a network, forming a (possibly dynamic) topology, and being able to control it.

In the typical WSN topology, see Fig. 1.3, there exists a destination node (also known as Fusion Center or *sink*) which collects all the data sensed by the rest of the nodes. This sink node is usually a more powerful device and is able to communicate with the user through conventional network services, e.g., the Internet. As seen in Fig. 1.3, sensor nodes do not work independently, but serve as relays of other nodes in the network. Since nodes are often resource-constrained, a considerable reduction in transmitting power can be easily obtained from node cooperation. That is the reason why most WSNs are multi-hop networks.

Apart from just relaying information to the sink node, there can be a further level of cooperation: nodes can process their data cooperatively, combining information from multiple sources. This cooperation introduces the concept of distributed computing and processing. Thanks to their processing capabilities, nodes can transmit partially processed data instead of raw data, removing the redundant information of the captured data through in-network aggregation and local compression. Such collaboration among sensors, can lead to topologies where there is no sink node anymore and all information is processed inside the network, which comes out as a more fault-tolerant approach.

1.1.2 Applications of wireless sensor networks

In the last years, due to the variety of sensors that can be incorporated into WSNs, many academic and commercial applications have been studied. This section is just a brief overview of these applications to provide readers with some insights of the potentials of WSNs. We will categorize the spectrum of applications into six categories:

- **Environmental.** An interesting area of application is environmental monitoring. WSNs allow us to obtain large amount of data of a natural phenomenon in wide or difficult to access areas. The range of applications in this scope goes from precision agriculture [22, 23] and animal tracking [137] to meteorological and pollution studies [32] or planetary exploration [95].

- **Industry.** Networks of wired sensors have long been used in industrial plants. However, the cost of deployment of these sensors has limited their applications. Consequently there is a benefit from turning to a wireless system. Some commercial applications of WSNs are product quality monitoring, inventory management, factory process control, or real-time nuclear plant monitoring [72, 76].

- **Health.** There is also a growing interest in biomedical applications of WSNs. The development of implanted devices and smart wearable sensors opens the door to applications such as patient monitoring [79], diagnostics [40], or drug administration [30].

- **Military.** WSNs can also be an integral part of military *Command and Control* (C2) systems. The rapid deployment, self-organization and fault tolerance characteristics of WSNs make them a very promising technology in a number of C2 applications. Some of them are monitoring friendly forces, battlefield surveillance, reconnaissance of opposing forces or nuclear, biological and chemical attack detection. Some successful examples are the *Smart Dust* DARPA project [126], Sniper Detection system [37], or the *VigilNet* project [58].

- **Home.** Apart from the most obvious surveillance applications, there are also interesting WSNs' applications related to Smart Grid in the household. WSNs can provide end-users with more information about their appliances usage or water and electricity consumption and consequently improve energy, gas and water efficiency at home [50].

- **Smart city.** The aim of *smart cities* is to make a more efficient use of town telecommunication, energy and transport resources for the benefit of its inhabitants and businesses. Some successful use cases of WSNs in this area are *smart parking*, monitoring of parking spaces availability in the city [56]; structural health monitoring of vibrations and material conditions in buildings, bridges and historical monuments [68]; or Smart Lighting, intelligent and weather adaptive illumination in street lights [25].

In Fig. 1.4, you can find visual examples of some of these applications: An application of stress monitoring with a *Shimmer 3 GSR* Unit [111], part of an ongoing work by Francisco Hernando-Gallego [59]; an Smart Parking system that informs about available spaces in the streets in downtown Santander (Spain)[18, 56]; a structural health monitoring system in San Francisco (USA) which measures vibrations on Golden Gate Bridge [68]; and an animal tracking project, ZebraNet, which has produced significant improvements in WSNs algorithms and protocols [137].

Whatever the application, WSNs must solve any of the following main problems:

- **Detection problem.** In many cases the detection of a particular (usually rare) event is the initial step before any other type of processing. Sometimes it is even the final objective of the network, for example in a surveillance or monitoring system, where an alarm is turned on if a particular event is detected.

- **Estimation problem.** In other cases, the objective of WSNs is to estimate from the collected measurements the state of some variables of interest, e.g., the position of some target or the parameters needed to map a spatially sampled

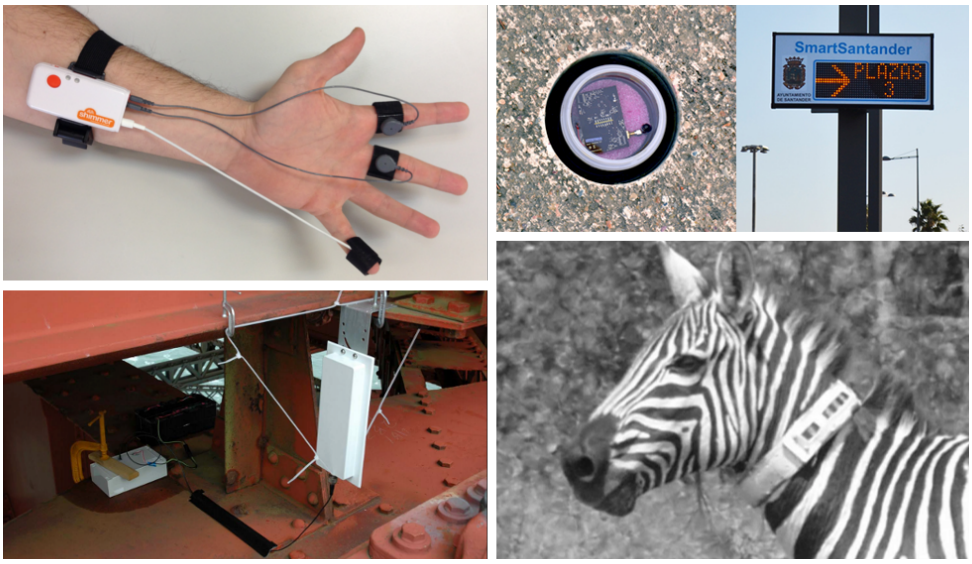


Figure 1.4: Example of applications of WSNs. **Top left:** Stress monitoring with Galvanic Skin Response (GSR) and Blood Volume Pressure (BVP) sensors. **Top right:** Smart Parking in Santander (Spain) taken from [18]. **Bottom left:** Structural monitoring of Golden Gate Bridge, San Francisco (USA) taken from [68]. **Bottom right:** ZebraNet animal tracking project taken from [137].

field. In addition, such state may change in time, in which case the network has to deal with a tracking problem, and should be able to timely react to those dynamics.

Detection and *estimation* are, in fact, the two main problems in classical statistical signal processing [93], but the constrained resources and distributed nature of WSNs introduce new challenges to that topic. The techniques presented in this thesis could be applied to detection problems, as we did in [46], but in this dissertation we will focus on estimation scenarios.

1.1.3 Wireless sensor network constraints

Wireless Sensor Networks have a number of limitations due to constrained resources of sensor nodes. In this section, we describe some of these constraints with special focus in energy as a crucial limiting resource.

Computation and memory constraints

Analyzing the characteristics of some popular sensor node platforms, showed in Table 1.1, we can see that the standard CPU speed and RAM memory of these nodes is extremely limited. Although some of them have the possibility of adding some extra Flash memory, the complexity of the algorithms that run in the nodes is significantly constrained both in computation power and memory. This limitation forces the network to communicate with more powerful devices to handle heavy computations unless distributed processing algorithms, where the computation burden is shared among the nodes, are developed.

Communication constraints

Most sensor nodes communicate using the IEEE 802.15.4 standard in the physical and Medium Access Control (MAC) layers and the Zigbee specification in the network layer [64, 12]. Both standards are conceived for low-power and, consequently, short-range communications. The typical range in IEEE 802.15.4 is around 100 m, so it is usually unfeasible to have a direct link between all the sensor nodes and the sink when we have a WSN covering a wide area. Consequently most WSNs are multi-hop networks. Similarly, when developing distributed processing algorithms, it must be taken into account that this short range also limits the number of neighbors of a node, i.e., the number of other nodes to which it can communicate directly.

Energy constraints

Even if one is able to overcome the previous limitations there is an additional problem in WSNs: They are usually composed of battery-powered nodes. That means that the operational lifetime of the network is limited when the batteries cannot be easily replaced. In some cases, this replacement is impractical because of the huge number of sensors which form the network, in other cases it is even unfeasible, e.g. when nodes are deployed underground, under water, or embedded in building structures [99].

1.2 Motivation

From the overview above we find a number of limitations that restrain the further development of applications of WSNs. The following approaches could help to overcome those limitations.

1.2.1 Extending the network lifetime

It is known that even with the standard low-power protocols of WSNs, the most energy demanding task in WSNs is communication [99]. Because of that, a great number of methods to minimize the energy expenditure due to communication processes have been proposed in the literature. Most of this research is oriented to produce energy-efficient hardware, building ultra-low-power microcontrollers, and energy-aware communication protocols [99]. Significant gain can be obtained by designing efficient physical layer protocols, e.g., designing energy-efficient modulations [110] and correctly allocating power [138]. Regarding the MAC layer, energy can be saved with low duty-cycle MAC protocols, i.e., nodes are put in a low-power sleep mode whenever communication is not needed. There are two obvious trade-offs here. Firstly, if the sleep-wake pattern is very fast, a significant amount of energy is consumed by just these switchings, making this strategy inefficient. Secondly, if the nodes are left in sleep mode for a long time they may not gather some data that could be important for the application. That makes the design of these protocols a difficult task, but there have been some successful proposals such as LEACH [129], TRAMA [101] or S-MAC [133]. From the network layer perspective, energy saving can be obtained by designing routing protocols that take energy cost into account, and not only the network throughput or delay, see [1] for a detailed review of routing algorithms for WSNs.

All these approaches are in some way *Cross-Layer*, i.e., the algorithms make use of information fed-back from different layers of the communication model to improve performance. Since WSNs are not general purpose networks, but conceived with a particular application in mind, it makes sense to use application-based metrics in the optimization of lower-level protocols. Some works explicitly follow this approach [3, 78, 121]. In all of them the physical, MAC, and network layer protocols are jointly built following some higher level metric, usually a balance between some measure of network performance and network lifetime.

Finally, a complementary approach to improve the energy efficiency in WSNs consists in carrying out a more efficient and intelligent data processing. The simplest approach is *data aggregation*, i.e., nodes can aggregate their sensed or received data to reduce the number of communicated packages [4]. In more advanced schemes, nodes can integrate the received information in a distributed processing scenario [74, 130]. In addition, nodes can be intelligent enough to censor their own measurements. These last two approaches are explained in detail as they constitute, together with the introduction of the disruptive technology of harvesting nodes, the starting point for this thesis.

1.2.2 Censoring schemes in WSNs

Not all the messages in a WSN have the same importance. In many practical scenarios, it makes sense to attribute a particular significance, priority, relevance, utility or *Quality of Information* (QoI) [136] value to the messages in the network. Consequently, in order to enlarge the network lifetime and optimize its performance, sensors nodes could weigh up: a) The potential benefits of transmitting information and b) the energy cost of the subsequent communication process.

Probably, one of the first works that took this approach is [100] by Rago *et al.* In that work, a censoring strategy for distributed detection in (wired) radar networks was proposed. This was a novel approach with respect to previous works that proposed different compression strategies for similar scenarios [115, 124]. In such scenarios the scarcest resource was bandwidth but in [6] Appadwedula *et al.* applied a similar censoring strategy for decentralized detection in WSNs, already with energy efficiency in mind.

In the last years, there has been significant interest in censoring low-importance data in various scenarios, such as distributed detection [21], distributed estimation [130], spectrum sensing [80], or medical applications [75]. Finally, in the works of Arroyo-Valles *et al.* [9, 10] a Markov Decision Process (MDP) was proposed as a general technique to design censoring algorithms. In an MDP, the environment is modeled as a set of states and there exist one or more agents that can take actions that modify the current state of the environment. An immediate reward is assigned to the agent for each action in each state and the agent objective is to optimize some long-term aggregated reward [15, 96]. This kind of model is an extremely useful optimization tool for systems where sequential decisions under uncertainty have to be taken and, for that reason, we will also use it in this work.

1.2.3 Harvesting devices

No matter what techniques are used, a battery-powered node will eventually deplete its batteries. Since battery substitution is often unfeasible, a more recent approach to this problem is to use energy harvesting nodes. These devices are able to obtain energy from the environment, from sources such as solar, indoor lighting, vibrational, thermal, chemical, electromagnetic, etc. [36, 65, 91, 97, 99, 102, 112]. This approach provides a promising future of self-sustainable networks with virtually perpetual operation lifetimes. However, even when nodes are capable to harvest energy, the availability of ambient energy is usually scarce and stochastic. Hence, energy-efficient strategies are still critical to achieve good network performance. The use of censoring for WSNs composed of harvesting devices seems a useful energy management strategy and has not been sufficiently explored. Hence, in this thesis we will focus on developing censoring algorithms for harvesting WSNs. As previously announced, we use an MDP to model the system and derive our censoring algorithms.

1.2.4 Distributed signal processing

Additionally, in most real-world applications the nodes only perform some data gathering for the subsequent transmission to some more powerful sink node. That approach makes sense as nodes have low computation and memory capabilities and we cannot expect them to perform complex processing tasks. However, as nodes are supposed to be densely deployed in the area of interest, they could benefit from local interactions in order to obtain a number of advantages with respect to non-cooperative strategies:

- **Robustness.** When there is only one sink node, there is a single point of failure; i.e., if the sink node fails, all the network gets disconnected. In-network processing is a way to significantly improve the robustness of the network —the performance of the whole network is not compromised if any node fails.

- **Efficiency.** When the processing is performed in a distributed manner and nodes only communicate with their neighbors, there can be a notable reduction in the number of communications, as nodes do not have to relay all the data to a sink node. This turns into a more efficient use of network energy.
- **Immediacy.** If the nodes in the network are not only sensor nodes but also actors, i.e. they have to respond in some way to the information they are gathering; then it is evident that delaying such response can be troublesome for some applications. In a distributed processing scheme, nodes have a more immediate access to the information they need to take decisions and they need no feedback from a fusion center.
- **Privacy.** In some cases, the information that the sensor nodes gather can be sensitive and there can be some privacy issues in communicating it throughout the network. In a distributed processing scheme, nodes can only share some processed version of the data in a way that their privacy is preserved.

In this direction, there are a number of researchers working in collaborative in-network signal processing where spatial cooperation is exploited. Nevertheless, this approach opens a number of interesting questions: How can we achieve low-complexity adaptive in-network processing? How must the cooperation among the nodes be? Does this communication need to be synchronous? How can we combine information from different nodes? What happens when we have nodes that are not working correctly? Again, this thesis tries to answer some of these questions.

1.3 Objectives and Main contributions

The objective of this thesis is to contribute to the development of *energy-efficient adaptive algorithms for Wireless Sensor Networks.* In that sense, the main objectives of this work are:

1. *To develop low-complexity censoring algorithms for energy harvesting WSNs.* Our approach takes the following considerations into account:
    - **Stochastic, scarce nature of environmental energy.** The algorithms should be able to handle uncertainties in the harvested energy without

prior knowledge of its distributions. In addition, the algorithms should be able to handle non-stationarities.

- **Different quality of information.** The censoring scheme has to take into account the different relevance/importance of the messages in the network to take the decisions. The distribution of those data importances is also *a priori* unknown.
- **Low Complexity.** The proposed strategy has to be able to work in an online manner, in devices of low computational capabilities.

2. *To design distributed and adaptive strategies for estimation in WSNs.* Our desired approach should have these characteristics:

    - **Tracking.** The estimated variables can be time-varying, i.e., our approach must be able to track variations in the estimated variables.
    - **Node diversity.** Networks can be composed of nodes of different nature.
    - **Low Complexity.** Due to the limited capabilities of sensor nodes, the estimation has to be performed by low-complexity algorithms. In addition, in order to keep the energy and bandwidth efficiency, the communication among the nodes must be as scarce as possible.
    - **Asynchrony.** The algorithms must work without an strict synchronization among the nodes. In addition. the estimation scheme should be able to work under node failures or when nodes censor their own estimations.

3. To unify the previous two techniques in a combined scheme, obtaining a censoring strategy for adaptive diffusion networks equipped with energy harvesting devices.

Thus, from the aforementioned objectives, the main contributions are summarized as follows.

In the first part of the thesis, we focus on developing censoring strategies for harvesting WSNs. As our strategy should be able to handle the uncertainties on energy and data distributions, we propose a censoring scheme based in Markov Decision Processes (MDPs) [15]. MDPs have been successfully used in different WSNs problems, where sequential decisions under uncertainty has to be taken, as shown in Chapter 2.

In our approach, we firstly propose a model based on MDPs. Optimal policies according to that model can be derived but they are complex to compute and assume *a priori* knowledge of data and energy processes. Consequently, we firstly analyze them in simple setups. Based on the insights obtained from that analysis, we propose two low-complexity approximate schemes. The first one is a *balanced* strategy where nodes try to level the average consumed and harvested energy. This strategy is computationally cheap but suboptimal when finite batteries are taken into account. The second one is based on stochastic approximation techniques. Simulations show that this strategy is significantly better than other state-of-the-art algorithms, such as model-free approaches based in *Reinforcement Learning* [113], in several relevant scenarios.

In the second part of the thesis, we design distributed and adaptive strategies for estimation in WSNs. Based on the objectives above, we have decided to focus on the framework of adaptive diffusion networks [107]. We propose a novel distributed estimation scheme which decouples the local learning and the fusion of information received from the neighbors. This allows us to easily accommodate different estimators in the network and to work with asynchronous networks. In addition, we theoretically analyze our scheme using two standard techniques for adaptive systems analysis.

Then, based on the theoretical and empirical studies we observe that the optimal way of combining the information of nodes is different at the transient and at the steady-state regimes. Consequently, different learning algorithms for the combination weights are proposed and simulated in a number of stationary estimation and tracking scenarios.

Finally, we propose a combined scheme where a censoring algorithm is added on top of our diffusion scheme. In order to do so, we will propose a way of measure the relevance of communications in diffusion networks. Preliminary simulation results show the potential benefit of integrating censoring schemes in energy-constrained diffusion networks.

1.4 Thesis organization

This thesis dissertation is divided in 5 chapters and 4 appendices. In this chapter we have presented the WSNs, their potentials and limitations, as a background for the thesis work. We have also introduced the main contributions of the thesis. These contributions are organized in the following three chapters and constitute the core of this dissertation:

- Chapter 2 is devoted to develop censoring strategies for harvesting WSNs equipped with finite batteries. After a state-of-the-art review, the modeling, optimization, design and evaluation of censoring strategies is presented.
- Chapter 3 introduces the Decoupled Adapt-then-Combine diffusion scheme and two adaptive rules to learn the combination parameters for that scheme. It also starts with a state-of-the-art revision. Then, the proposed diffusion strategy is theoretically studied and, together with the suggested combination rules, compared with other state-of-the-art techniques.
- Chapter 4 unifies the censoring and diffusion approaches. In order to evaluate the potential of this combined scheme, it is numerically analyzed in different energy harvesting scenarios.

Then, in Chapter 5 the contributions of all the dissertation are summarized and some future research directions are outlined. Finally, the dissertation is closed with four appendices. Appendix A contains some proofs and derivations drawn from Chapter 2, while in Appendix B we have placed the derivations needed for Chapter 3. Later, Appendix C lists the acronyms used throughout the thesis and finally the publications where part of the work of this thesis has been published are listed in Appendix D.

2

MDP models for censoring in harvesting sensor networks

In this chapter, we focus on designing optimal censoring policies for energy-harvesting Wireless Sensor Networks (WSNs). As censoring is a decision-making process under uncertainty we use a Markov Decision Process (MDP) to model it. Later, a *model-based stochastic approximation* algorithm is proposed as a method to solve the previous decision process. We will show the good performance of our strategy with respect to previous works, of which we present a detailed study in the next section.

The rest of the chapter is organized as follows. After the study of previous works in Section 2.1, we present the mathematical model that describes the decision process in Section 2.2. Then, in Section 2.3, we derive the optimal policy for that decision model under some assumptions. The behavior of this optimal policy is analyzed in Section 2.4 in simplified but meaningful scenarios. Later, in Section 2.5, low-complexity approximated schemes are proposed to compute the optimal policy. Finally, the chapter is closed with several numerical experiments to evaluate the proposed schemes.

2.1 State-of-the-art in energy management in harvesting sensor networks

Efficient management of energy resources is essential to operate WSNs equipped with finite-size batteries and energy-harvesting devices [55]. Numerous works have designed energy-saving strategies that account for the limited and stochastic nature of the harvested energy. Many of them were aimed at solving general communication problems such as utility-based cross-layer design [52], power allocation [61, 118], or rate adaptation [66]. At the same time, in the field of WSNs there has been a growing interest in strategies that take into account the importance of the information to be transmitted for the application at hand. The "importance value" can be, for instance, the traffic priority of a routing protocol, the deviation from the mean in distributed estimation [63], the likelihood ratio in decentralized detection [7], or the difference among consecutive estimations in a tracking scenario [92]. Such strategies, sometimes referred to as *selective communication* [9] or *censoring* strategies [7], assume that nodes can evaluate/quantify the importance of the current message and use it to decide whether transmitting or censoring it. To make the decision, additional parameters such as the cost of the communication, the confidence that the message will arrive to its destination, or the *available energy resources*, should also be taken into account.

Since the aforementioned parameters are correlated across time, decisions, which are made sequentially, should be designed to optimize the long-term behavior of the system (for example, by maximizing the aggregated importance of all messages transmitted by the WSN). The current "transmit vs. censor" decision changes the amount of energy stored in the battery and, therefore, has an impact not only on the current battery state, but also on future ones. Therefore, efficient policies have to balance the benefits of an immediate reward with the (expected) impact of each decision on future costs/rewards. From an algorithmic viewpoint, the design of such censoring policies is a Dynamic Programming (DP) problem that, under certain assumptions, can be modeled as an MDP.

An MDP is an optimization model for decision making under uncertainty [15, 96]. The MDP describes a stochastic decision process of an agent interacting with an environment or system. At each decision time, the environment stays in a certain state and the agent chooses an action that is available at this state. After the action is performed, the agent receives an immediate reward and the system transits to a new state according to the transition probability. Consequently actions are chosen not only to maximize the immediate reward but some kind of long-term aggregate reward. For WSNs, MDPs are used to model the interaction between a wireless sensor node (i.e., an agent) and their surrounding environment to achieve some objectives, e.g., optimize a transmission or routing decision in the WSN. In general, the resulting DP problems are difficult to solve and approximate solutions are often required [15, 94].

There are several approaches in the literature to tackle these difficulties in the context of communications with energy harvesting devices. The first one, which has been called *information-theoretic* approach [120], assumes a full knowledge of the environment dynamics. When assuming a "predictable" setting where one knows not only the past, but also the future values of the state information (e.g., energy to be harvested in the future instants), optimal off-line decision policies can be applied [119, 132]. Although such schemes serve as a benchmark or to obtain communication limits, they do not cope well with many practical scenarios, where energy and packet-arrival processes are not known in advance. Consequently, many works in the literature are MDP based, where only some statistics of the processes need to be known. Focusing in decision problems in harvesting WSNs, in [69] a Policy Iteration algorithm [15] was used to estimate the decision policy maximizing the long-term average reward, for a dual recharge/replace battery harvesting model with unitary transmission costs. The scheme in [83] is also based on unitary costs and an average reward optimization. This work and its later extension in [84] show that (a) the optimal transmission policy applies a threshold over the importance values that is a decreasing function of the available energy, and (b) a *balanced policy* is close to optimal. The balanced policy is a simple scheme, also analyzed here, that takes into account the long-term distribution of the energy harvested and ignores the *instantaneous* battery level [84].

The main drawback of these approaches is that they need to know the distributions of the energy and packet arrival processes, which may not be available in practical scenarios. In such cases, nodes have to be able to learn whatever information is needed in real time. One approach to handle this problem, recently proposed in [19], is to apply Q-learning (a widely used stochastic method proposed in the context of *reinforcement learning* [15, 94]) to solve the DP problem. But this approach has its own problems. Since Q-learning is a model-free algorithm that tries to estimate the value function in a non-parametric manner, it can result in a slow convergence and high computational complexity when the size of the state space grows (the problem is even more severe if the state space is continuous).

In this chapter, we propose a *model-based stochastic approximation* algorithm to solve the previous difficulties. More specifically, we show that, under reasonable assumptions on the battery dynamics, the optimal censoring policy is a threshold function on the importance value. An ad-hoc *stochastic approximation* algorithm [134] exploiting this property is developed that, compared with a conventional Q-learning algorithm, is more efficient in terms of computational complexity, memory requirements, and convergence speed.

It is important to remark that MDPs have already been used in similar problems for non-rechargeable WSNs [5]. Some of the possible decisions to take are the transmission/delay of a message in a scheduling scheme [122], the control of transmit power [51, 67] or even moving an actor/sensor node in detection problems to improve the detection [90]. A more relevant approach for our case is the censoring algorithm for non-rechargeable nodes proposed by Arroyo-Valles *et al.* for single-hop communications [9] and for local optimization of multi-hop networks [10]. These works, like [69, 83, 84], assumed that the importance is a value assigned by the application in hand. Some other contributions have recently focused in implementing these techniques in more specific scenarios, such as decentralized estimation [8], decentralized detection [46], or target tracking [92].

Compared to those scenarios with *non-rechargeable* batteries, e.g., [9, 10], the results in the harvesting scenario are substantially different. While, in a non-rechargeable case, a censoring policy discarding messages whose importance is below a *constant threshold* is quasi-optimal, censoring policies based on *energy-dependent thresholds* are significantly more efficient in harvesting sensors. Finally, we have recently proposed a cooperative censoring strategy for non-rechargeable sensor nodes [43]. In that work, an MDP was proposed at a network level, i.e., the actions take the state and the rewards for the whole network into account. Although this method is expected to obtain a better performance in multi-hop networks, it is difficult to derive implementable online algorithms and its generalization to the harvesting problem is not straightforward either.

2.2 System model

In this section, we introduce notation, explain the mathematical model that describes the dynamics of our system under a censoring policy, and formulate the objective to optimize.

Our decision model is defined by four main components: a) a set of state variables, b) a set of possible actions, c) a probabilistic model of the state dynamics (that describes how future states depend on the current state and the actions taken), and d) a reward model (that describes the immediate reward obtained when some action is taken at a given state) [15]. As explained in the previous section, since we are interested in maximizing the long-term reward and current actions have an impact into the future states, our problem will fall into the DP framework. Moreover, because the state dynamics are assumed Markovian, the problem will be modeled as an MDP.

2.2.1 Notation

Throughout this thesis, we use boldface lowercase letters to denote vectors and boldface uppercase letters to represent matrices. The superscript 'T' represents the transpose of a matrix or a vector. In addition, to simplify the arguments, we assume that all the quantities are real. The notation used in this chapter is summarized in Table 2.1.

2.2.2 State vector

In this chapter we model each node as an individual decision agent. Consequently, consider a node that receives a sequence of requests to transmit different messages. The messages can be received from another node or generated from local measurements. The state of the node will be characterized by two real random variables:

- $e(n)$: the battery level at step (slot) n. It reflects the "internal state" of the node.

Table 2.1: Summary of the notation used in Chapter 2

n	Temporal epoch or slot index
$\mathbf{s}(n)$	State vector of the node, takes values in the set $\mathcal{S}$
$e(n)$	Battery level at slot n
$x(n)$	Importance value of the message to be sent at slot n
$a(n)$	Action about sending message at n, $a(n) \in \{0, 1\}$
π	Policy. Sequence of actions performs by the node
$\hat{b}(n)$	Energy consumption at slot n. It depends on $a(n)$
$h(n)$	Energy harvested at slot n
$b(n)$	Remaining energy cost, discounting harvested energy
$r(n)$	Reward given to the node at slot n
$w(n)$	Success index for transmission at slot n
γ	Discount factor for the computation of long-term reward
$V_\pi(\mathbf{s})$	Value of policy π starting at state $\mathbf{s}$
$W(\mathbf{s})$	Success probability of transmission at state $\mathbf{s}$
$\tau(e)$	Decision threshold
$\lambda(e)$	Reduced value function

- $x(n)$: the importance of the message to be sent at step n.

Following the typical terminology in MDP models, the state vector of the node is defined as $\mathbf{s}(n) = [e(n), x(n)]$; i.e., the state vector contains all and only the information that is *available at the node* to make a decision at time n. The set of all possible states is denoted as $\mathcal{S}$.

To facilitate exposition, n is considered an *epoch* or slot index, which starts when the node has to decide whether to censor or transmit a message (either received from one of its neighbors or generated from its sensing devices) and ends when the next message is received. Besides transmitting or censoring the message, during each epoch the node (eventually) collects some energy from the environment. This approach, which is used by many authors, implies that the actual duration of each slot n is stochastic. Clearly, the results in the paper also hold true if the system operates with a constant sampling period (the only modification required is to set

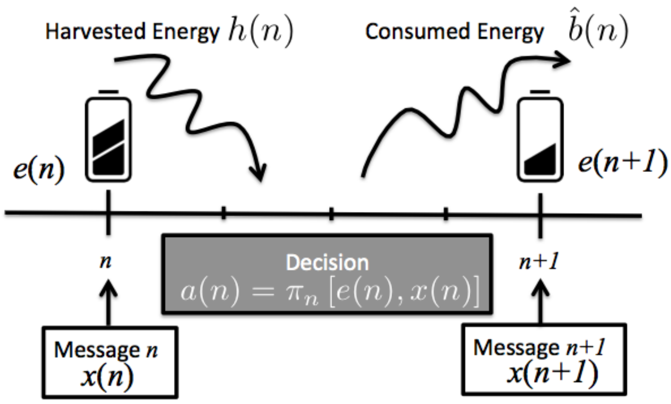


Figure 2.1: Data and energy operation model. At each time epoch n the node receives a message and a decision about censoring or transmitting it has to be made, according to the importance of the message $x(n)$ and the energy state $e(n)$. During that epoch the node may harvest some energy $h(n)$ and eventually it will consume some energy $\hat{b}(n)$ according to its action.

$x(n) = 0$ for the time instants n where no message has been received). In Fig. 2.1 you can find a diagram of this data and energy operations.

Note that, besides $e(n)$ and $x(n)$, the node could use additional information to make decisions. This information can be local (the packet length, the state of the communication channel) or belonging to other (neighboring) nodes. Additional local information can be easily incorporated into the formulation, provided that the state dynamics are similar to those of $e(n)$ and $x(n)$; see, e.g., [10]. Incorporating information about the state or the eventual actions of neighboring nodes (e.g., battery levels, or information about their censoring policy) will lead to a better network operating point but it raises issues such as the accuracy and the cost of acquiring non-local information (exchange of information requires, for example, additional energy consumption). In this work we focus on the design of separate (per-node) censoring policies, so we will work with local information. As we will see in Section 2.3, the success index variable defined in Section 2.2.5 could be used as a mean to couple the decisions across the network —see [10] for details.

2.2.3 Actions and policies

At each time epoch n, the sensor node must take an action (decision) $a(n)$ about sending the current message ($a(n) = 1$), or censoring it ($a(n) = 0$). A forwarding policy $\pi = \{a(1), a(2), \ldots\}$ at a given node is a sequence of decision rules, which are functions of the state vector; i.e.,

$$a(n) = \pi_n [\mathbf{s}(n)] = \pi_n [e(n), x(n)] . \tag{2.1}$$

2.2.4 State dynamics

Next, we describe the model for the stochastic processes $e(n)$ and $x(n)$ that form the state vector. The energy consumed at each time epoch depends on the taken action. Let cost $\hat{b}(n)$ denote the energy consumed by the node and $h(n)$ the amount of energy (if any) harvested by the node since the last action $a(n-1)$. Then, $e(n+1)$ can be written recursively as $e(n+1) = \boldsymbol{\phi}_B \left[e(n) - \hat{b}(n) + h(n) \right]$, where $\boldsymbol{\phi}_B(e) = \max(0, \min(e, B))$ is a clipping (projection) function that guarantees that the energy stored in the battery is never negative, nor exceeds its maximum capacity B[1].

Cost $\hat{b}(n)$ may include the cost of data sensing (if the sensor is the source of the message), the cost of data *reception* (when data come from other nodes), the cost of idle periods, or whatever other costs incurred since the last action. When $a(n) = 1$, $\hat{b}_n$ includes the previous costs plus the cost of *transmitting* the message. This statistical model allows us to deal with a broad range of scenarios: stochastic packet arrivals, communications over fading channels, packet losses, or automatic repeat request (ARQ) schemes, to name a few. In those networks, the energy consumption during node communications can vary depending on the amount of retransmissions required for a successful packet arrival. The range of values and statistical model for $h(n)$ depend on both the type of harvesting device and the source of energy considered [65]. To simplify notation, we define $b(n) = \hat{b}(n) - h(n)$, so that battery dynamics can be rewritten in a more compact form as

$$e(n+1) = \boldsymbol{\phi}_B [e(n) - b(n)] . \tag{2.2}$$

[1]Similar models are used in related works [55, 83]. For example, [83] use a slightly different model, $e(n+1) = \min\{\max\{e(n) - \hat{b}(n), 0\} + h(n), B\}$, which assumes that the energy recharge happens at the end of the decision slot. Using this alternative model does not state special difficulties, and would not change the qualitative analysis in this thesis.

Note that high values of harvested energy can render $b(n)$ negative.

2.2.5 Rewards

The *reward* at time n is given by

$$r(n) = a(n)w(n)x(n), \tag{2.3}$$

where $w(n) \in \{0, 1\}$ denotes the *success index* (a binary variable taking value 1 if the transmission is successful, and zero otherwise). Thus, the reward $r(n)$ that each node receives is a positive value $x(n)$ if and only if it decides to transmit the message ($a(n) = 1$) and the transmission is successful ($w(n) = 1$). Otherwise, the reward is zero.

The meaning of the success index $w(n)$ depends on the application scenario. In general, a reasonable choice is to set $w(n) = 1$ if and only if the message is properly received at its final destination. However, in multi-hop networks, the information about the reception of messages at the sink node may not be available to all nodes along the route. In such a case, other (suboptimal) choices for $w(n)$ are possible. For instance, in [10] it is shown that setting $w(n) = 1$ if the neighboring node forwards the transmitted message can be nearly as effective as using the actual information from the sink. Another (simpler) way to decouple the decisions between nodes is just setting $w(n) = 1$ when the node is able to transmit a message, as proposed in [9]. This choice is optimal in single-hop networks with star topology. In any case, the optimal policy in Section 2.3 will demonstrate that the optimal action depends on $w(n)$ or, to be more precise, on the knowledge of $w(n)$ available at the agent making the decision.

2.2.6 Problem formulation

Our transmission policies will be designed so that the expected aggregate reward is maximized. Following a standard approach in DP, the discount factor $0 < \gamma < 1$ is considered [15] and, based on it, the expected aggregate reward at each node is defined as

$$V_\pi(\mathbf{s}) = \mathbb{E}\left\{\sum_{n=0}^{\infty}\gamma^n r(n)\middle|\mathbf{s}(0) = \mathbf{s}\right\} = \mathbb{E}\left\{\sum_{n=0}^{\infty}\gamma^n a(n)w(n)x(n)\middle|\mathbf{s}(0) = \mathbf{s}\right\}. \tag{2.4}$$

The optimal transmission policy is then

$$\pi^* = \arg\max_{\pi} V_\pi. \tag{2.5}$$

Note that only messages *successfully* transmitted by the nodes are relevant in (2.4). Eq. (2.4) states a DP problem with infinite horizon (because all future time instants are considered) and discounted cost (due to the presence of γ which penalizes future rewards exponentially) [15]. Indeed, $V_\pi(\mathbf{s})$ is typically referred to as either value function or reward-to-go function. Mathematically, the presence of γ eases the existence of a stationary policy that optimizes (2.4); see, e.g., [15].

Other objectives are possible in DP problems. The most popular ones are either assuming a finite horizon and optimize only up to that horizon, or assuming an infinite horizon but maximizing a long-term average reward [15]. This second approach is followed by Michelusi *et al.* ([82, 83, 84, 85]) but we decide to use the discounted cost in (2.4) because it is, in general, better behaved and the derivation of the policies is somewhat less intricate, requiring fewer assumptions (see, e.g., [15]). Second, from a practical point of view, discounted infinite-horizon formulations are able to handle a larger class of uncertainties about the future. In particular the proposed algorithms can work in short-term stationary environments (see again [15]). Additional details will be given in the ensuing section.

2.3 Optimal stationary policy

This section is devoted to design stationary solutions that solve the DP formulated in Section 2.2.6. Since the objective in (2.4) depends on the stochastic processes $x(n)$, $b(n)$, and $w(n)$, assumptions on the stationarity of such processes are required. The relationships among the main variables in the MDP are represented in the graphical model in Fig. 2.2. Arrows in this model encode direct dependency relationships between variables: the action is a function of the state, the success index depends only on the state (and also on $a(n)$, which is a deterministic function of the state), the consumed or harvested energy depends on the taken action, and the energy at the next state depends on the current battery level and the energy consumed or harvested at time n. The model assumptions underlying the graphical model representation in Fig. 2.2 and that will be used in our analysis, are the following:

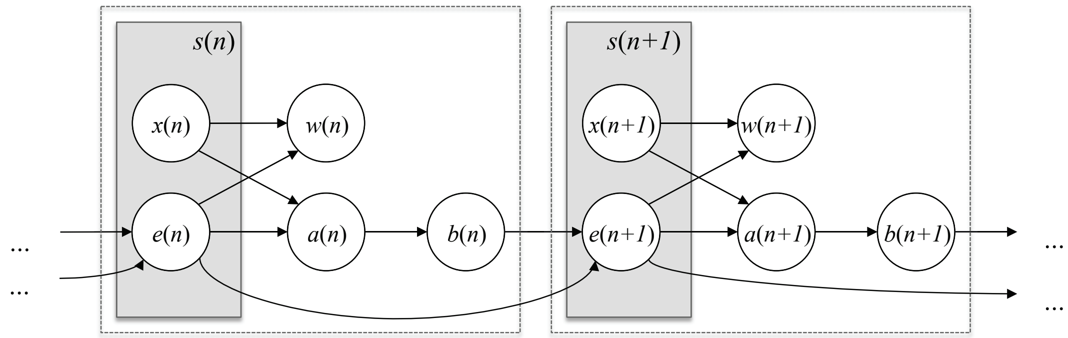


Figure 2.2: Graphical model relating the dependencies among the main variables in the MDP.

A2.1- the process $x(n) \geq 0$ is independent, identically distributed (i.i.d.) and independent of $e(n)$.

A2.2- $b(n)$ is independent of $x(n)$, $e(n)$, and all its previous history, given the action, $a(n)$, and $p(b(n)|a(n))$ (being p the probability density function) does not depend on n.

A2.3- $w(n)$ is independent of all its previous history, given $e(n)$ and $x(n)$, and $p(w(n)|e(n), x(n))$ does not depend on n.

Some independence assumptions may be oversimplifying for some applications: in particular, the independence of the importance values can be non realistic in scenarios where consecutive sensor measurements are correlated. Also, the harvested energy can be time-correlated when it depends on environmental variables that span over several epochs (on the other hand, the energy harvested by wind sensors is oftentimes modeled as i.i.d. [24]). Nonetheless, it is worth mentioning that: i) incorporating time-dependence into our model (while preserving Markovianity) does not state special difficulties, though it would imply some extra computational load and memory requirements, ii) the independence of $b(n)$ with respect to $x(n)$ or $e(n)$ can be relaxed without entailing a big penalty in terms of complexity [10]; iii) due to the presence of the discount factor γ, stationarity can be relaxed to short-term stationarity (more specific comments will be provided in this section after presenting the optimal solution); and iv) the stochastic schemes proposed in Section 2.5 will be able to handle non-stationarities.

Under the previous assumptions and due to the recursive definition of $e(n)$ given by (2.2), the state dynamics are Markovian. Hence, the tuple $(\mathcal{S}, \mathcal{A}, P, r)$, where $\mathcal{S}$ is the set of states, $\mathcal{A} = \{0, 1\}$ is the *finite* set of possible decisions (actions), P is the transition probability measure that can be expressed as $p(\mathbf{s}(n+1)|\mathbf{s}(n), a(n)) = p(e(n+1)|e(n), a(n))p(x(n+1))$, and r is the instantaneous reward function, constitutes an MDP. As a result, the existence of a Markovian and stationary optimal policy π^* is guaranteed [15, 96].

Since Bellman's work in 1952 [13], it is know that the value function associated with the optimal policy has to satisfy the so-called *Bellman's optimality equation* [15]

$$V_{\pi^*}(\mathbf{s}) = \max_{a \in \{0,1\}} \mathbb{E}\left\{ r(n) + \gamma V_{\pi^*}(\mathbf{s}(n+1)) \big| a(n) = a, \mathbf{s}(n) = \mathbf{s} \right\}, \tag{2.6}$$

which can be used to obtain the optimal decision rule. This is accomplished by Theorem 1. All expectations in the following are computed over $x(n)$, $b(n)$ and $w(n)$, unless otherwise stated through the conditional operators.

Theorem 1 U*nder A2.1-A2.3, it holds that* (2.4) *is maximized by a stationary policy* $a(n) = \pi^*(\mathbf{s})$ *satisfying*

$$a(n) = u \left\{ W\left[e(n), x(n)\right] x(n) - \tau\left[e(n)\right] \right\}, \tag{2.7}$$

where u is the Heaviside step function, $W(e, x) = \mathbb{E}\{w(n)|e(n) = e, x(n) = x\}$ is the success probability and the threshold function τ is defined recursively through the pair of coupled equations

$$\tau(e) = \gamma \left(\mathbb{E}\{\lambda\left[\boldsymbol{\phi}_B(e - b(n))\right] \big| a(n) = 0\} - \mathbb{E}\{\lambda\left[\boldsymbol{\phi}_B(e - b(n))\right] \big| a(n) = 1\} \right), \tag{2.8}$$

$$\lambda(e) = \gamma \mathbb{E}\{\lambda\left[\boldsymbol{\phi}_B(e - b(n))\right] \big| a(n) = 0\} + \mathbb{E}\{(W\left[e, x(n)\right] x(n) - \tau(e))^+\}, \tag{2.9}$$

with $(z)^+ = \max\{z, 0\}$, for any z.

The auxiliary function $\lambda(e)$ represents the expected value function for an initial battery $e(0) = e$, i.e.,

$$\lambda(e) = \mathbb{E}\{V_{\pi^*}(\mathbf{s})|e(0) = e\}. \tag{2.10}$$

Proof: See Appendix A.1.

This theorem comes from the direct application of Bellman's equation to a case where some part of the state, x, is uncontrollable, and the "reduced" value function λ only depends on e. A discussion on this kind of problems can be found in [16, Chap. 6.1]. Focusing for now on (2.7), the theorem establishes that censoring decisions are made by comparing the expected instantaneous reward $W[e(n), x(n)]\, x(n)$ with an energy-dependent threshold $\tau[e(n)]$. The threshold quantifies the loss of the future reward associated with the transmission. In other words, the value of $\tau[e(n)]$ is the difference between the future reward if $a(n) = 0$ (and thus transmission energy is preserved) and that if $a(n) = 1$. Clearly, the expected future reward will be higher if the energy stored in the battery is higher (because more transmissions can be afforded), so that $\lambda(\cdot)$ is an increasing function of e and $\tau(\cdot)$ is always positive. This implies that the optimal policy compares the instantaneous and future rewards and acts accordingly. Moreover, the instantaneous reward depends on the (expected) value of the success index, confirming that the optimal action depends not only on $e(n)$ and $x(n)$, but also on $w(n)$.

Due to A2.1-A2.3, the policy in Theorem 1 is stationary. As a result, the optimal censoring policy (mapping from state variables to actions) does not depend on the specific time instant, but only on the value of the state variables. As mentioned earlier, the presence of γ opens the door to deal with non-stationarities as long as the state information is stationary in the short-term. Intuitively, the reason is that $\sum_{n=0}^{\infty} \gamma^n$ can be safely approximated by $\sum_{n=0}^{N} \gamma^n$ with $N < \infty$ (for instance, for $\gamma = 0.95$ and $N = 100$ the error in the approximation is less than 1%). Provided that the processes are stationary during at least an interval of length N, using the results in Theorem 1 will lead to a small error. In such a case, the censoring policy would need to be recomputed every time the *distribution* of the random processes changes.

Although Theorem 1 holds for any cost and importance distributions, it does not provide a clear intuition on how such distributions influence the optimal policies. Additionally, the resulting equations are difficult to solve (even if the involved expectations can be computed). The remaining of the chapter is devoted to handle some of these issues. In Section 2.4, we consider several simplifying assumptions that render the theoretical analysis more tractable, so that we can get further insights on the behavior of the optimal solution. Finally, in Section 2.5 we design low-complexity stochastic approximations to the analytical schemes that can be applied in general scenarios (not only to those in Section 2.4).

2.4 Analysis of the optimal policy

In this section, we will analyze different aspects of the optimal policies. In the first subsection, we will derive recursive expressions to compute these optimal policies. In the second subsection, we will obtain the steady-state energy distributions and assess their impact on the optimal policies. To facilitate the computation of the optimal schemes, we will consider three additional simplifying *assumptions*:

A2.4- process $w(n)$ does not depend on $x(n)$.

A2.5- the energy variables are discretized, so that $e(n)$, $b(n)$, and B take integer values. As a result, the energy space is approximated by a finite space, but the approximation error can be minimized by choosing the energy resolution ε small enough, though at the expense of increasing the memory requirements and the computational complexity. Discretization is a widely used approach to deal with continuous-state DP problems.

A2.6- the success probability can be written as

$$W(e) = P\{b(n) \leq e \big| a(n) = 1\}. \tag{2.11}$$

In words, the transmission is successful if the node has energy enough to transmit the message. This is the case if, for example, the communications are error free. In the presence of path losses, (2.11) also holds if the message is retransmitted until a confirmation is received - the path-loss probability would modify the distribution of the energy cost $b(n)$, but not the formal expression in (2.11). Alternatively, if retransmissions are finite (or zero) path losses can be accommodated by just multiplying the right-hand side of (2.11) by the packet loss probability. This equation is specially suited to single-hop communications. In multi-hop networks, it is suboptimal because it does not consider whether the message is eventually forwarded through the network up to the sink. Nonetheless, the equation decouples variables across nodes, simplifying the derivation of analytical expressions that can be useful even for scenarios where it entails a loss of optimality.

Note that these assumptions are only needed to have a simple scenario where the optimal policy can be computed, and consequently to get some insights on its structure and performance. Once that goal is achieved, they will be relaxed in the following sections.

2.4.1 Optimal policies for particular cases

Under assumptions A2.1 and A2.2, the selective transmitter given by (2.7), (2.8) and (2.9) can be described by the following set of discrete equations

$$a = u\left(x - \frac{\tau(e)}{W(e)}\right) \tag{2.12}$$

$$\tau(e) = \gamma\mathbb{E}\{\lambda(\phi_B(e-b))\big|a=0\} - \gamma\mathbb{E}\{\lambda(\phi_B(e-b))\big|a=1\} \tag{2.13}$$

$$\lambda(e) = \gamma\mathbb{E}\{\lambda(\phi_B(e-b))\big|a=0\} + W(e)\mathbb{E}\left\{\left(x - \frac{\tau(e)}{W(e)}\right)^+\right\} \tag{2.14}$$

Note that, since $b(n)$ and $a(n)$ are stationary, index n has been dropped to simplify notation in (2.12), (2.13), and (2.14).

Even in this simplified scenario, (2.13) and (2.14) cannot be solved analytically, so that neither the reduced value function, λ, nor the transmission threshold, τ, can be found in closed form. However, the considered assumptions reduce the size of the state space and thus, facilitate the implementation of classic DP iterative methods, such as Value Iteration and Policy Iteration [15, 94, 96]. For example, using Value Iteration and with l denoting an iteration index, the optimal schemes can be found iterating the equations

$$\tau_l(e) = \gamma\mathbb{E}\{\lambda_{l-1}(\phi_B(e-b))\big|a=0\} - \gamma\mathbb{E}\{\lambda_{l-1}(\phi_B(e-b))\big|a=1\} \tag{2.15}$$

$$\lambda_l(e) = \gamma\mathbb{E}\{\lambda_{l-1}(\phi_B(e-b))\big|a=0\} + W(e)\mathbb{E}\left\{\left(x - \frac{\tau_l(e)}{W(e)}\right)^+\right\} \tag{2.16}$$

where $\tau_0(e)$ and $\lambda_0(e)$ are arbitrary initial values. This iteration is repeated until convergence ($\|\lambda_l - \lambda_{l-1}\|$ is small enough), which is guaranteed.

To gain some insights, next we numerically solve and analyze the optimal solution for different scenarios with stochastic energy costs. We consider the case when $\mathbb{E}\{b|a = 0\} < 0$, which implies that nodes can discard messages to recharge batteries during operation and, thus, the lifetime can be extended indefinitely. To be more meaningful, we will assume a scenario where each decision epoch can be split into a variable number of fixed-duration time slots, and b is decomposed as

$$b = n_S \cdot b_I + b_R - h + a \cdot \Delta, \tag{2.17}$$

where n_S is the number of time slots since the last decision, b_I is the (stand-by) energy consumed from battery during each time slot, b_R is the cost of receiving or sensing the current message, h is the amount of energy harvested since the last node decision, a is the action, and Δ is the incremental cost of deciding $a = 1$. We assume a lossy channel where retransmission trials are repeated until the message is successfully received at destination. Thus,

$$\Delta = n_T b_T \tag{2.18}$$

where b_T is the cost of each transmission trial and n_T is the number of transmission trials.

We have simulated a scenario where $b_R = 2$, $b_I = 1$ and n_S follows a geometric distribution (2.19) with mean $1/p = 2$.

$$P\{n_s = k\} = (1-p)^k p \tag{2.19}$$

We assume a very poor channel, so that transmission trials fail with probability 0.4. The cost of each transmission trial is set to 4. This configuration tries to simulate WSN configurations where the energy cost of transmitting a message is substantially higher than that of sensing or receiving a message. The amount of harvested energy, h, is also stochastic. We assume that the amount of harvested energy can be decomposed as $h = \sum_{i=1}^{m} h_i$, where h_i are i.i.d. variables accounting for the battery recharged at each time slot i. During each time slot, the probability of a nonzero battery recharge is $p_h = 1/3$, and, when $h_i > 0$, h_i is also geometrically distributed with mean m_h. Three different values of m_h, namely, 5, 10 and 15, have been explored. For these values, the corresponding values of

$$\bar{b}_0 = \mathbb{E}\{b|a = 0\} \tag{2.20}$$

are -0.1, -3.4, and -6.7, respectively. Finally, an i.i.d. exponential importance distribution with unit mean was assumed, and $\gamma = 0.999$.

Fig. 2.3(a) shows the threshold function for each value of $\bar{b}_0$. Note that, except for very small values of e, threshold $\tau(e)$ is a decreasing function of e (for very small values of e, the influence of $W(e)$ in (2.12) is non-negligible and, consequently, it is more difficult to understand the behavior with respect to e). This can be explained as follows: for small values of e, the node increases the threshold to avoid that messages with low importance deplete batteries. For e close to the maximum battery load, there is almost no benefit of refusing transmission, because the battery cannot be indefinitely charged and, thus, only very unimportant messages are censored. Additionally, as $\bar{b}_0$ gets smaller (more negative), τ gets smaller too. The reason is that faster battery recharge allows for a higher transmission rate.

Fig. 2.4 illustrates the effect of changes in the battery size, B, in the threshold function for the second test-case ($m_h = 10$ and $\bar{b}_0 = -3.4$). Here, we also show as a baseline a constant threshold policy $\bar{\tau}$ (horizontal dotted line), whose value is chosen so that the *average* (long-term) energy consumption coincides with the *average* harvested energy, without considering the battery limits. This simple policy, which is related to the "energy neutral operation" concept proposed in [65], is analyzed in different works, either under the name of balanced policy (BP) in [83] or under the name of non-adaptive balanced policy (NABP) in [55, 84]. Using this approach, the problem reduces to estimate the *constant* threshold $\bar{\tau}$ that renders $\mathbb{E}\{b\} = 0$, assuming $B = \infty$. This condition can also be written as

$$\mathbb{E}\{b|a=0\}F_X(\bar{\tau}) + \mathbb{E}\{b|a=1\}\left[1 - F_X(\bar{\tau})\right] = 0 \tag{2.21}$$

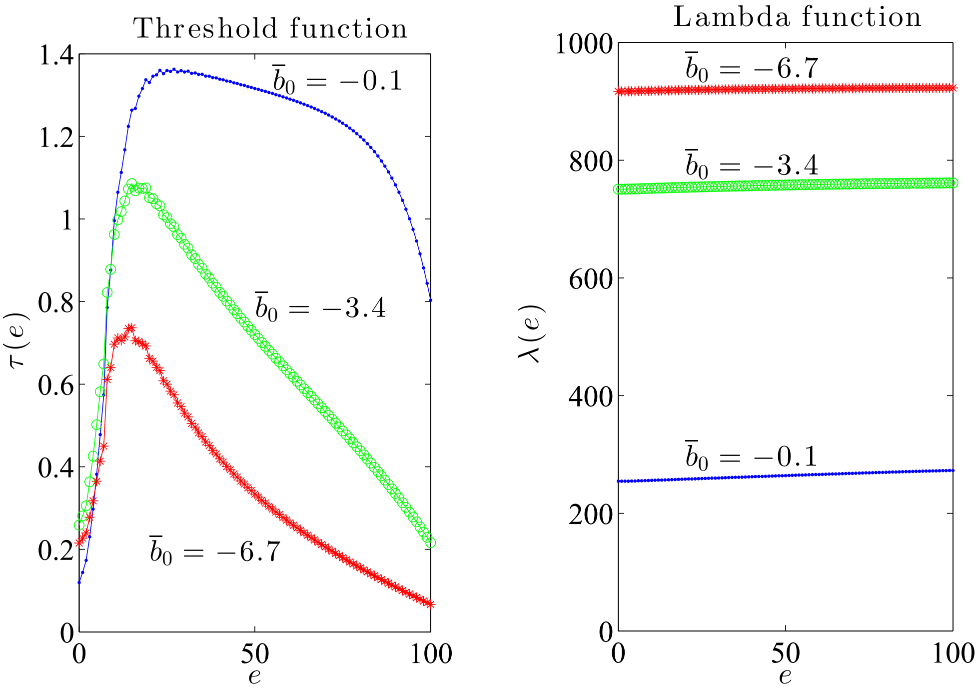
with F_X^{-1} being the cumulative distribution function of x. This equation is solved for

$$\bar{\tau} = F_X^{-1}\left(\frac{\bar{b}_1}{\bar{b}_1 - \bar{b}_0}\right), \tag{2.22}$$

where $\bar{b}_1 = \mathbb{E}\{b|a=1\}$ and F_X^{-1} is the inverse of the cumulative distribution function of x.

2.4.2 Asymptotic behavior: steady-state distributions

In this subsection, we will focus on asymptotic behavior, so that the effects of the initialization are disregarded.



(a) (b)

Figure 2.3: (a) Optimal thresholds for a harvesting node with $B = 100$, stochastic energy costs, a unit-mean exponential importance distribution and $\gamma = 0.999$, for different values of $\bar{b}_0$. (b) The value function $\lambda(e)$ for the same setup.

Using elementary Markov Chain properties [17], it can be shown that, under some general conditions, the statistical distribution of the battery level converges, as n goes to infinity, to a stationary distribution $\boldsymbol{\phi}$ that is the solution of $(\mathbf{I} - \mathbf{P})\boldsymbol{\phi} = \mathbf{0}$ subject to $\phi_i \geq 0$ and $\sum_{i=0}^{B} \phi_i = 1$, where $\mathbf{P}$ is the transition probability matrix with entries

$$p_{ij} = P\{e(n) = j \big| e(n-1) = i\}, \qquad i, j = 0, \ldots, B \tag{2.23}$$

and where, for notational convenience, we started the matrix indexing at 0.

Using the stationary distributions, the expected performance of a selective trans-

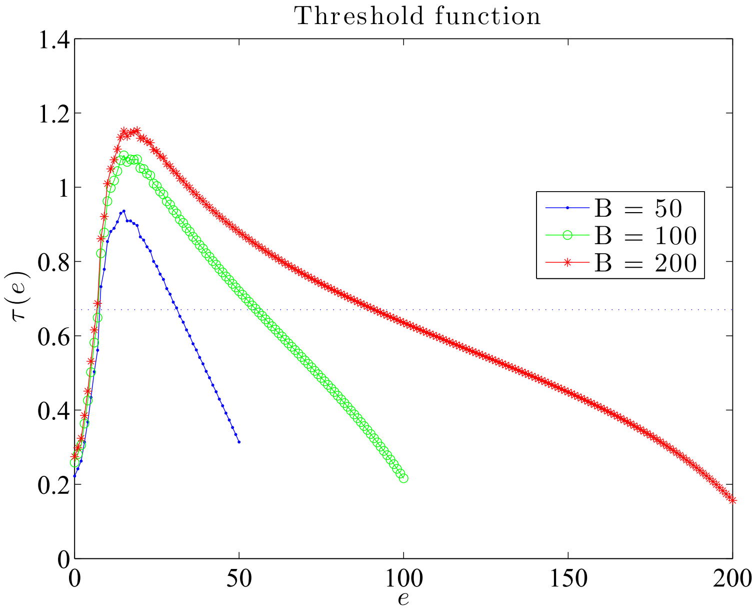


Figure 2.4: Optimal thresholds for a harvesting node with $\bar{b}_0 = -3.4$, $\bar{b}_T = 4$, exponential importance distribution and $\gamma = 0.999$, for different values of the battery size. The horizontal dotted line shows the constant threshold value balancing the average energy consumption with the recharging rate.

mitter can be computed as

$$V^* = \lim_{n \to \infty} \sum_{t=n}^{\infty} \gamma^{t-n} \mathbb{E}\{a(t)w(t)x(t)\}. \tag{2.24}$$

Leveraging A2.3, and using (2.7), equation (2.11) can be substituted into (2.24) to yield,

$$V^* = \frac{1}{1-\gamma}\mathbb{E}\{a(t)w(t)x(t)\} = \frac{1}{1-\gamma}\sum_{e=0}^{B} \mathbb{E}\left\{a(t)w(t)x(t)\middle| e(t) = e\right\}\phi_e \tag{2.25}$$

Taking into account that $w(t)$ does not depend on $x(t)$, $a(t)$ and e, the expectation in the sum can be computed as

$$\begin{aligned}\mathbb{E}\{a(t)w(t)x(t)\big|e(t)=e\} =&\mathbb{E}\left\{w(t)x(t)\big|a(t)=1,e(t)=e\right\}P\{a(t)=1|e(t)=e\}\\ =&\mathbb{E}\left\{w(t)\big|a(t)=1,e(t)=e\right\}\mathbb{E}\left\{x(t)\big|a(t)=1,e(t)=e\right\}\\ &\cdot P\{a(t)=1\big|e(t)=e\}\\ =&W(e)\mathbb{E}\left\{x(t)\big|a(t)=1,e(t)=e\right\}P\{a(t)=1\big|e(t)=e\}\\ =&W(e)\mathbb{E}\left\{x(t)\cdot u\left(x-\frac{\tau(e)}{W(e)}\right)\right\} \end{aligned} \tag{2.26}$$

Defining function g as

$$g(\tau)=\mathbb{E}\{u(x-\tau)x\}, \tag{2.27}$$

and substituting (2.26) into (2.25) we arrive at

$$V^*=\frac{1}{1-\gamma}\sum_{e=0}^{B} g\left(\frac{\tau(e)}{W(e)}\right)W(e)\phi_e \tag{2.28}$$

To compute $\boldsymbol{\phi}$, one has to compute first the transition matrix $\mathbf{P}$, which depends on the censoring policy [cf. (2.23)]. This is accomplished in Appendix A.2.

It is important to remark here that, strictly speaking, optimizing (2.25) is only equivalent to optimizing (2.4) (our objective) when $\gamma \to 1$. Nevertheless, studying (2.25) is useful to evaluate the long-term performance of our strategy.

Additionally, we compute the expected performance of the balanced policy (2.22) based on a constant threshold $\bar{\tau}$. For such constant threshold, $\tau(e)=\bar{\tau}W(e)$, and (2.28) can be simplified as

$$V_{\bar{\tau}}=\frac{g(\bar{\tau})}{1-\gamma}\sum_{e=0}^{B} W(e)\phi_{\bar{\tau},e}. \tag{2.29}$$

Finally, for a non-selective transmitter, $\bar{\tau}=0$, $g(\bar{\tau})=\mathbb{E}\{x\}$ and

$$V_0=\frac{\mathbb{E}\{x\}}{1-\gamma}\sum_{e=0}^{B} W(e)\phi_{0,e}. \tag{2.30}$$

Fig. 2.5 shows the expected discounted reward [cf. (2.24)] for m_h ranging from 1 to 29 (equivalently $\overline{b}_0=[-12,2]$) and for $p_h=1/3$. Fig. 2.6 illustrates a similar behaviour for a scenario with $b_T=b_R=2$ and $p_h=0.04$. In the horizontal axis we show the corresponding value of $\overline{b}_0$ (average energy consumption when there is

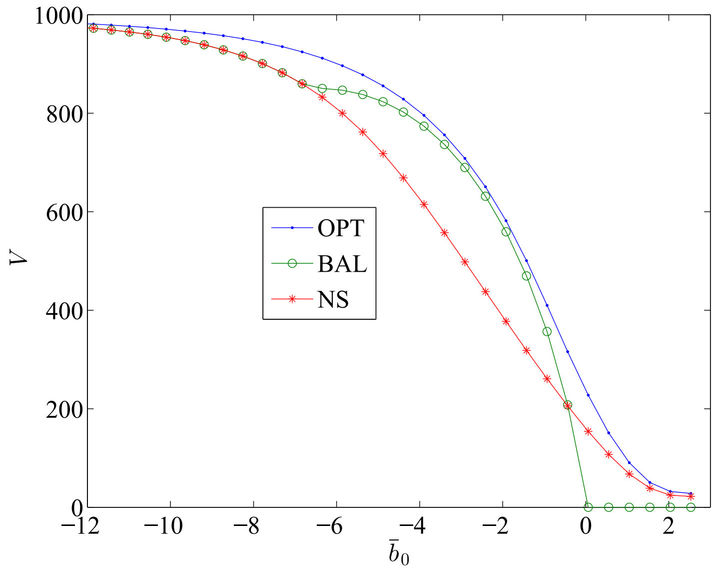


Figure 2.5: Expected performance for a scenario with stochastic energy costs and high refill probability $p_h = 1/3$, as a function of $\bar{b}_0$. Battery size is set to $B = 100$ and exponential importance distribution with unit mean, and $\gamma = 0.999$.

no transmission). The figures demonstrate that: as $\bar{b}_0$ increases, the performance of the balanced transmitter (BAL) deteriorates much faster than the one of the optimal transmitter (OPT), being even worse than that a non-selective strategy (NS). This effect is more noticeable in Fig. 2.6, where the harvesting is more occasional (lower probability of refill). Another relevant behavior is observed on the left region of Fig. 2.5 (very negative values of $\bar{b}_0$). In that region, the energy harvesting is enough to compensate on average the communications costs, so that OPT, BAL and NS obtain the same performance. As energy decreases censoring is needed, but still OPT and BAL perform closely. Fig. 2.6 also shows that, even in situations where the transmissions costs are similar to reception costs, the performance of OPT is noticeably superior to BAL and NS. From both figures, we can conclude that there are some scenarios where the performance of BAL is far from optimal.

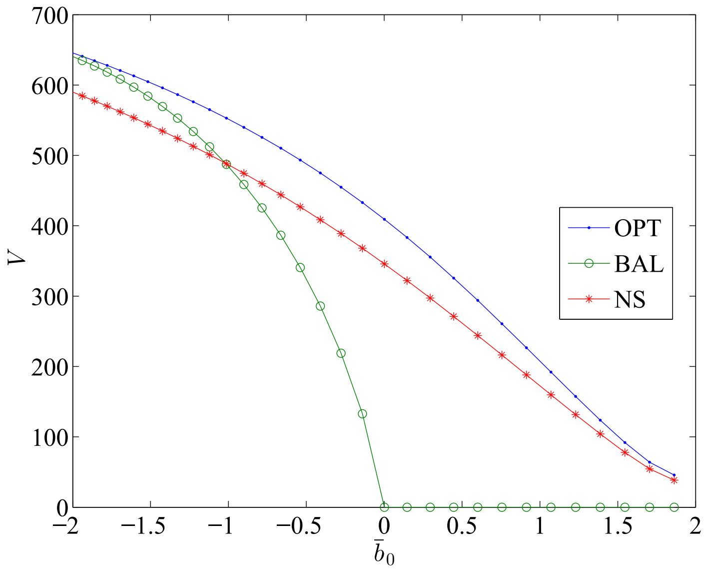


Figure 2.6: Expected performance for a scenario with stochastic energy costs and low refill probability $p_h = 0.04$, as a function of $\bar{b}_0$. Battery size is set to $B = 100$ and exponential importance distribution with unit mean and $\gamma = 0.999$.

2.5 Stochastic approximate schemes

The analysis in the previous sections provided insights on the behavior of optimal censoring policies in harvesting sensor nodes. However, the optimal policies presented so far are computationally very expensive, so that they can not be easily implemented in real time by sensors with limited computational capabilities. In this section, we present different ways to develop suboptimal adaptive stochastic schemes that reduce the computational complexity and, additionally, are able to deal with non-stationarities.

2.5.1 A stochastic approximation to the optimal policy

The *threshold-based* optimal policy presented in Section 2.3 stands on two main assumptions on the energy dynamics: (a) neither the energy consumption nor the recharge depend on the importance value, but only on the taken action, and (b) in the linear regime (i.e., with the exception of the battery saturation points) the variation of the energy stored in the battery does not depend on the current battery level.

The main difficulty to obtain the optimal solution using the value iteration method proposed in (2.15) and (2.16) is the computation of the expectations involved. But if we assume again that $e(n)$ is discrete[2], they can be stochastically approximated in a sample-based manner.

In order to do so, we will represent the policy as an instance of Robbins-Monro algorithm [134] and use stochastic approximation techniques to get our algorithm. First of all, we will decompose cost $b(n)$ as

$$b(n) = b_0(n) + a(n)\Delta(n) \tag{2.31}$$

that is, $b_0(n)$ represents the energy consumption when a message is censored ($a(n) = 0$) and $\Delta(n)$ represents the incremental cost of transmitting a message. In addition, it is useful to write (2.15) and (2.16) in matrix form. Let us define vectors $\boldsymbol{\tau} = (\tau(0), \tau(1), \ldots, \tau(B))^T$, $\boldsymbol{\lambda} = (\lambda(0), \lambda(1), \ldots, \lambda(B))^T$, and $\boldsymbol{\omega} = (W(0), \ldots, W(B))^T$, and the vector of success indexes $\mathbf{w}_b = (u(0-b), u(1-b), \ldots, u(B-b))^T$. We assume in the following that vectors are indexed from 0, in such a way that, for instance, $\lambda_e = \lambda(e)$. Also, we define the transformation $\boldsymbol{\lambda}' = \mathbf{T}_b\boldsymbol{\lambda}$ such that $\lambda'_e = \lambda_{\phi_B(e-b)}$. In Appendix A.3 we derive the following adaptive rules as an instance of Robbins-Monro algorithm [134]

$$\boldsymbol{\omega}_{n+1} = (1 - \eta_n)\boldsymbol{\omega}_n + \eta_n \mathbf{w}_{b_{0,n}+\Delta_n}, \tag{2.32}$$

$$\boldsymbol{\alpha}_{n+1} = (1 - \eta_n)\boldsymbol{\alpha}_n + \eta_n \mathbf{T}_{\mathbf{b}_{0,n}}\boldsymbol{\lambda}_n, \tag{2.33}$$

$$\boldsymbol{\beta}_{n+1} = (1 - \eta_n)\boldsymbol{\beta}_n + \eta_n \mathbf{T}_{\mathbf{b}_{0,n}+\Delta_n}\boldsymbol{\lambda}_n, \tag{2.34}$$

$$\boldsymbol{\lambda}_{n+1} = (1 - \eta_n)\boldsymbol{\lambda}_n + \eta_n \left(\gamma\boldsymbol{\alpha}_n + [x(n)\boldsymbol{\omega}_n - \gamma\left(\boldsymbol{\alpha}_n - \boldsymbol{\beta}_n\right)]^+\right), \tag{2.35}$$

$$\boldsymbol{\tau}_{n+1} = \gamma(\boldsymbol{\alpha}_{n+1} - \boldsymbol{\beta}_{n+1}) \tag{2.36}$$

where we have defined the auxiliary vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$. η_n stands for the learning step size, which can be set either to diminish with time (for instance in stationary scenarios where one wants $\boldsymbol{\tau}(n)$ to converge to a fixed function) or to a small constant (for

[2]Even if not discrete, a common strategy to deal with the estimation of continuous (state) policies is to discretize the input state and use linear interpolation.

adaptation to changes in non-stationary scenarios). The step size must be chosen in order to balance a good convergence speed and a low steady-state error, but always satisfying the Robbins-Monro step-size conditions for convergence [134],

$$\sum_{n=0}^{\infty} \eta_n = \infty, \text{ and } \sum_{n=0}^{\infty} \eta_n^2 < \infty.$$

Note that, at each iteration, the above rules update all components of vector $\boldsymbol{\lambda}$, i.e., the whole estimate of $\lambda(e)$ is updated for all values of e. Thus, the computational load and memory requirements grow linearly with the number of discrete energy values. Neither the importance nor the energy distribution are required to be known to use (2.32), (2.33), and (2.34). Instead, they are run every time a sample of $b_0(n)$ or $\Delta(n)$ is observed. Regarding the observability of $b_0(n)$ and $\Delta(n)$, some remarks are in order:

- When $a(n) = 0$, $\Delta(n)$ is not observed, and $\boldsymbol{\beta}_n$ cannot be updated.
- When $a(n) = 1$, we assume that the sensor can measure the energy level right before taking the necessary actions to transmit the packet. Thus, the node can measure an intermediate energy level $e(n') = \phi_B(e(n) - b_0(n))$, that can be used to estimate $b_0(n)$ and $\Delta(n)$, as follows:
  - If there is no battery underflow or overflow, then $b_0(n) = e(n') - e(n)$ and $\Delta(n) = e(n+1) - e(n')$.
  - If there is battery depletion ($e(n+1) = 0$), $\boldsymbol{\alpha}_n$, $\boldsymbol{\beta}_n$ and $\boldsymbol{\omega}_n$ are not updated.
  - If there is battery overflow ($e(n+1) = B$), we estimate $b_0(n)$ and $\Delta(n)$ as $\tilde{b}_0(n) = e(n) - e(n')$ and $\tilde{\Delta}(n) = e(n+1) - e(n')$.

Note that the cost estimates under battery overflow are biased, and more involved *imputation methods*, see e.g. [53], could be applied to solve this issue, but this is left for future work. In Table 2.2, we summarize the main steps required to run the stochastic approximate policy (SAP) algorithm.

SAP algorithm

INPUTS: Initial battery e_0, γ, and η

Initialize $\boldsymbol{\omega}_0 = \mathbf{0}$, $\boldsymbol{\lambda}_0 = \mathbf{0}$, $\boldsymbol{\alpha}_0 = \mathbf{0}$, $\boldsymbol{\beta}_0 = \mathbf{0}$.

At each time step n, at the sensor node:

1. Sense or receive a message of importance $x(n)$.
2. Harvest energy $h(n)$ and consume $\hat{b}_0(n)$, ($b_0(n) = \hat{b}_0(n) - h(n)$)
   Energy after sensing $e(n') = \phi_B(e(n) - b_0(n))$.
3. Decide about transmitting the message:
   $a(n) = u\left[\omega_{e(n)} \cdot x(n) - \tau_{e(n)}\right]$.
4. Consume additional cost $\Delta(n)$ if $a(n) = 1$,
   $b(n) = b_0(n) + a(n)\Delta(n)$
   $e(n+1) = \phi_B(e(n) - b(n))$.
5. Approximate $b_0(n)$ as $\tilde{b}_0(n) = e(n') - e(n)$
   and $\Delta(n)$ as $\tilde{\Delta}(n) = e(n+1) - e(n^{'})$.
6. Update $\boldsymbol{\lambda}_n$ according to (2.35).
7. If $(e(n+1) > 0)$: update $\boldsymbol{\alpha}_n$ using (2.33) with $\tilde{b}_0(n)$.
   If $(a(n) = 1)$: update $\boldsymbol{\omega_n}$ and $\boldsymbol{\beta}_n$ according to (2.32) and (2.34) using $\tilde{\Delta}(n)$.
9. $\boldsymbol{\tau}_{n+1} = \gamma(\boldsymbol{\alpha}_{n+1} - \boldsymbol{\beta}_{n+1})$.

Table 2.2: Description of stochastic approximate policy (SAP) algorithm.

2.5.2 Q-learning

An alternative design is to use universal stochastic approximation methods that do not require any assumption on the state dynamics, like Q-learning [113, 127]. Q-learning is a *temporal-difference* (TD) reinforcement learning method. These methods combine dynamic programming with Monte Carlo ideas in such a way that they do not need to model the environment dynamics [113]. They can be expected to outperform SAP in scenarios where the above assumptions are too unrealistic. However, there is a price to pay for this flexibility. The SAP algorithm leverages the structure of the optimal decision to reduce the search space and speed up convergence. On the other hand, Q-learning has to compute a Q value for each possible action and state (energy and importance value); as a consequence, the memory requirements may be too high. Furthermore, at each iteration, the Q-learning algorithm only updates the estimate of the value functions *at the current state*, and though convergence can be theoretically guaranteed, it requires to visit all possible states infinitely often [128]. In practice, for large state spaces, convergence to the optimal solution is difficult and learning time is much larger than that of model-based approaches as it will be shown in the numerical experiments in Section 2.6.

Our Q-learning implementation is a minor variation of the algorithm proposed in [19, Eq. (8)] for a similar application. In order to be able to apply it to our setup, we quantized the importance value, which is a real number, into a number of levels (standard Q-learning needs a discrete state space), and apply the algorithm in [19]. This implementation uses an ϵ-greedy action selection method, i.e., the node takes its best action at current state with probability $(1-\epsilon)$ and a random action with probability ϵ. Hence, there are two free parameters the learning rate η, and ϵ the exploration probability in the ϵ-greedy action selection method. The whole algorithm is summarized in Table 2.3.

2.5.3 Adaptive balanced transmitter

A further step to decrease computational complexity is to restrict the attention to suboptimal policies that are easy to compute. A good candidate is the balanced policy presented in (2.22). This policy estimates a constant (*energy independent*) threshold that tries to balance (on average) the harvested and consumed energy under the infinite battery assumption.

The main difficulty of solving (2.22) is that it requires knowledge of the importance distribution. In most cases such a knowledge is not available, or the distribution may not be stationary. In the next lines, we present a adaptive scheme to bypass those problems. An equivalent method for obtaining an adaptive balance policy was presented in the preliminary work in [49].

Upon defining $\rho = \frac{\bar{b}_1}{\bar{b}_1 - \bar{b}_0}$, equation (2.22) states that the constant threshold $\bar{\tau}$ is the ρ-quantile of the distribution function of x, i.e., F_X. Based on the results in [103], which builds functions that are minimized at specified statistics, we can define the following cost function whose expectation attains its minimum at the ρ-quantile of

Q-learning based censoring algorithm

INPUTS: Initial battery $e(0)$, γ, and η

Initialize $Q_1(x,e)=0$, $Q_0(x,e)=0$, for all x and e.

At each time step n, at the sensor node:

1. Sense or receive a message of importance $x(n)$ and quantize it.
2. Harvest energy $h(n)$ and consume $\hat{b}_0(n)$, $(b_0(n)=\hat{b}_0(n)-h(n))$.
3. Decide about transmitting the message using
   ϵ-greedy action selection method.
4. Consume additional cost $\Delta(n)$ if $a(n)=1$,
   $b(n)=b_0(n)+a(n)\Delta(n)$.
   $e(n+1)=\phi_B(e(n)-b(n))$.
5. If $(n>1)$: Update Q function
   If $(a(n-1)=1)$:
   $$\begin{aligned}Q_1(x(n-1),e(n)) &= (1-\eta)Q_1(x(n-1),e(n))\\ &\quad +\eta(r(n-1)+\gamma\max\{Q_1(x(n),e(n+1)),Q_0(x(n),e(n+1)\})\end{aligned}$$
   Else:
   $$\begin{aligned}Q_0(x(n-1),e(n)) &= (1-\eta)Q_0(x(n-1),e(n)).\\ &\quad +\eta(\gamma\max\{Q_1(x(n),e(n+1)),Q_0(x(n),e(n+1)\}).\end{aligned}$$

Table 2.3: Description of censoring algorithm based on Q-learning.

some conditional probability density

$$J(x,\bar{\tau})=\rho(x-\bar{\tau})^{+}+(1-\rho)(\bar{\tau}-x)^{+}. \tag{2.37}$$

To minimize (2.37) we implement a stochastic gradient method

$$\tau(n)=\tau(n-1)+\eta_n\left\{\rho_n u\left[x(n)-\tau(n-1)\right]-(1-\rho_n)u\left[\tau(n-1)-x(n)\right]\right\}, \tag{2.38}$$

where η_n represents, again, a learning step. Both $\overline{b}_1$ and $\overline{b}_0$ have also to be estimated in order to calculate ρ. This can be easily accomplished using the sample mean. In the following, this method will be referred to as *Adaptive Balanced Transmitter* (ABT) and is summarized in Table 2.4.

ABT algorithm

INPUTS: Initial battery $e(0)$ and η

Initialize $\tau(n) = 0$, $n_0 =$, $\bar{b}_0 = 0$, $n_1 = 0$, $\bar{b}_1 = 0$.

At each time step n, at the sensor node:

1. Sense or receive a message of importance $x(n)$.
2. Harvest energy $h(n)$ and consume $\hat{b}_0(n)$, $(b_0(n) = \hat{b}_0(n) - h(n))$
3. Decide about transmitting the message:
   $a(n) = u\,[x(n) - \tau(n-1)]$.
4. Consume additional cost $\Delta(n)$ if $a(n) = 1$,
   - $b(n) = b_0(n) + a(n)\Delta(n)$
   - $e(n+1) = \phi_B(e(n) - b(n))$.
5. Update estimated costs:
   If $(e(n+1) > 0)$:
   $n_0 = n_0 + 1$,
   $\bar{b}_0 = ((n_0 - 1)/n_0)\bar{b}_0 + (1/n_0)b(n)$.
   If $(a(n) = 1)$:
   $n_1 = n_1 + 1$,
   $\bar{b}_1 = ((n_1 - 1)/n_1)\bar{b}_1 + (1/n_1)b(n)$.
6. Compute $\rho_n = \min\{\max\{\frac{\bar{b}_1}{\bar{b}_1 - \bar{b}_0}, 0\}, 1\}$.
7. Update $\tau(n)$ using (2.38).

Table 2.4: Description of Adaptive Balanced Transmitter (ABT) algorithm.

2.6 Simulation results

In this section, we run simulations to compare the performance of the three presented stochastic policies: SAP, described in Table 2.2; Q-learning in Table 2.3; and ABT, given by Table 2.4. In addition, a non-selective scheme (NS) is included as a baseline and, when possible, the theoretical optimal performance (OPT), calculated as in Section 2.4. Three sets of numerical experiments are simulated. The first one analyzes a single-hop network with stationary energy harvesting processes. The second one considers a non-stationary energy harvesting scenario, also with single-hop topology, where the statistics of the energy refill process vary in a periodical way. The third one analyzes the behavior of the developed schemes in multi-hop networks. Although the proposed algorithms are not optimal for that case, the idea is to test their performance and compare them with other existing alternatives.

To run the experiments we assume that: 1) nodes have a way to measure the energy consumed after each decision. 2) Performance is measured in terms of a sample-based estimation ($\hat{V}$) of the steady-state discounted aggregate reward received at the destination, and, when possible, it is compared to the optimal discounted value obtained using (2.28). $\hat{V}$ is calculated as the discounted (γ-weighted) sum of the importance values of the messages received at the sink during the second half of the simulated horizon. 3) In all experiments, the value of the reward discount factor γ is set to 0.999. 4) Message importances follow an exponential distribution (with mean $\overline{x} = 2$), which is a sensible approximation for several practical scenarios, such as monitoring applications, where most messages are of low importance and a small number of them (alarms) have a high importance. For the non-harvesting case, [9, 43] found that these assumptions on the importance distribution were not very critical, and the main conclusions in those papers also apply here. In any case, neither ABT nor SAP nor Q-learning require prior information about the importance/energy distributions.

2.6.1 Single-hop network

Stationary energy refill

The first set of examples is aimed at validating the ABT and SAP schemes. The idea is to show that they are good approximations to the optimal decision rule. In this section, a single-hop (cellular) network is considered. This allows us to assume that if the energy available in the battery is greater than that required to transmit a message, the message is successfully received at the *sink*. For this experiment, the learning rate of both adaptive algorithms is set to $\eta_n = 1/(1 + \delta \cdot n)$, with $\delta < 1$ denoting a small constant. Since all processes are stationary, convergence is then guaranteed. For each of the schemes, the value of δ is selected to maximize the value of $\hat{V}$. For the comparison with Q-learning, we quantized the importance value into 100 different levels, used a learning rate $\eta = 0.2$, and random exploration

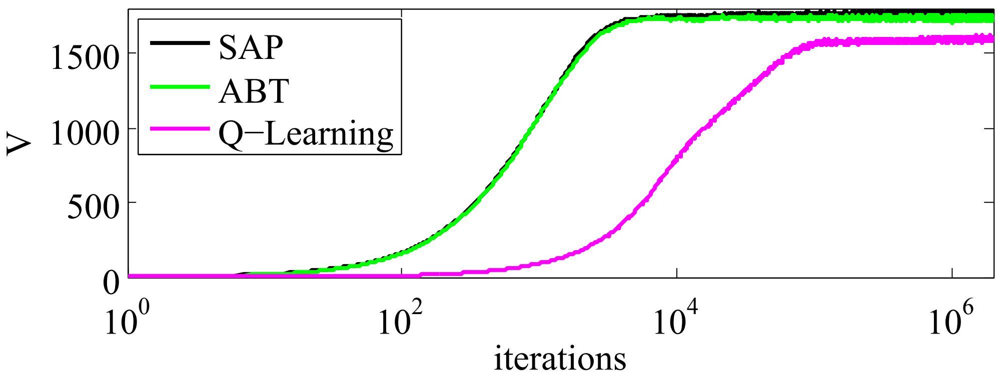


Figure 2.7: Comparison between the SAP, ABT, and Q-learning algorithms.

with probability $\epsilon = 0.1$ [94]. The energy refill values are drawn from a Bernoulli distribution such that the harvested energy at each slot (if non-zero) is $h = 30$ with a fixed probability of harvesting (the value of the probability will vary to simulate different harvesting scenarios). Moreover, we use the cost model in (2.17) and (2.18), with $b_R = 3$, $b_T = 5$, and assuming that the message is retransmitted until it is correctly received (the package error probability is set to 0.3). As already pointed out, by varying the value of the probability of refill, we vary the value of $\bar{b}_0$ and $\bar{b}_1$. The results presented next are obtained after averaging 100 simulations with different random message sequences and energy patterns.

Fig. 2.7 (where the probability of harvesting is 0.3, and therefore $\bar{b}_0 = -6$) confirms the expected behavior: i) the convergence speed of Q-learning (non parametric, non model-based) is much slower than that of SAP and ABT; and ii) within the simulated time horizon (note the logarithmic scale in the horizontal axis), Q-learning does not converge to the optimal solution, mainly because it does not visit some of the states (hence, the corresponding values of the action-value function can never be properly estimated). Due to these limitations, Q-learning will not be included in any further simulations. It is important to remark that the use of more complex reinforcement learning algorithms, e.g., based on function approximation schemes [128], is likely to perform well in this class of problems. We consider this an interesting research direction to be explored in future works.

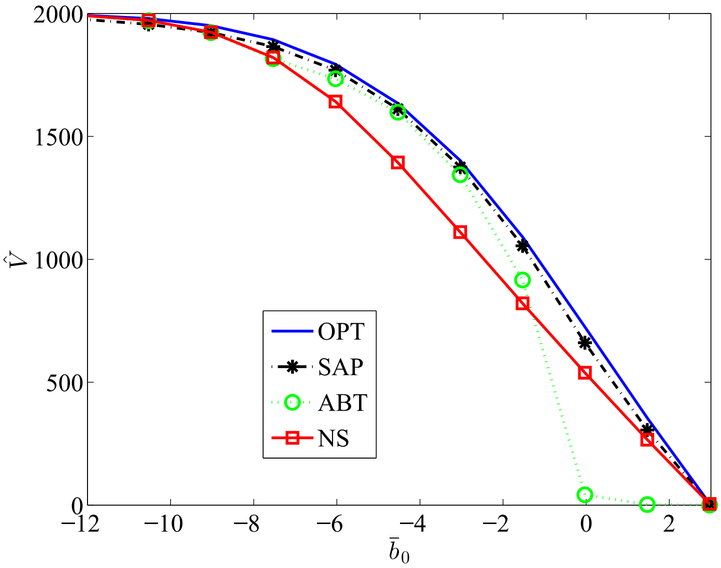


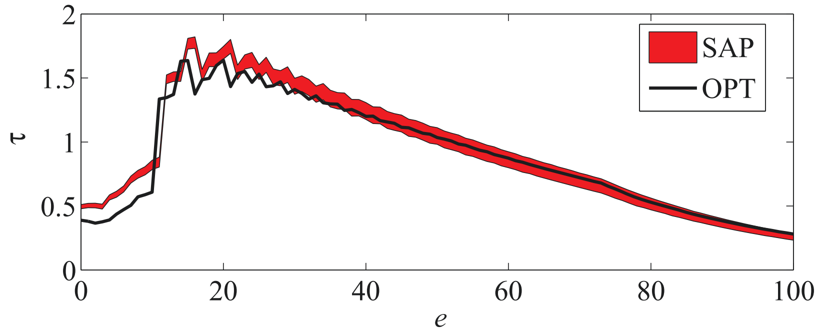
Figure 2.8: Performance in terms of $\hat{V}$ of the proposed algorithm compared to the theory for different $\bar{b}_0$ values in a single-hop network.

In Fig. 2.8, where the probability of harvesting varies from 0.001 to 0.5, we plot the average performance $\hat{V}$ versus $\bar{b}_0$ for SAP, ABT and NS, together with the optimal performance (OPT) calculated using (2.28). It is clear that the SAP scheme outperforms the ABT scheme for all tested cases. More importantly, the SAP scheme achieves a performance very similar to that of the OPT scheme for all $\bar{b}_0$. Fig. 2.9 shows the estimated and optimum thresholds. In general they are close but in case (b), where $\bar{b}_0 = -9$, the energy costs are typically underestimated (due to battery overflows), and consequently the estimation of the threshold is biased. Nevertheless, the performance of SAP is almost optimal (cf. Fig. 2.8).

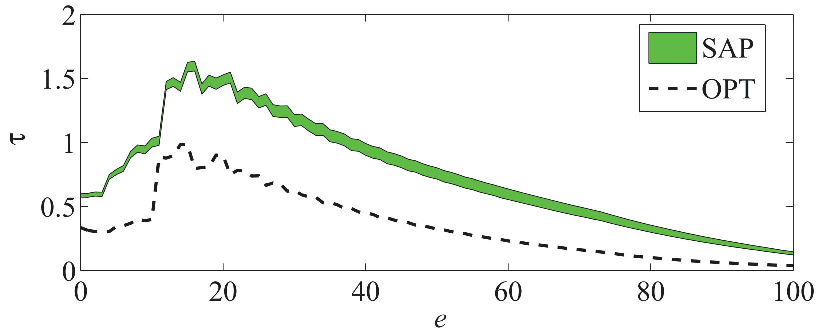
In a nutshell, the simulations validate our approach for the tested scenarios and demonstrate that the stochastic approximation is able to approximate the correct thresholds and, consequently, achieves an almost optimal performance.

Non-stationary energy refill

Table 2.5 shows that, although our derivations assumed that the processes were stationary, our proposed scheme can be applied to non-stationary scenarios. Specifically,



(a) $\bar{b}_0 = -3$



(b) $\bar{b}_0 = -9$

Figure 2.9: Estimation of τ obtained by the SAP (shaded area) and OPT (solid and dashed line) schemes for (a) $\bar{b}_0 = -3$ and (b) $\bar{b}_0 = -9$. For the approximated τ, the average value $\pm 2\sigma$ is shown.

we simulate a periodic refill for which $\bar{b}_0$ is positive during some periods of time (when the harvested energy does not compensate the operating cost) and negative during others. The actual refill in each slot follows a Bernoulli distribution with probability 0.3 and with different h in the two regimes. This way, we simulate a simplified version of a number of harvesting devices that have a periodical behavior with some random component, such as solar energy harvesters.

Since the environment is not stationary, the learning rate is set to a constant value that trades off convergence rate and (gradient) noise in "steady state". In this

	SAP	ABT	NS
$\hat{V}$			
mean	**1188.32**	947.93	685.94
std	3.25	**2.39**	3.23

Table 2.5: Mean and standard deviation of $\hat{V}$ achieved by the SAP, ABT and NS algorithms in a non-stationary environment. Listed values were obtained by running 200 different simulations.

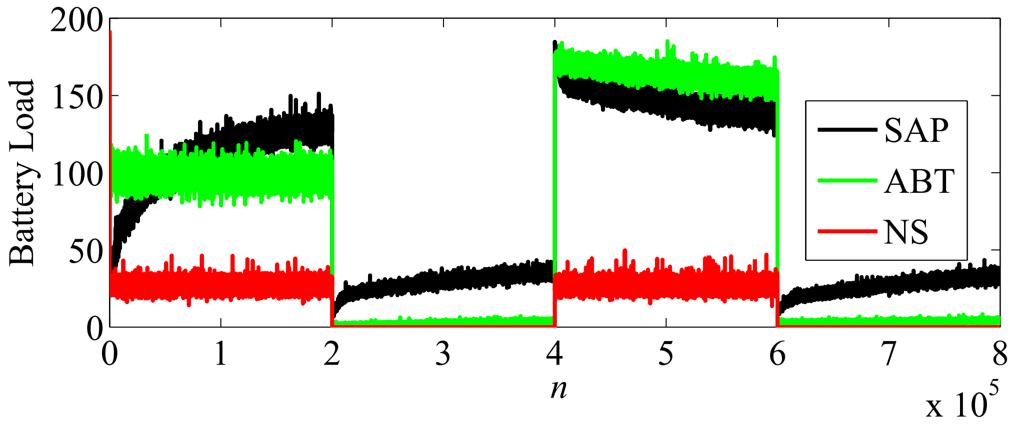


Figure 2.10: Battery evolution (across time) of SAP, ABT and NS algorithms in a non-stationary environment.

experiment, we use $\eta = 0.5$ for SAP and $\eta = 0.05$ for ABT, which are, respectively, the values that empirically maximize $\hat{V}$. The use of an heuristic rule for step-size selection in a real application is an important issue, but it is out of the scope of this thesis.

As already mentioned, the results listed in Table 2.5 show that the benefits of our stochastic approximation schemes also hold true in the tested non-stationary environment. The time evolution of the battery load with time in Fig. 2.10 provides additional insights. It shows that, when no censoring is applied, the battery of the node is empty during long periods of time, leading to a poor performance. On the other hand, the adaptive balanced transmitter (ABT) is able to keep some small amount of energy in the battery during the low-refill periods and, hence, it achieves a bet-

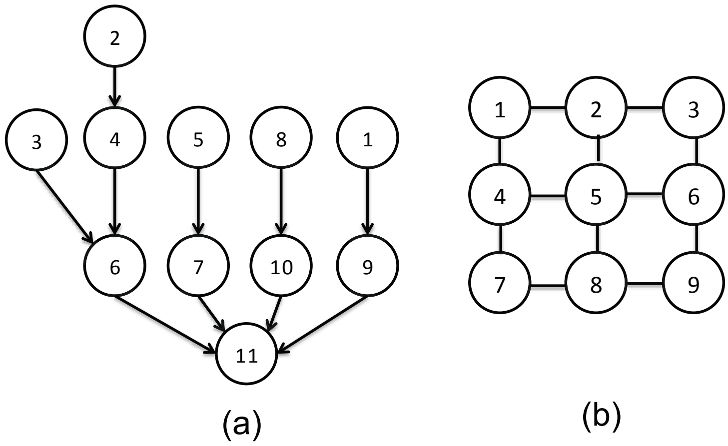


Figure 2.11: Example of network routing topology. In (a) all messages are routed to sink node 11, which is connected to a power line and has no battery limitations. In (b) we choose different nodes as sink nodes.

ter performance. More importantly, the proposed stochastic approximation method is able to modify (adapt) rapidly enough the transmission threshold when the refill pattern changes, so that it always has some battery to transmit important messages and, hence, it obtains the best performance.

2.6.2 Multi-hop networks

In this section, we test our algorithms in a more complex scenario: a multi-hop network with two different topologies. Fig. 2.11(a) represents a random tree topology with a variable number of nodes, where all messages are routed through the tree to the root (sink) node, which is assumed to be wired to a power line. The network tree is randomly constructed so that for each node of index i, we choose with equal probability one and only one node of index j (with $j > i$) to be connected with. The tree graph can be understood either as the actual topology of a cycle-free network or as the spanning tree obtained after running a specific routing protocol to a network with cycles. On the other hand, Fig. 2.11(b) represents a squared-lattice network with fixed number of nodes. In this case, the routing to the sink is fixed and computed using the Dijkstra algorithm [34]. Although the simulated networks are relatively simple, they are useful to illustrate the interaction between the censoring strategies across different nodes.

In addition, we consider a success index $w(n) = 1$ when the node is able to transmit the message, consequently the scheme is suboptimal relative to the maximization of the discounted aggregated reward of the messages received at the sink. It is also assumed that all nodes generate messages with the same probability, so that nodes closer to the sink will handle more traffic. As in the previous subsection, the importance of the messages is an exponential i.i.d. process with mean $\overline{x} = 2$.

We simulated 6 different scenarios, all of them with fixed b_R and b_T, and without channel losses. The harvesting is random and different harvesting probabilities are considered. Scenarios 1, 2, and 3 correspond to a tree network [Fig. 2.11(a)] with 20 nodes (scenarios 1 and 2) and 10 nodes (scenario 3). The harvesting probability is 0.1 for scenario 1, and 0.5 for 2, 3. Scenarios 4, 5, 6 correspond to a grid network of 9 nodes (8 sensors and a sink). In scenarios 4 and 5, the sink is located at one of the corners, e.g., node 9 in Fig. 2.11(b); while for scenario 6 it is at the center of the grid (node 5 in Fig. 2.11(b)). The harvesting probability is 0.5 for scenario 4 and 0.9 for scenarios 5 and 6. Fig. 2.12 shows the average of 200 simulations.

The results point out that, although SAP algorithm was not explicitly designed for multi-hop networks, it achieves a better performance than that of the tested alternatives. In fact, the gain of the SAP scheme relative to the ABT scheme in Fig. 2.12 in scenarios 2, 4, and 5 is much larger than that observed in the experiments presented in the previous section. Hence, the combination of the local optimization processes at each node has a positive global influence. Although the results are not comprehensive, they serve as a preliminary validation. Designing decisions jointly across nodes, accounting for the costs of exchanging information, or investigating the effect of interference and medium access control are all aspects worth analyzing (for example, [85] showed that balanced policies are suboptimal if interference is present), but they are out of the scope of this thesis and are left as future work.

2.7 Summary

In this chapter, we have designed and analyzed a censoring scheme for WSNs with harvesting devices. The problem has been modeled using the MDP framework and

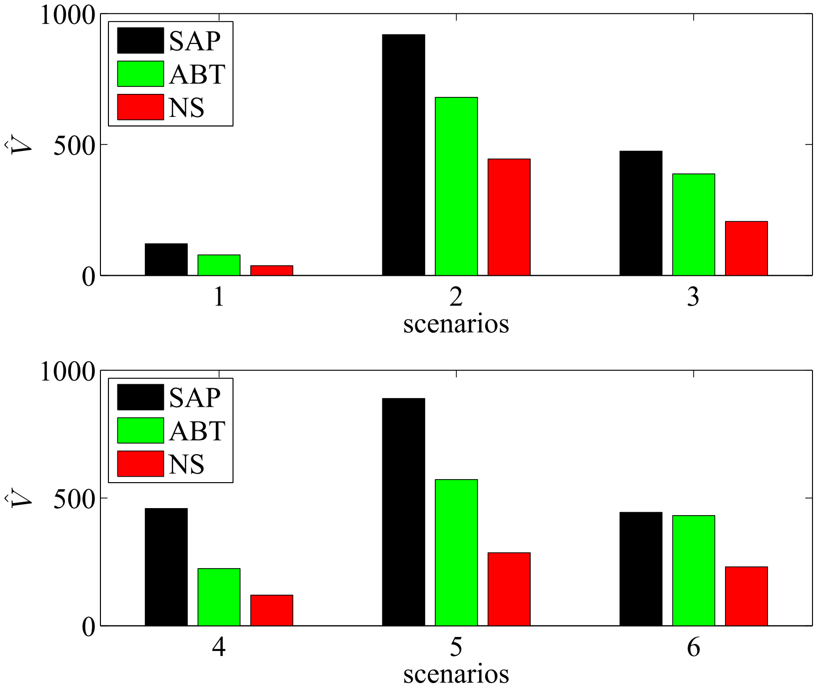


Figure 2.12: Performance in terms of $\hat{V}$ of the two proposed algorithms and the non selective method for 6 different multi-hop scenarios.

some assumptions were made in order to obtain a threshold-based optimal policy. Some insights about these optimal policies were provided and suboptimal schemes based on stochastic approximation were developed too. Numerical experiments confirmed the theoretical claims and showed the benefits of our approach with respect to previous works, especially when the harvested energy is scarce. Finally, experiments showed that these schemes perform well even if some of the assumptions under which they were designed (i.e., stationarity, single-hop networks) do not hold.

A more detailed discussion of the results and some future research lines can be found in Chapter 5.

3

Decoupled diffusion for adaptive distributed estimation in WSN

In this chapter, we propose a novel diffusion strategy for adaptive distributed estimation in sensor networks called Decoupled Adapt-then-Combine (D-ATC). Our strategy, which is specially convenient for heterogeneous networks, is compared with standard diffusion schemes. Such comparison shows the need of implementing adaptive combination rules to obtain a good performance in case of heterogeneous networks for both strategies. Therefore, we propose two adaptive rules to learn the combination coefficients that are useful for our diffusion strategy. Two different theoretical analyses and several experiments simulating both stationary and tracking estimation problems show that our method outperforms state-of-the-art techniques, becoming a competitive approach in different scenarios.

The rest of the chapter is organized as follows. We start presenting a survey of related work in the field of adaptation and learning in networks. Then, the D-ATC diffusion strategy is presented in Section 3.2 and we theoretically analyze its performance in Section 3.3. The learning rules for adapting network combiners are derived in Section 3.4. Finally, we close the chapter with experimental results.

3.1 State-of-the-art in adaptive networks

Over the last years, adaptive diffusion networks have become an attractive and robust approach to estimate a set of parameters of interest in a distributed manner (see, e.g., [29, 44, 74, 107, 108, 109] and their references). Compared to other distributed schemes, such as incremental [73] and consensus [109] strategies, diffusion techniques present some advantages, e.g., they are more robust to link failures or they do not require the definition of a cyclic path that runs across the nodes as in incremental solutions [28]. Furthermore, they perform better than consensus techniques in terms of stability, convergence rate, and tracking ability [108]. For these reasons, adaptive diffusion networks are considered an efficient solution in several applications, such as target localization and tracking [107], environment monitoring [108], and spectrum sensing in mobile networks [33], among others. Moreover, they are also suited to model some complex behaviors exhibited by biological or socioeconomic networks [108].

Diffusion WSNs consist of a collection of connected sensor nodes, linked according to a certain topology, that cooperate with each other through local interactions to solve a distributed inference or optimization problem in real time. Each node is able to extract information from its local measurements and combine it with the information received from its neighbors [107, 108]. This is typically performed in two stages: adaptation and combination. The order in which these stages are performed leads to two possible strategies: Adapt-then-Combine (ATC) and Combine-then-Adapt (CTA) [27, 74]. In both cases, the adaptation and combination steps are interleaved with the communication of the intermediate estimates among neighbors. It is also important to mention that in these diffusion schemes the adaptation and combination steps are mutually dependent, i.e., the combined estimate is fed back into the adaptation phase.

A key aspect in diffusion networks is the way nodes fuse neighbors information. Indeed, the combination weights play an essential role in the overall performance of the network. For instance, diffusion least-mean-squares (LMS) strategies can perform similarly to classical centralized solutions when the weights used to combine the neighbors estimates are optimally adjusted [107, 114, 140]. Initially, different static combination rules were proposed such as Uniform [20], Laplacian [131], Metropolis [131], and Relative Degree [26]. As neither of these rules take into account that the nodes may be operating under different signal-to-noise ratio (SNR) conditions, they result in suboptimal performance when the SNR varies across the network. For this reason, some adaptive schemes for adjusting the combination weights (e.g., [114, 116, 135, 140]) have also been proposed in order to optimize the network performance under these circumstances. Although these adaptive rules can reduce the steady-state error with respect to a scheme with static combiners, some experiments show a deterioration in the convergence behavior [135]. Consequently, some schemes propose the use of different rules for transient and steady-state regimes, and include mechanisms to switch from one rule to the other in an online manner [41, 135]. Particularly, in [41] we proposed a method to estimate the convergence time of the network and, consequently, select different combination rules for transient and steady state.

Most of these works consider just the case of networks composed of homogeneous nodes, i.e., nodes that implement the same adaptive rules with the same parameters (filter length, step size, etc.). However, there are circumstances where the use of heterogeneous nodes may be advantageous, e.g., recurring to nodes with different adaptation speeds to improve the network tracking capability. In this case, the previous approaches for adjusting the combiners fail, and there are presently no alternatives for dealing with that problem in a general case.

For that reason, in this thesis we focus on an alternative diffusion scheme that overcomes this important drawback. In our approach, which will be called Decoupled ATC (D-ATC), the adaptation phase is kept decoupled from the combination phase, i.e., the local estimation of each node is combined with the estimates received from its neighbors, as in standard ATC, but the resulting combined estimation is not fed back into the next adaptation step. This scheme presents a more clear separation between the adaptation and combination phases. As it will be shown later, this allows us to implement two different rules for the combination phase specially suited for heterogeneous networks. With these rules we can obtain a significant improvement in convergence and steady-state performance with respect to previous approaches, both in tracking and stationary scenarios. In addition, our proposal seems to be a more natural scheme for asynchronous environments, which are receiving increasing attention [139].

3.2 Decoupled Adapt-then-Combine diffusion scheme

3.2.1 Notation

As in the previous chapter, we use boldface lowercase letters to denote vectors and boldface uppercase letters to represent matrices. The superscript 'T' represents the transpose of a matrix or a vector. Depending on the context, $\mathbf{0}_N$ represents an $N \times N$ matrix or a length-N column vector with all elements equal to zero, and $\mathbb{1}_N$ is an all-ones column vector with length N. The notation used in this chapter is summarized in Table 3.1.

3.2.2 Description of the diffusion strategy

Consider a collection of N sensor nodes connected according to a certain topology, as depicted in Fig. 3.1. Each node k shares information with its neighbors and we denote this neighborhood of k, excluding the node itself, as $\bar{\mathcal{N}}_k$, while $\mathcal{N}_k = \bar{\mathcal{N}}_k \cup \{k\}$. The network objective at every time instant n is to obtain, in a distributed manner, a vector $\mathbf{w}(n)$ minimizing certain global cost function $J\left[\mathbf{w}(n)\right]$. In particular, in this work we consider the following setting: At every time instant n, each node k has access to a scalar measurement $d_k(n)$ and a regression column vector $\mathbf{u}_k(n)$ of length M, both realizations of zero-mean random processes. We assume that these measurements are related via some unknown column vector $\mathbf{w}_\text{o}(n)$ of length M through a linear model

$$d_k(n) = \mathbf{u}_k^T(n)\mathbf{w}_\text{o}(n) + v_k(n), \tag{3.1}$$

where $v_k(n)$ denotes measurement noise and is assumed to be a realization of a zero-mean white random process with power $\sigma_{v,k}^2$ and independent of all other variables across the network. The objective of the network is to estimate the (possibly) time-varying parameter vector $\mathbf{w}_\text{o}(n)$.

In standard ATC and CTA diffusion strategies, an adaptation and combination phases are iterated to solve the estimation problem in an adaptive and distributed

Table 3.1: Summary of the notation used in Chapter 3

N	Number of nodes in the network
$\mathcal{N}_k$	Neighborhood of node k, including itself
N_k	Cardinality of $\mathcal{N}_k$
$\bar{\mathcal{N}}_k$	Neighborhood of node k, excluding itself
$\bar{N}_k$	Cardinality of $\bar{\mathcal{N}}_k$
$\bar{\mathbf{b}}_k$	Vector with the indexes of all nodes belonging to $\bar{\mathcal{N}}_k$
$\bar{b}_k^{(m)}$	Index of the m^{th} node connected to node k
$\mathbf{w}_{\text{o}}(n)$	Unknown time-varying parameter vector
$\boldsymbol{\psi}_k(n)$	Local estimate of $\mathbf{w}_{\text{o}}(n)$ (based only on local data at node k)
$\mathbf{w}_k(n)$	Combined estimate of $\mathbf{w}_{\text{o}}(n)$ at node k
$\{d_k(n), \mathbf{u}_k(n)\}$	Local desired value and regression vector at node k
$v_k(n)$	Local noise at node k
$y_k(n)$	Local output of node k
$\xi_k(n)$	Local error of node k
$c_{\ell k}(n)$	Combination weight assigned by node k to the estimate received from node $\ell \in \mathcal{N}_k$
$\mathbf{c}_k(n)$	Vector with all weights assigned by node k to the estimates from its neighbors
$\bar{\mathbf{c}}_k(n)$	Vector with the same entries of $\mathbf{c}_k(n)$, excluding $c_{kk}(n)$

manner. In particular the ATC scheme has the following two steps

$$\boldsymbol{\phi}_k(n) = f\left[\mathbf{w}_k(n-1), \mathbf{u}_k(n), d_k(n), \boldsymbol{\theta}_k(n)\right], \tag{3.2}$$

$$\mathbf{w}_k(n) = \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{\phi}_\ell(n). \tag{3.3}$$

where an intermediate estimation $\boldsymbol{\phi}_k(n)$ is calculated as a function of these elements: the previous estimation $\mathbf{w}_k(n-1)$, current local data $\{d_k(n), \mathbf{u}_k(n)\}$, and a state vector $\boldsymbol{\theta}_k(n)$ that incorporates any other information that is needed for filter adaptation. Some typical choices for (3.2) are least-mean-squares (LMS), normalized least-mean-squares (NLMS), Affine Projections Algorithm (APA) [106], etc. This estimation is then shared with the neighbors and combined using the coefficients $c_{\ell k}$, $\ell \in \mathcal{N}_k$.

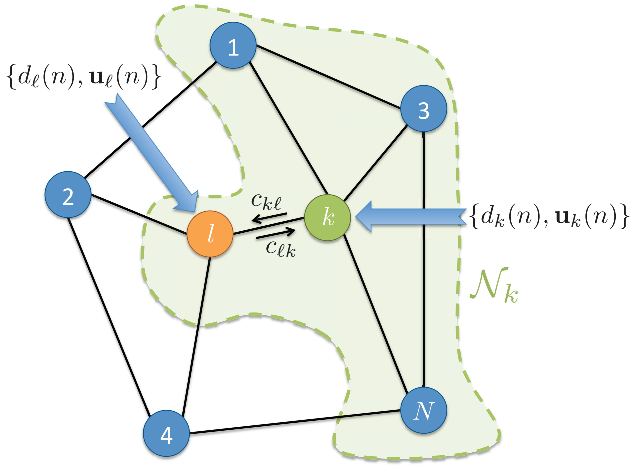


Figure 3.1: Example of diffusion network. At every time step n, each node k takes a measurement $\{d_k(n), \mathbf{u}_k(n)\}$. In this example, the neighborhood of node k is $\mathcal{N}_k = \{1, 3, \ell, k, N\}$ and its cardinality is $N_k = 5$.

The diffusion scheme that we propose here, also iterates an adaptation and combination phase. However, differently from standard ATC or CTA diffusion schemes [107, 108], each node in our scheme preserves a purely local estimation $\boldsymbol{\psi}_k(n)$, which is then combined with the combined estimates, $\mathbf{w}_\ell(n-1)$, received from the neighboring nodes $\ell \in \bar{\mathcal{N}}_k$ at the previous iteration. Note that, although we have selected an ATC approach as the basis of our algorithm, our scheme could be straightforwardly extended to CTA. Consequently, the proposed diffusion scheme can be written as follows

$$\boldsymbol{\psi}_k(n) = f\left[\boldsymbol{\psi}_k(n-1), \mathbf{u}_k(n), d_k(n), \boldsymbol{\theta}_k(n)\right], \tag{3.4}$$

$$\mathbf{w}_k(n) = c_{kk}(n)\boldsymbol{\psi}_k(n) + \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k}(n)\mathbf{w}_\ell(n-1). \tag{3.5}$$

In the adaptation phase (3.4), an updated local estimation $\boldsymbol{\psi}_k(n)$ is calculated as a function of: $\boldsymbol{\psi}_k(n-1)$, $\{d_k(n), \mathbf{u}_k(n)\}$, $\boldsymbol{\theta}_k(n)$. As most adaptive filtering schemes converge to unbiased estimations of the optimal solution in stationary scenarios, i.e., $\mathbb{E}\{\boldsymbol{\psi}_k(n) - \mathbf{w}_\mathrm{o}\} \to 0$ as $n \to \infty$, we constrain all coefficients at each node to sum up to one, so that the combined estimates from (3.5) satisfy the same property. In addition

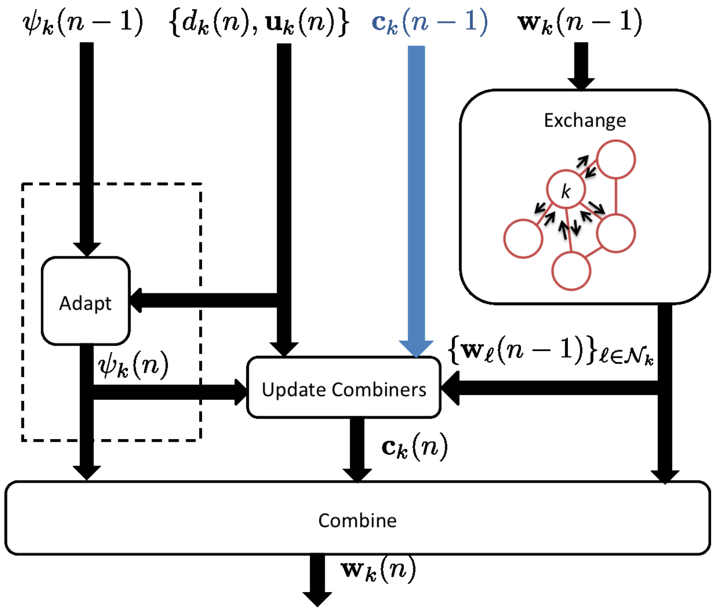


Figure 3.2: Block diagram of Decoupled ATC diffusion stategy, where the decoupled adaptation phase has been highlighted by means of a dashed rectangle. In addition, we have included a blue arrow showing a feedback of the previous combination weights, $\mathbf{c}_k(n-1)$, into the block "update combiners" for such algorithms that compute them recursively (as in Subsection 3.4.1).

to this, for reasons that we explain in Subsection 3.3.3, we also impose non-negativity constraints on such combiners,

$$c_{\ell k}(n) \geq 0, \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(n) = 1, \forall k. \tag{3.6}$$

These conditions on combination coefficients have also been considered in other diffusion schemes available in the literature to guarantee certain stability properties.

In Fig. 3.2 we can see a schematic representation of the proposed diffusion strategy, where the dashed rectangle highlights the decoupled adaptation phase. There are some potential advantages of decoupling the adaptation step from the combination phase:

- The inclusion of any adaptation algorithm to update local estimates is straightforward, as it is to consider heterogeneous networks with nodes that implement different adaptive filters. On the contrary, in standard ATC it is not so clear how to accommodate networks composed of different estimators, because of the feedback of the combined estimate in the adaptation step.

- Related to this, the adaptation phase of our scheme is not influenced by an erroneous selection of the combination weights. In contrast, in standard ATC, if the adaptation of the combination weights is suboptimal, the adaptation phase of the diffusion algorithm can also get negatively affected.

- Since the adaptation of each node is completely independent of other nodes' adaptation, we can more easily deal with synchronization problems or with nodes that receive observations at different rates. Furthermore, the combination stage can easily be modified to include the last available estimates received from the neighbors so that a delay in a particular node does not slow down the network.

In the next subsection, we analyze in more detail the characteristics of the proposed diffusion strategy and compare it to the ATC approach for a simple network with just 2 nodes. As we will see, both diffusion strategies can achieve similar performance, during convergence and in steady state, provided that the combination weights have been adequately chosen (for both schemes).

3.2.3 Comparison between ATC and D-ATC

Before presenting the full theoretical analysis of the new diffusion method, we have carried out a toy experiment considering a small heterogeneous network composed of two interconnected nodes differing just in their adaptation rate. We compare the ATC scheme [107] and the proposed D-ATC scheme (Equations (3.4) and (3.5)) for this setup, considering fixed combination weights for both diffusion schemes. Here, we assume that an NLMS filter is used as the adaptation algorithm for both nodes. As a consequence, the general Equation (3.2) becomes

$$\boldsymbol{\phi}_k(n) = \mathbf{w}_k(n-1) + \tilde{\mu}_k(n)\mathbf{u}_k(n)\left[d_k(n) - \mathbf{u}_k^T(n)\mathbf{w}_k(n-1)\right], \tag{3.7}$$

where $\tilde{\mu}_k(n) = \mu_k / \left[\delta + \|\mathbf{u}_k(n)\|^2\right]$, μ_k is a step size, δ is a regularization factor to prevent division by zero, and (3.4) becomes

$$\boldsymbol{\psi}_k(n) = \boldsymbol{\psi}_k(n-1) + \tilde{\mu}_k(n)\mathbf{u}_k(n)\xi_k(n), \tag{3.8}$$

where the local estimation error signals are

$$\xi_k(n) = d_k(n) - \mathbf{u}_k^T(n)\boldsymbol{\psi}_k(n-1) \triangleq d_k(n) - y_k(n), \tag{3.9}$$

with $y_k(n)$ being the local output. For this toy experiment, we select step sizes $\mu_1 = 0.1$ and $\mu_2 = 1$, for the first and second nodes, respectively.

In Figs. 3.3 and 3.4, we evaluate the steady-state and transitory performance of ATC and D-ATC in terms of steady-state Network Mean-Square Deviation (NMSD) and convergence rate respectively. The NMSD is defined as

$$\text{NMSD}(n) = \frac{1}{N}\sum_{k=1}^{N} \text{MSD}_k(n) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left\{[\mathbf{w}_{\text{o}}(n) - \mathbf{w}_k(n)]^2\right\}, \tag{3.10}$$

where $\text{MSD}_k(n)$ is the mean-square deviation of each node k in the network at iteration n. The steady-state NMSD is defined as $\text{NMSD}(\infty) = \lim_{n\to\infty} \text{NMSD}(n)$. The convergence rate is calculated as the average slope of the NMSD learning curve computed at its initial region (between iterations 200 and 250). There are some interesting conclusions that can be extracted from this simple experiment:

- There is a pair of optimal combination weights in terms of convergence rate and steady-state performance. In homogeneous networks, i.e., networks with nodes that implement a common adaptation algorithm with the same set of parameters, it has been proven that any fixed combination weights in the standard ATC algorithm provides the same convergence rate [41, 135]. However this is not true for heterogeneous networks, as can be seen in the left panel of Fig. 3.4.
- The optimal combination weights for convergence and steady state are different for both schemes.
- Both schemes can reach the same optimal performance if suitable combination weights are chosen.

Consequently, one can use the most suitable diffusion scheme for the application in hand, provided that we are able to select appropriate combination weights.

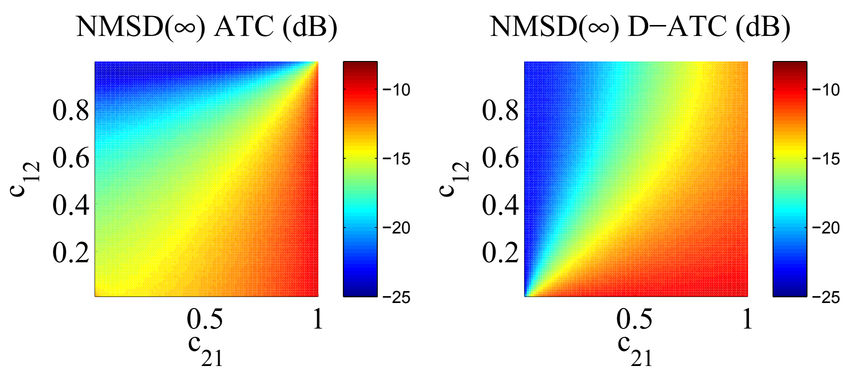


Figure 3.3: Steady-state Network MSD for 2 nodes with different step sizes as a function of the (fixed) combination weights c_{12} and c_{21}.

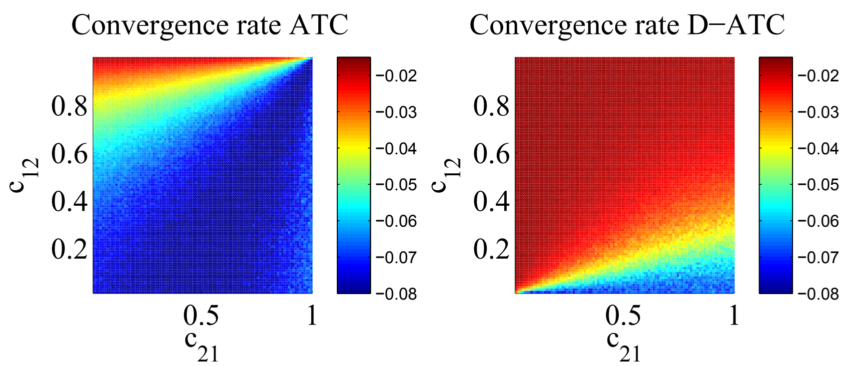


Figure 3.4: Network MSD convergence rate for 2 nodes with different step sizes as a function of the (fixed) combination weights c_{12} and c_{21}.

3.3 Theoretical analysis of D-ATC

In this section, we analyze the performance of the D-ATC diffusion strategy both in stationary and non-stationary scenarios using two alternative techniques. In the first subsection, we analyze the transient behavior of the network using the classical analysis for adaptive systems [35, 57]. Later, we derive expressions for the steady-state NMSD using the energy conservation method [105, 106]. Although, we assume NLMS adaptation for all nodes, the analysis can be straightforwardly extended to the LMS adaptation case.

3.3.1 Data model and definitions

We start by introducing two assumptions to make the following analysis more tractable:

A3.1- Variations of the unknown parameter vector $\mathbf{w}_o(n)$ follow a *random-walk* model [106]. According to this widespread model, the optimal solution varies in a nonstationary environment as

$$\mathbf{w}_o(n) = \mathbf{w}_o(n-1) + \mathbf{q}(n), \tag{3.11}$$

where $\mathbf{q}(n)$ is a zero-mean, independent and identically distributed (i.i.d.) vector with positive-definite autocorrelation matrix $\mathbf{Q} = \mathbb{E}\{\mathbf{q}(n)\mathbf{q}^T(n)\}$, independent of the initial conditions $\boldsymbol{\psi}_k(0)$, $\mathbf{w}_k(0)$, and of $\{\mathbf{u}_k(n'), v_k(n')\}$ for all k and n'. Although this model implies that the covariance matrix of $\mathbf{w}_o(n)$ diverges as $n \to \infty$, it has been commonly used in the literature to keep the analysis of adaptive systems simple [106] and, therefore, it is assumed throughout this thesis. Additionally, assuming $\mathbf{Q} = \mathbf{0}_M$ we can particularize the analysis for the stationary case.

A3.2- Input regressors are zero-mean and have covariance matrix $\mathbf{R}_k = \mathbb{E}\{\mathbf{u}_k(n)\mathbf{u}_k^T(n)\}$. Furthermore, they are spatially independent, i.e.,

$$\mathbb{E}\{\mathbf{u}_k(n)\mathbf{u}_\ell^T(n)\} = \mathbf{0}_M, \quad k \neq \ell.$$

This assumption is widely employed in the analysis of diffusion algorithms and is realistic in many practical applications [107]. Furthermore, the noise processes $\{v_k(n)\}$ are assumed to be temporally white and spatially independent, i.e.,

$$\mathbb{E}\{v_k(n)v_k(n')\} = 0, \quad \text{for all} \;\; n \neq n',$$
$$\mathbb{E}\{v_k(n)v_\ell(n')\} = 0, \quad \text{for all} \;\; n, n' \;\; \text{whenever} \;\; k \neq \ell.$$

Additionally, noise is assumed to be independent (not only uncorrelated) of the regression data $\mathbf{u}_\ell(n')$, so that $\mathbb{E}\{v_k(n)\mathbf{u}_\ell(n')\} = \mathbf{0}_M$, for all $k, \ell, n,$ and n'. As a result, $\boldsymbol{\psi}_k(n-1)$ is independent of $v_\ell(n)$ and $\mathbf{u}_\ell(n)$ for all k and ℓ. This condition matches well with simulation results for sufficiently small step sizes, even when the independence assumptions do not hold [108]. A similar condition can be observed in the behavior of stand-alone adaptive filters [57, 81, 86, 106] and is widely used in analyses of diffusion schemes [107, 108, 114, 116, 140].

As a measure of performance, we consider the MSD at each node and the NMSD, as defined in (3.10).

3.3.2 Statistical transient analysis

In this section, we present a theoretical model for the transient performance of the proposed D-ATC scheme for adaptive networks. In order to do so, we follow the classical approach in adaptive systems based on the computation of cross-covariance matrices [57]. We start by introducing two additional simplifying assumptions:

A3.3- The adaptation of the combination weights $c_{\ell k}(n)$ is slow when compared to the adaptation of the local and combined estimates. Therefore, the correlation between combination parameters and local and combined estimates can be disregarded. This assumption holds when we have fixed combination coefficients. In the case of time-varying combiners, like the ones in Section 3.4, this assumption follows from observations: simulations show that the combination weights converge slowly compared to variations in the input regressor $\mathbf{u}_k(n)$, and thus to variations on the local and combined estimates.

A3.4- Finally, for the computation of the combined estimates $\mathbf{w}_k(n)$ [Eq. (3.5)], we also assume that $\boldsymbol{\psi}_k(n) \approx \boldsymbol{\psi}_k(n-1)$, $k = 1, 2, \ldots, N$. This assumption makes the analysis more tractable and does not affect the behavior of the proposed diffusion algorithm, as observed by simulations. Note that this assumption will not be employed when studying the behavior of the local estimates $\boldsymbol{\psi}_k(n)$ [Eq. (3.8)].

To analyze adaptive diffusion strategies, it is usual to define weight-error vectors, taking into account the local and combined estimates of each node, i.e.,

$$\widetilde{\boldsymbol{\psi}}_k(n) \triangleq \mathbf{w}_\text{o}(n) - \boldsymbol{\psi}_k(n), \tag{3.12}$$

$$\widetilde{\mathbf{w}}_k(n) \triangleq \mathbf{w}_\text{o}(n) - \mathbf{w}_k(n), \tag{3.13}$$

with $k = 1, \ldots, N$. In addition, *a-priori* errors can be defined as

$$\varepsilon_k(n) = \xi_k(n) - v_k(n) = \mathbf{u}_k^T(n)\widetilde{\boldsymbol{\psi}}_k(n-1), \tag{3.14}$$

using local estimates, and

$$\breve{\varepsilon}_{k,\ell}(n) = \breve{\xi}_{k,\ell}(n) - v_k(n) = \mathbf{u}_k^T(n)\widetilde{\mathbf{w}}_\ell(n-1), \tag{3.15}$$

for combined estimates, for $k = 1, 2, \ldots, N$, and $\ell \in \mathcal{N}_k$, where $\breve{\xi}_{k,\ell}(n) = d_k(n) - \breve{y}_{k,\ell}(n)$ is the error that would be obtained when filtering the observations available at node k with the combined estimate received from node ℓ at the previous iteration.

Mean-square analysis

Subtracting both sides of (3.5) from $\mathbf{w}_\text{o}(n)$, using (3.11), and applying Assumption **A3.4**, we can approximate the weight-error vectors of the combined estimates as

$$\begin{aligned}
\widetilde{\mathbf{w}}_k(n) - \mathbf{q}(n) &= \overbrace{\left[c_{kk}(n) + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\right]}^{=1} \mathbf{w}_\text{o}(n-1) - c_{kk}(n)\boldsymbol{\psi}_k(n) \\
&\quad - \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\mathbf{w}_\ell(n-1) \\
&= c_{kk}(n)[\mathbf{w}_\text{o}(n-1) - \boldsymbol{\psi}_k(n)] + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\widetilde{\mathbf{w}}_\ell(n-1) \\
&\approx c_{kk}(n)\widetilde{\boldsymbol{\psi}}_k(n-1) + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\widetilde{\mathbf{w}}_\ell(n-1).
\end{aligned} \tag{3.16}$$

Premultiplying both sides of (3.16) by their transposes, we obtain

$$\begin{aligned}
\|\widetilde{\mathbf{w}}_k(n)\|^2 + \|\mathbf{q}(n)\|^2 - 2\mathbf{q}^T(n)\widetilde{\mathbf{w}}_k(n) \approx\ & c_{kk}^2(n)\|\widetilde{\boldsymbol{\psi}}_k(n-1)\|^2 \\
&+ \sum_{\ell\in\bar{\mathcal{N}}_k}\sum_{m\in\bar{\mathcal{N}}_k} c_{\ell k}(n)c_{mk}(n)\widetilde{\mathbf{w}}_\ell^T(n-1)\widetilde{\mathbf{w}}_m(n-1) \\
&+ 2\sum_{\ell\in\bar{\mathcal{N}}_k} c_{kk}(n)c_{\ell k}(n)\widetilde{\boldsymbol{\psi}}_k^T(n-1)\widetilde{\mathbf{w}}_\ell(n-1).
\end{aligned} \tag{3.17}$$

Then, taking expectations on both sides of (3.17) and using **A3.1**, we arrive at

$$\begin{aligned}
\text{MSD}_k(n) \triangleq \mathbb{E}\{\|\widetilde{\mathbf{w}}_k(n)\|^2\} \approx\ & \mathbb{E}\left\{c_{kk}^2(n)\|\widetilde{\boldsymbol{\psi}}_k(n-1)\|^2\right\} \\
&+ \sum_{\ell\in\bar{\mathcal{N}}_k}\sum_{m\in\bar{\mathcal{N}}_k} \mathbb{E}\left\{c_{\ell k}(n)c_{mk}(n)\widetilde{\mathbf{w}}_\ell^T(n-1)\widetilde{\mathbf{w}}_m(n-1)\right\} \\
&+ 2\sum_{\ell\in\bar{\mathcal{N}}_k} \mathbb{E}\left\{c_{kk}(n)c_{\ell k}(n)\widetilde{\boldsymbol{\psi}}_k^T(n-1)\widetilde{\mathbf{w}}_\ell(n-1)\right\} - \mathbb{E}\{\|\mathbf{q}(n)\|^2\}.
\end{aligned} \tag{3.18}$$

where we have also used the independence between vector $q(n)$ and all other variables to set expectation $\mathbb{E}\{\mathbf{q}^T(n)\mathbf{w}_k(n)\}$ equal to zero. To solve (3.18) we first need to define the cross-covariance matrices of the local, combined, and local-combined weight-error vectors, i.e.,

$$\mathbf{S}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n)\widetilde{\boldsymbol{\psi}}_m^T(n)\}, \tag{3.19}$$

$$\mathbf{W}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\mathbf{w}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\}, \tag{3.20}$$

$$\mathbf{X}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\}, \tag{3.21}$$

so that we can compute

$$\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell^T(n)\widetilde{\boldsymbol{\psi}}_m(n)\} = \text{Tr}[\mathbf{S}_{\ell m}(n)], \tag{3.22}$$

$$\mathbb{E}\{\widetilde{\mathbf{w}}_\ell^T(n)\widetilde{\mathbf{w}}_m(n)\} = \text{Tr}[\mathbf{W}_{\ell m}(n)], \tag{3.23}$$

$$\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell^T(n)\widetilde{\mathbf{w}}_m(n)\} = \text{Tr}[\mathbf{X}_{\ell m}(n)], \tag{3.24}$$

where $\text{Tr}(\cdot)$ stands for the trace of a matrix.

Thus, under Assumption **A3.3** and using (3.22)-(3.24), (3.18) can be rewritten as

$$\begin{aligned}\text{MSD}_k(n) \approx\ & \mathbb{E}\left\{c_{kk}^2(n)\right\}\text{Tr}[\mathbf{S}_{kk}(n-1)] + \sum_{\ell\in\bar{\mathcal{N}}_k}\sum_{m\in\bar{\mathcal{N}}_k}\mathbb{E}\left\{c_{\ell k}(n)c_{mk}(n)\right\}\text{Tr}[\mathbf{W}_{\ell m}(n-1)] \\ & + 2\sum_{\ell\in\bar{\mathcal{N}}_k}\mathbb{E}\left\{c_{kk}(n)c_{\ell k}(n)\right\}\text{Tr}[\mathbf{X}_{k\ell}(n-1)] - \text{Tr}(\mathbf{Q}).\end{aligned} \tag{3.25}$$

The network MSD can be estimated theoretically from the expression above by averaging the local MSD of all network nodes.

To complete the analysis, we must obtain analytical expressions for $\mathbf{S}_{\ell m}(n)$, $\mathbf{W}_{\ell m}(n)$, and $\mathbf{X}_{\ell m}(n)$. Recursions for these cross-covariance matrices are provided, with proofs, in Appendix B.

Finally, note that this analysis is valid for time-varying combination weights. Nevertheless, obtaining a theoretical model of the cross-correlation between the different combination weights is a very difficult task given their dependencies with the combined estimation vectors, which are shared among network nodes at each iteration. In [44] we presented a technique to obtain the optimal time-varying combiners and compute the performance at the same time. The drawback of that analysis is that the combiners computation is an ill-conditioned problem for some cross-covariance matrices and, consequently, the initialization and regularization of those matrices needs to be carefully tuned. For that reason, in Section 3.5.1 we limit ourselves to analyzing the performance for fixed combination parameters.

3.3.3 Energy conservation analysis

In this section, we analyze the performance of the D-ATC diffusion strategy in the mean and mean-square sense and derive expressions for the steady-state NMSD in stationary and nonstationary environments. Different from the previous analysis, and thanks to the energy conservation method [106], we directly obtain steady-state results, bypassing several of the difficulties encountered when obtaining them as a limiting case of a transient analysis. Moreover, in order to simplify the analysis, the combiners $c_{\ell k}(n)$ are assumed to be static.

First of all, for notational convenience, we collect all weight-error vectors and products $v_k(n)\mathbf{u}_k(n)$ across the network into column vectors:

$$\widetilde{\boldsymbol{w}}(n)=\text{col}\{\widetilde{\boldsymbol{\psi}}_1(n),\cdots,\widetilde{\boldsymbol{\psi}}_N(n),\widetilde{\mathbf{w}}_1(n),\cdots,\widetilde{\mathbf{w}}_N(n)\}, \tag{3.26}$$

$$\boldsymbol{s}(n)=\text{col}\{v_1(n)\mathbf{u}_1(n),v_2(n)\mathbf{u}_2(n),\cdots,v_N(n)\mathbf{u}_N(n)\}, \tag{3.27}$$

where col$\{\cdot\}$ represents the vector obtained by stacking its entries on top of each other. Note that the length of $\widetilde{\boldsymbol{w}}(n)$ is equal to $2MN$, whereas the length of $\boldsymbol{s}(n)$ is MN. We also define the length-$(2MN)$ column vector

$$\mathbf{q}_a(n) = \text{col}\{\mathbf{q}(n),\mathbf{q}(n),\cdots,\mathbf{q}(n)\}, \tag{3.28}$$

and the following $MN \times MN$ block-diagonal matrices containing the step sizes and information related to the autocorrelation matrices of the regressors:

$$\boldsymbol{\mathcal{M}}(n) = \text{diag}\{\tilde{\mu}_1(n)\mathbf{I}_M,\tilde{\mu}_2(n)\mathbf{I}_M,\cdots,\tilde{\mu}_N(n)\mathbf{I}_M\}, \tag{3.29}$$

$$\boldsymbol{\mathcal{R}}(n) = \text{diag}\{\mathbf{u}_1(n)\mathbf{u}_1^T(n),\cdots,\mathbf{u}_N(n)\mathbf{u}_N^T(n)\}, \tag{3.30}$$

where diag$\{\cdot\}$ generates a block-diagonal matrix from its arguments and $\mathbf{I}_M$ is the $M \times M$ identity matrix. Finally, we also define the following matrices containing the combination weights:

$$\mathbf{C}_1 = \text{diag}\{c_{11}, c_{22}, \cdots, c_{NN}\}, \tag{3.31}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & c_{12} & \cdots & c_{1N} \\ c_{21} & 0 & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & 0 \end{bmatrix}, \tag{3.32}$$

and their extended versions

$$\boldsymbol{\mathcal{C}}_i \triangleq \mathbf{C}_i \otimes \mathbf{I}_M, \quad i = 1, 2, \tag{3.33}$$

where $\otimes$ represents the Kronecker product of two matrices.

In the following subsections we derive the three different types of analyses normally carried out with energy conservation methods: 1) mean stability and convergence analysis of the system, 2) mean-square convergence, and 3) steady-state mean-square performance analysis.

Mean stability analysis

First, we present the mean convergence and stability analysis of our scheme. To do so, we start subtracting both sides of (3.5) and (3.8) from $\mathbf{w}_{\text{o}}(n)$. Under Assumption **A3.1**, using (3.1) and recalling that $c_{kk} + \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k} = 1$, we obtain

$$\widetilde{\boldsymbol{\psi}}_k(n) - \mathbf{q}(n) = \mathbf{A}_k(n)\widetilde{\boldsymbol{\psi}}_k(n-1) - \tilde{\mu}_k(n)v_k(n)\mathbf{u}_k(n), \tag{3.34}$$

$$\widetilde{\mathbf{w}}_k(n) - \mathbf{q}(n) = c_{kk}\mathbf{A}_k(n)\widetilde{\boldsymbol{\psi}}_k(n-1) + \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k}\widetilde{\mathbf{w}}_\ell(n-1) - c_{kk}\tilde{\mu}_k(n)v_k(n)\mathbf{u}_k(n), \tag{3.35}$$

where $\mathbf{A}_k(n) \triangleq \mathbf{I}_M - \tilde{\mu}_k(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)$.

From (3.34) and (3.35), using the definitions (3.26)-(3.33) and following algebraic manipulations similar to those of [107], we obtain the following equation characterizing the evolution of the weight-error vectors:

$$\widetilde{\boldsymbol{w}}(n) - \mathbf{q}_a(n) = \boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1) - \boldsymbol{z}(n), \tag{3.36}$$

where

$$\begin{aligned}
\boldsymbol{\mathcal{B}}(n) &\triangleq \begin{bmatrix} \boldsymbol{\mathcal{B}}_{11}(n) & \mathbf{0}_{(MN)} \\ \boldsymbol{\mathcal{B}}_{21}(n) & \boldsymbol{\mathcal{B}}_{22} \end{bmatrix}, \\
\boldsymbol{\mathcal{B}}_{11}(n) &= \mathbf{I}_{(MN)} - \boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n), \\
\boldsymbol{\mathcal{B}}_{21}(n) &= \boldsymbol{\mathcal{C}}_1^T[\mathbf{I}_{(MN)} - \boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n)], \\
\boldsymbol{\mathcal{B}}_{22} &= \boldsymbol{\mathcal{C}}_2^T, \\
\boldsymbol{z}(n) &\triangleq [\boldsymbol{\mathcal{M}}(n)\boldsymbol{s}(n) \quad \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{M}}(n)\boldsymbol{s}(n)]^T.
\end{aligned}$$

Under Assumption **A3.2**, all regressor vectors $\mathbf{u}_k(n)$ are independent of $\widetilde{\boldsymbol{\psi}}_\ell(n-1)$ and $\widetilde{\mathbf{w}}_\ell(n-1)$ for $k, \ell = 1, 2, \cdots, N$. Furthermore, independence of the noise w.r.t. the rest of variables implies that $\mathbb{E}\{\boldsymbol{s}(n)\} = \mathbb{E}\{\boldsymbol{z}(n)\} = \mathbf{0}_M$. Thus, taking expectations on both sides of (3.36) and recalling that $\mathbb{E}\{\mathbf{q}_a(n)\} = \mathbf{0}_{2MN}$, we obtain

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}(n)\} = \mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}\mathbb{E}\{\widetilde{\boldsymbol{w}}(n-1)\}. \tag{3.37}$$

A necessary and sufficient condition for the mean stability of (3.37) is that the spectral radius of $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ is less than or equal to one, i.e.,

$$\rho(\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}) = \max_i\{\lambda_i\} \leq 1,$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument and λ_i, with $i = 1, 2, \cdots, 2MN$, are the eigenvalues of $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ [107]. Since $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ is a block-triangular matrix, its eigenvalues are the eigenvalues of the blocks of its main diagonal, i.e., the eigenvalues of $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}$ and $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{22}\}$ [62].

Focusing first on matrix $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}$, we notice that it is also a block-diagonal matrix, so the step sizes need to be selected to guarantee

$$\rho(\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}) = \max_{1\leq k\leq N} \rho\left(\mathbf{I}_M - \mu_k\overline{\mathbf{R}}_k\right) \leq 1, \tag{3.38}$$

where

$$\overline{\mathbf{R}}_k \triangleq \mathbb{E}\left\{\frac{\mathbf{u}_k(n)\mathbf{u}_k^T(n)}{\delta + \|\mathbf{u}_k(n)\|^2}\right\}.$$

Condition (3.38) will be ensured if the step sizes μ_k satisfy [106]

$$0 < \mu_k < 2, \quad \text{for } k = 1, 2, \cdots, N. \tag{3.39}$$

This condition, which is a well-known result for the NLMS algorithm [106], guarantees that the local estimators $\{\boldsymbol{\psi}_k(n)\}$ are asymptotically unbiased, i.e., $\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_k(n)\} \rightarrow \mathbf{0}_M$ as $n \rightarrow \infty$ for all nodes of the network.

For the spectral radius of $\boldsymbol{\mathcal{B}}_{22} = \boldsymbol{\mathcal{C}}_2^T$, we can rely on the following bound from [62]:

$$\rho(\boldsymbol{\mathcal{B}}_{22}) \leq \|\boldsymbol{\mathcal{B}}_{22}\|_\infty = \max_k \sum_{\ell \in \bar{\mathcal{N}}_k} |c_{\ell k}|. \tag{3.40}$$

A sufficient (but not necessary) condition to guarantee $\rho(\boldsymbol{\mathcal{B}}_{22}) \leq 1$ is to keep all combination weights non-negative. In effect, since the sum of all combiners associated to a node is one, using non-negative weights we have

$$\rho(\boldsymbol{\mathcal{B}}_{22}) \leq \max_k \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k} = \max_k (1 - c_{kk}) \leq 1. \tag{3.41}$$

When combiners are learned by the network, non-negativity constraints can be applied at every iteration to ensure mean stability. Although our derivations show that this is just a sufficient condition, we should mention that in [44, 45], where we allowed combination weights to become negative, the network showed some instability problems and the application of these constraints resulted in the removal of these instability issues.

Mean-square convergence

We present next a mean-square performance analysis, following the energy conservation framework of [106]. First, let $\boldsymbol{\Sigma}$ denote an arbitrary nonnegative definite $2MN \times 2MN$ matrix. Different choices of $\boldsymbol{\Sigma}$ allow us to obtain different performance measurements of the network [108].

Thus, computing the weighted squared norm on both sides of (3.36) using $\boldsymbol{\Sigma}$ as the weighting matrix, we arrive at

$$\begin{aligned} &\widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n) - \widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n) - \mathbf{q}_a^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n) + \mathbf{q}_a^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n) \\ &\quad = \widetilde{\boldsymbol{w}}^T(n-1)\boldsymbol{\mathcal{B}}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1) + \boldsymbol{z}^T(n)\boldsymbol{\Sigma}\boldsymbol{z}(n) - 2\boldsymbol{z}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1). \end{aligned} \tag{3.42}$$

As before, independence of the noise terms in $\boldsymbol{z}(n)$ with respect to all other variables implies that the last element in (3.42) vanishes under expectation. Furthermore, under Assumption **A3.1**, we can verify that

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n)\} = \mathbb{E}\{\mathbf{q}_a^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n)\} = \mathbb{E}\{\mathbf{q}_a^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n)\} = \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a), \tag{3.43}$$

where

$$\mathbf{Q}_a \triangleq \mathbb{E}\{\mathbf{q}_a(n)\mathbf{q}_a^T(n)\} = \mathbf{J}_{(2N)} \otimes \mathbf{Q},$$

with $\mathbf{J}_{(2N)}$ a $2N \times 2N$ matrix with all entries equal to one. Defining the matrices

$$\widetilde{\mathbf{R}}_k \triangleq \mathbb{E}\left\{\frac{\mathbf{u}_k(n)\mathbf{u}_k^T(n)}{[\delta + \|\mathbf{u}_k(n)\|^2]^2}\right\}, \tag{3.44}$$

$$\boldsymbol{\mathcal{S}} \triangleq \mathrm{diag}\{\mu_1^2\sigma_{v_1}^2\widetilde{\mathbf{R}}_1,\ \mu_2^2\sigma_{v_2}^2\widetilde{\mathbf{R}}_2,\ \cdots,\ \mu_N^2\sigma_{v_N}^2\widetilde{\mathbf{R}}_N\}, \tag{3.45}$$

$$\boldsymbol{\mathcal{Z}} \triangleq \mathbb{E}\{\boldsymbol{z}(n)\boldsymbol{z}^T(n)\} = \begin{bmatrix} \boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1 \\ \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1 \end{bmatrix}, \tag{3.46}$$

using (3.43), and taking expectations of both sides of (3.42), we obtain

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\} = \mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}(n-1)\|_{\boldsymbol{\mathcal{B}}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)}^2\right\} + \mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{Z}}) + \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a), \tag{3.47}$$

where $\|\mathbf{x}\|_{\boldsymbol{\Sigma}}^2$ denotes the weighted squared norm $\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$.

In order to make the analysis more tractable, we will replace the random matrix $\boldsymbol{\mathcal{B}}(n)$ by its steady-state mean value $\overline{\boldsymbol{\mathcal{B}}} = \lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$, which is equivalent to replacing matrix $\mathbf{u}_k(n)\mathbf{u}_k^T(n)/[\delta + \|\mathbf{u}_k(n)\|^2]$ by its mean $\overline{\mathbf{R}}_k$. In a sense, this approximation amounts to an ergodicity assumption on the regressors, which is a common assumption in statistical analysis of adaptive filters [106]. Thus, (3.47) reduces to

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\} \approx \mathbb{E}\left\{\{\|\widetilde{\boldsymbol{w}}(n-1)\|_{\overline{\boldsymbol{\mathcal{B}}}^T\boldsymbol{\Sigma}\overline{\boldsymbol{\mathcal{B}}}}^2\right\} + \mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{Z}}) + \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a). \tag{3.48}$$

As in [108], the convergence rate of the series is governed by $[\rho(\overline{\boldsymbol{\mathcal{B}}})]^2$, in terms of the spectral radius of $\overline{\boldsymbol{\mathcal{B}}}$. From Section 3.3.3, we can obtain a superior limit for $\rho(\overline{\boldsymbol{\mathcal{B}}})$, which is given by

$$\rho(\overline{\boldsymbol{\mathcal{B}}}) \leq \max\left\{\max_{k,i}\left[1 - \mu_k\lambda_i(\overline{\mathbf{R}}_k)\right],\ \max_{k}\left(1 - c_{kk}\right)\right\}. \tag{3.49}$$

Choosing μ_k into the interval (3.39) and imposing non-negativity constraints to the combiners, $\rho(\overline{\boldsymbol{\mathcal{B}}}) \leq 1$ and the convergence of $\lim_{n\to\infty}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\}$ is ensured. Furthermore, from the superior limit (3.49) we can see that, in the worst case, our diffusion scheme would converge with the same convergence rate of the noncooperative solution, whose spectral radius is $\max_{k,i}\left\{1-\mu_k\lambda_i(\overline{\mathbf{R}}_k)\right\}$ (considering that all the nodes are adapted using NLMS). However, we will show by means of simulations that in practice this limit is very conservative and the proposed diffusion scheme converges much faster than the noncooperative solution.

Steady-state MSD performance

It it important to notice that variance relations similar to (3.48) have often appeared in the performance analysis of diffusion schemes [108]. Iterating (3.48) and taking the limit as $n \to \infty$, we conclude that (see, e.g., [117])

$$\lim_{n\to\infty} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|^2_{\boldsymbol{\Sigma}}\} \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Sigma}]. \tag{3.50}$$

To obtain analytical expressions for the steady-state MSD of the network and of its individual nodes, we will replace $\boldsymbol{\Sigma}$ by the the following matrices

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} \mathbf{0}_{NM} & \mathbf{0}_{NM} \\ \mathbf{0}_{NM} & \frac{1}{N}\mathbf{I}_{NM} \end{bmatrix}, \tag{3.51}$$

$$\boldsymbol{\Upsilon}_k \triangleq \begin{bmatrix} \mathbf{0}_{NM} & \mathbf{0}_{NM} \\ \mathbf{0}_{NM} & \mathbf{E}_k \otimes \mathbf{I}_M \end{bmatrix}, \tag{3.52}$$

where $\mathbf{E}_k$ is an $N \times N$ zero matrix, except in the element (k,k), that is equal to one. Replacing $\boldsymbol{\Sigma}$ in (3.50) by either $\boldsymbol{\Gamma}$ or $\boldsymbol{\Upsilon}_k$, the MSD performance of the network and of its individual nodes can be expressed, respectively, by

$$\mathrm{NMSD}(\infty) \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Gamma}], \tag{3.53}$$

$$\mathrm{MSD}_k(\infty) \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Upsilon}_k]. \tag{3.54}$$

Since $\overline{\boldsymbol{\mathcal{B}}}$ is lower triangular, the matrix $\overline{\boldsymbol{\mathcal{B}}}^j$ is given by

$$\overline{\boldsymbol{\mathcal{B}}}^j = \begin{bmatrix} \overline{\boldsymbol{\mathcal{B}}}^j_{11} & \mathbf{0}_{(MN)} \\ \bar{\boldsymbol{\mathcal{X}}}(j) & \overline{\boldsymbol{\mathcal{B}}}^j_{22} \end{bmatrix}, \tag{3.55}$$

being

$$\bar{\boldsymbol{\mathcal{X}}}(j)=\sum_{k=0}^{j-1}\overline{\boldsymbol{\mathcal{B}}}_{22}^{k}\overline{\boldsymbol{\mathcal{B}}}_{21}\overline{\boldsymbol{\mathcal{B}}}_{11}^{j-k-1}=\sum_{k=0}^{j-1}[\boldsymbol{\mathcal{C}}_2^T]^k\boldsymbol{\mathcal{C}}_1^T\left[\mathbf{I}_{(MN)}-\boldsymbol{\mathcal{L}}\right]^{j-k}, \tag{3.56}$$

where we have defined

$$\boldsymbol{\mathcal{L}}\triangleq\lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n)\}=\text{diag}\{\mu_1\overline{\mathbf{R}}_1,\mu_2\overline{\mathbf{R}}_2,\cdots,\mu_N\overline{\mathbf{R}}_N\}. \tag{3.57}$$

Replacing (3.55) and (3.46) in (3.53) and (3.54), we arrive at

$$\begin{aligned}\text{NMSD}(\infty)\approx\frac{1}{N}\sum_{j=0}^{\infty}\text{Tr}\Big[&\bar{\boldsymbol{\mathcal{X}}}(j)(\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j)+2(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j)\\&+(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1+\boldsymbol{\mathcal{Q}})\boldsymbol{\mathcal{C}}_2^j\Big],\end{aligned} \tag{3.58}$$

$$\begin{aligned}\text{MSD}_k(\infty)\approx\sum_{j=0}^{\infty}\text{Tr}\Big[\Big(&\bar{\boldsymbol{\mathcal{X}}}(j)(\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j)+2(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j)\\&+(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1+\boldsymbol{\mathcal{Q}})\boldsymbol{\mathcal{C}}_2^j\Big)\mathbf{E}_k\otimes\mathbf{I}_M\Big],\end{aligned} \tag{3.59}$$

where $\boldsymbol{\mathcal{Q}}=\mathbf{J}_N\otimes\mathbf{Q}$. Note that the $NM\times NM$ matrix $\boldsymbol{\mathcal{Q}}$ is similar to matrix $\mathbf{Q}_a$, but has half its size.

In order to compute the theoretical steady-state MSD using (3.58) and (3.59), we still have to obtain approximations for matrices $\overline{\mathbf{R}}_k$ and $\widetilde{\mathbf{R}}_k$, which appear in $\bar{\boldsymbol{\mathcal{X}}}(j)$ and $\boldsymbol{\mathcal{S}}$, respectively. For this purpose, we assume that

A3.7- The number of coefficients M is large enough for each element of the matrix $\mathbf{u}_k(n)\mathbf{u}_k^T(n)$ to be approximately independent from $\sum_{l=0}^{M-1}|u(n-l)|^2$. This is equivalent to applying the averaging principle of [104], since for large M, $\|\mathbf{u}_k(n)\|^2$ tends to vary slowly compared to the individual entries of $\mathbf{u}_k(n)\mathbf{u}_k^T(n)$.

A3.8- The regressors $\mathbf{u}_k(n)$, $k=1,2,\ldots,N$ are formed by a tapped-delay line with Gaussian entries and the regularization factor is equal to zero ($\delta=0$). This is a common assumption in the analysis of adaptive filters and leads to reasonable analytical results [57].

Under **A3.7** and **A3.8**, we obtain the following approximations from [31]:

$$\overline{\mathbf{R}}_k \approx \frac{\mathbf{R}_k}{\sigma_{u_k}^2 (M-2)}, \tag{3.60}$$

$$\widetilde{\mathbf{R}}_k \approx \frac{\mathbf{R}_k}{\sigma_{u_k}^4 (M-2)(M-4)}. \tag{3.61}$$

Under these additional assumptions, we obtain a model to compute the steady-state MSD of the network and of its individual nodes, which can be summarized as follows: (i) compute the matrices of the combination weights using (3.31)-(3.33) and the matrix $\boldsymbol{\mathcal{Q}}$, i.e, according to the environment variation; (ii) use the approximations (3.60) and (3.61) in the computation of matrices $\boldsymbol{\mathcal{S}}$ and $\bar{\boldsymbol{\mathcal{X}}}(j)$, defined respectively by (3.45) and (3.56); and finally, (iii) use these matrices in (3.58) and (3.59).

To sum up, in this section we have presented two alternative analyses for our diffusion scheme. The first one is valid both for the transient and steady-state performance but it is somewhat more involved to compute because of the recursive expressions of cross-covariance matrices. In the second one, we bypass this problem using an energy conservation approach, obtaining formulas for steady-state performance and bounds on the convergence rate. This analysis also serves as a justification for using convex combination weights, as this is a (sufficient) condition —together with the stability of the individual nodes— for convergence.

3.4 Adaptive combiners for D-ATC scheme

As shown in Section 3.2, the implementation of adaptive combiners is crucial for heterogeneous networks, whose nodes operate under different conditions, e.g., different step sizes in the adaptation step. For instance, in such case the combiners should favor the diffusion of the estimates of the fastest nodes during network convergence, whereas in steady state the network should favor the nodes with better SNR and smaller adaptation step size, as they produce lower steady-state misadjustment.

In this section, we present two strategies for learning the combiners which are suitable for our Decoupled ATC scheme. These two strategies are based on an approximate minimization of the Network Mean-Square Error at each step n, NMSE(n):

$$\text{NMSE}(n) = \frac{1}{N}\sum_{k=1}^{N} \text{MSE}_k(n) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\{\check{\xi}_k^2(n)\}, \tag{3.62}$$

where $\check{\xi}_k(n) = d_k(n) - \check{y}_k(n) = d_k(n) - \mathbf{u}_k^T(n)\mathbf{w}_k(n-1)$ represents the error at node k using combined estimates, while $\check{y}_k(n)$ stands for the corresponding combined output.

Since every node can only manage its own combination coefficients, and this only affects the local MSE, $\text{MSE}_k(n)$, in the following we derive rules that precisely attempt to minimize $\text{MSE}_k(n)$ at each node by relying on available local information. Note that this approach contrasts with the strategies in [114, 116] where an approximation or bound of the steady-state NMSD is minimized. Minimizing the NMSE indirectly minimizes the NMSD as well, and has the advantage that approximations of (3.62) can be easily obtained whereas the estimation of the NMSD would require access to the (unknown) optimal vector. For this reason, MSE-based cost functions are normally used in adaptive filtering as a design criterion.

Finally, let us emphasize that we are considering convex combination coefficients, i.e., $c_{\ell k} \geq 0 \; \forall k, \ell \in \{1, \cdots, N\}$ and $\sum_{\ell \in \mathcal{N}_k} c_{\ell k} = 1, \forall \in \{1, \cdots, N\}$. However, note that a direct application of the algorithms below may give rise to values of $c_{\ell k}(n)$ outside range $[0, 1]$. Therefore, in order to satisfy the non-negativity constraint of Section 3.3.3 to guarantee stability (and also following the criterion of other works in this field, e.g., [26, 107, 114, 116, 140]), we constrain the values of $c_{\ell k}(n)$ to remain in the desired interval $[0, 1]$ at each iteration. To do this, if any $c_{\ell k}(n)$ results negative after its update, we simply set it to zero and then rescale the remaining combination weights so that they sum up to one.

3.4.1 Affine projection algorithm

In this subsection we present an Affine Projection Algorithm (APA) for the stochastic minimization of the MSE in (3.62). First, we stack the combination coefficients $c_{\ell k}$ of node k, with $\ell \in \bar{\mathcal{N}}_k$, in a length-$\bar{N}_k$ vector $\bar{\mathbf{c}}_k(n)$. Doing so, we can write

$$c_{kk}(n) = 1 - \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k} = 1 - \mathbb{1}_{\bar{N}_k}^T \bar{\mathbf{c}}_k(n). \tag{3.63}$$

Then, defining $y_{\ell k}(n) = \mathbf{w}_\ell^T(n-1)\mathbf{u}_k(n)$ and $\tilde{y}_{\ell k} = y_{\ell k}(n) - y_k(n)$ with $\ell \in \bar{\mathcal{N}}_k$, collecting all these differences into a column vector $\tilde{\mathbf{y}}_k(n)$, and using (3.63), $\text{MSE}_k(n)$ can be rewritten as

$$\text{MSE}_k(n) = \mathbb{E}\left\{\left[\xi_k(n) - \bar{\mathbf{c}}_k^T(n)\tilde{\mathbf{y}}_k(n)\right]^2\right\}. \tag{3.64}$$

Applying the regularized Newton's method [106] to minimize (3.64), we obtain

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c[\epsilon \mathbf{I}_{\bar{N}_k} + \mathbf{R}_{\tilde{\mathbf{y}}_k}]^{-1}[\mathbf{R}_{\xi_k,\tilde{\mathbf{y}}_k} - \mathbf{R}_{\tilde{\mathbf{y}}_k}\bar{\mathbf{c}}_k(n-1)] \quad (3.65)$$

where μ_c is a step size to control the adaptation of $\bar{\mathbf{c}}_k(n)$, $\mathbf{R}_{\tilde{\mathbf{y}}_k}$ is the autocorrelation matrix of vector $\tilde{\mathbf{y}}_k(n)$, $\mathbf{R}_{\xi_k,\tilde{\mathbf{y}}_k}$ is the cross-correlation vector between $\tilde{\mathbf{y}}_k(n)$ and $\xi_k(n)$, ϵ is a small regularization parameter to avoid division by zero, and $\mathbf{I}_{\bar{N}_k}$ represents the $\bar{N}_k \times \bar{N}_k$ identity matrix, with $\bar{N}_k$ the cardinal of $\bar{\mathcal{N}}_k$.

Replacing $\mathbf{R}_{\tilde{\mathbf{y}}_k}$ and $\mathbf{R}_{\xi_k,\tilde{\mathbf{y}}_k}$ by their approximations based on averages over the L most recent values of $\tilde{\mathbf{y}}_k(n)$ and $\xi_k(n)$ [106], we obtain a regularized affine projection algorithm for the adaptation of $\bar{\mathbf{c}}_k(n)$:

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c[\epsilon \mathbf{I}_{\bar{N}_k} + \tilde{\mathbf{Y}}_k^T(n)\tilde{\mathbf{Y}}_k(n)]^{-1}\tilde{\mathbf{Y}}_k^T(n)[\xi_k(n) - \tilde{\mathbf{Y}}_k(n)\bar{\mathbf{c}}_k(n-1)], \quad (3.66)$$

where $\tilde{\mathbf{Y}}_k(n)$ is an $L \times \bar{N}_k$ matrix whose L rows correspond with the last L values of vector $\tilde{\mathbf{y}}_k(n)$, and $\xi_k(n) = [\xi_k(n), \xi_k(n-1), \cdots, \xi_k(n-L+1)]^T$. This recursion requires the inversion of an $\bar{N}_k \times \bar{N}_k$ matrix at each iteration, resulting in an attractive implementation if the projection order L is larger than the number of neighbors of node k, $\bar{N}_k$. Otherwise, if for any node $\bar{N}_k > L$, we can invoke the matrix inversion lemma [106] to rewrite (3.66) as

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c\tilde{\mathbf{Y}}_k^T(n)[\epsilon \mathbf{I}_L + \tilde{\mathbf{Y}}_k(n)\tilde{\mathbf{Y}}_k^T(n)]^{-1}[\xi_k(n) - \tilde{\mathbf{Y}}_k(n)\bar{\mathbf{c}}_k(n-1)], \quad (3.67)$$

which requires the inversion of an $L \times L$ matrix.

Equations (3.66) —or (3.67)— and (3.63), constitute the ϵ-APA algorithm for adapting the combiners at each node.

3.4.2 Least-Squares algorithm

In this subsection, we follow a Least-Squares approach. Instead of minimizing (3.62) using a stochastic minimization algorithm as in the previous section, we replace $\text{MSE}_k(n)$ by the following related cost function [106],

$$J_k(n) = \sum_{i=1}^{n} \beta(n,i)\check{\xi}_k^2(n,i), \quad (3.68)$$

where $\beta(n,i)$ is a temporal weighting window, and

$$\check{\xi}_k(n,i) = d_k(i) - y_k(n,i), \text{ with} \tag{3.69}$$

$$\begin{aligned} y_k(n,i) &= \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n) y_{\ell k}(i) + \left[1 - \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\right] y_k(i) \\ &= y_k(i) + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\left[y_{\ell k}(i) - y_k(i)\right] \end{aligned} \tag{3.70}$$

representing the combined output of node k at time i when the outputs of all nodes belonging to $\mathcal{N}_k$ are combined using the combiners at time n. Introducing (3.70) into (3.69), we obtain

$$\check{\xi}_k(n,i) = \xi_k(i) + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\left[y_k(i) - y_{\ell k}(i)\right]. \tag{3.71}$$

Taking now the derivatives of (3.68) with respect to each combination weight $c_{mk}(n)$, with $m = 1, 2, \ldots, \bar{N}_k$, we obtain

$$\frac{\partial J_k(n)}{\partial c_{mk}(n)} = 2\sum_{i=1}^{n} \beta(n,i)\check{\xi}_k(n,i)\left[y_k(i) - y_{mk}(i)\right]. \tag{3.72}$$

Replacing (3.71) in (3.72), setting the result to zero, and after some algebraic manipulations, we obtain

$$\sum_{i=1}^{n}\sum_{\ell\in\bar{\mathcal{N}}_k} \beta(n,i) c_{\ell k}(n)\tilde{y}_{\ell k}(i)\tilde{y}_{mk}(i) = \sum_{i=1}^{n} \beta(n,i)\xi_k(i)\tilde{y}_{mk}(i). \tag{3.73}$$

We can then write for each node k a system with $\bar{N}_k$ equations of the form (3.73) that, introducing the usual matrix notation, reads

$$\mathbf{P}_k(n)\bar{\mathbf{c}}_k(n) = \mathbf{z}_k(n), \tag{3.74}$$

where $\mathbf{P}_k(n)$ is a square symmetric matrix of size $\bar{N}_k$ with components

$$[\mathbf{P}_k(n)]_{p,q} = \sum_{i=1}^{n} \beta(n,i)\tilde{y}_{(\bar{b}_k^{(p)},k)}(i)\tilde{y}_{(\bar{b}_k^{(q)},k)}(i), \tag{3.75}$$

with $p, q = 1, 2, \ldots, \bar{N}_k$. We introduce the index $\bar{b}_k^{(p)}$ which is the index of the p-th neighbor of k. In addition, $\mathbf{z}_k(n)$ is a column vector of length $\bar{N}_k$, whose p^{th} element is given by

$$z_k^{(p)}(n) = \sum_{i=1}^{n} \beta(n,i)\xi_k(i)\tilde{y}_{(\bar{b}_k^{(p)},k)}(i), \tag{3.76}$$

for $p = 1, 2, \ldots, \bar{N}_k$. Thus, the solution of the problem is obtained from (3.74) using Tikhonov regularization method [125] as

$$\bar{\mathbf{c}}_k(n) = \left(\mathbf{P}_k(n) + \epsilon \mathbf{I}_{\bar{N}_k}\right)^{-1} \mathbf{z}_k(n), \tag{3.77}$$

where a small regularization constant ϵ is required since $\mathbf{P}_k(n)$ could be ill-conditioned [45]. Similarly to the case of combination of multiple filters [11], $\mathbf{P}_k(n)$ can be interpreted as the autocorrelation matrix of a vector $\tilde{\mathbf{y}}_k(n)$ while $\mathbf{z}_k(n)$ could be seen as the cross-correlation vector between $\tilde{\mathbf{y}}_k(n)$ and $\xi_k(n)$.

Temporal weighting window

The selection of the temporal weighting window $\beta(n, i)$ deserves some discussion. Typical choices for this kind of window are the rectangular window with length L, i.e.,

$$\beta(n, i) = \begin{cases} 1, & n - i < L \\ 0, & n - i \geq L, \end{cases} \tag{3.78}$$

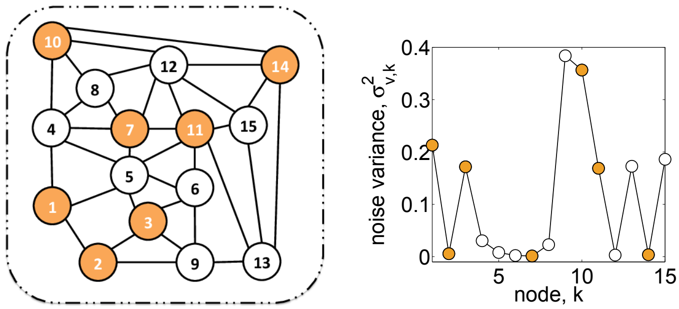
or the exponential window

$$\beta(n, i) = \gamma^{n-i}, \tag{3.79}$$

where γ is a forgetting factor $0 \leq \gamma < 1$.

The choice of the window length L in the rectangular window is generally not straightforward, and is subject to a well-known trade-off between convergence capabilities (faster for small L) and steady-state performance (better for large L). First of all, $L \gg \bar{N}_k$ to guarantee that matrix $\mathbf{P}_k(n)$ is well conditioned. Secondly, we should remark that the estimation of the optimal combination weights is itself a time-varying problem and according to the literature [89, 88] there is an optimal window length that depends on the particular filtering scenario.

Selection of forgetting factor γ in the exponential window suffers from a similar trade-off but this window has two remarkable advantages with respect to a rectangular one: 1) It is more efficient in terms of memory and computation; and 2) it allows a recursive implementation. In addition, as we show in the experiments, it outperforms other state-of-the art approaches.



(a) (b)

Figure 3.5: (a) Network topology for the simulation experiments: orange shaded nodes are adapted with $\mu_k = 0.1$ and the rest with $\mu_k = 1$. (b) Noise power $\sigma^2_{v,k}$ at each node in the network.

3.5 Simulation results

In this section, we present a number of simulation results to illustrate the behavior of D-ATC and the proposed adaptive combiners rules in stationary and tracking estimation scenarios. We simulate the 15-node network of Fig. 3.5, where, as in the example of Section 3.2.3, all the nodes employ NLMS adaptation. The nodes step sizes are taken as $\mu_k \in \{0.1, 1\}$ as illustrated in Fig. 3.5. The input signals $\mathbf{u}_k(n)$ follow a multidimensional Gaussian with zero mean and the same scalar covariance matrix, $\sigma_u^2\mathbf{I}_M$, with $\sigma_u^2 = 1$. Unless otherwise stated, the observation noise $v_k(n)$ at each node is also Gaussian distributed with zero mean and variance $\sigma^2_{v,k}$ randomly chosen between $[0, 0.4]$ as shown in Fig. 3.5(b). For the stationary estimation problem, the unknown parameter vector $\mathbf{w}_{\rm o}$ is a length-50 vector with values uniformly taken from range $[-1, 1]$. As a tracking model, we use the one presented in equation (3.11). Finally, in all the empirical curves, 500 experiments have been averaged.

First, we present a set of experiments with the aim to validate the theoretical analyses of Section 3.3. Then, we compare the behavior of our rules to state-of-the-art adaptive combination algorithms for standard ATC [116], both in stationary and tracking scenarios.

3.5.1 Validation of the theoretical analysis for D-ATC

In this section, we carry out some numerical simulations to validate the analyses presented in Section 3.3. In order to do so, we compare the theoretical (according to each model) and empirical performance for the nodes of a D-ATC scheme with Metropolis combiners [107] in the stationary estimation scenario described above. The metropolis rule is defined as

$$c_{\ell k} = \begin{cases} 1/\max\{n_k, n_\ell\}, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - \sum_{m \in \bar{\mathcal{N}}_k} c_{mk}, & \text{if } k = \ell \\ 0, & \text{otherwise} \end{cases} \tag{3.80}$$

where n_k denotes the degree of node k, i.e., its number of neighbors.

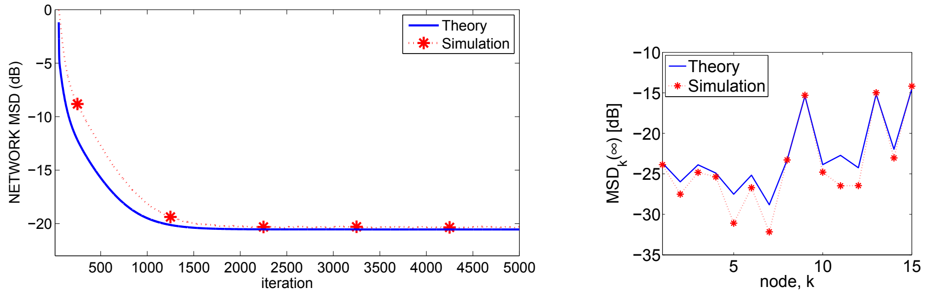
Note that these weights are not optimal, as we can deduce from the preliminary analysis of Section 3.2.3. However, this is not an issue since our objective in this subsection is just to show that the analysis correctly predicts the performance of each individual node, as well as the NMSD. Although we consider just the case of Metropolis combination rule, other rules, e.g., uniform combiners, lead to similar conclusions about the accuracy of the analyses.

Validation of transient analysis

Firstly, we carry out some numerical simulations to validate the transient analysis of Section 3.3.2. Fig. 3.6(a) displays the curve for the network performance and the theoretical MSD that is computed using (3.25). In Fig. 3.6(b) we also show the theoretical and empirical steady-state MSD of each node. The deviation both in convergence and steady state is not very significant and, more importantly, the model predicts well the qualitative behavior of the network MSD and the time instants where the MSD has roughly converged to -20 dB.

Validation of energy conservation analysis

Secondly, we aim to validate the steady-state model of Section 3.3.3 based on energy conservation analysis. In Fig. 3.7, we plot the steady-state MSD for 4 different scenarios where the step sizes μ_k and the noise variances $\sigma_{v,k}^2$ have been varied from



(a) Network MSD

(b) Steady-state MSD at each node

Figure 3.6: Comparison between theoretical transient model (blue solid line) and empirical performance (red dotted line with star markers) in terms of: (a) Network MSD learning curve, and (b) steady-state MSD at each node.

those in Fig. 3.5, according to Table 3.2. From Fig. 3.7, we can conclude that the matching between the analysis and the simulation is quite good, even for the case with large step sizes, when the last part of assumption **A2** in Section 3.3.1 is less accurate.

We have also studied the accuracy of the model in tracking situations. In Fig. 3.8, we plot the steady-state NMSD for different speeds of change, i.e., values of Tr$\{\mathbf{Q}\}$. We can see that the matching is also quite good, in particular for fast speeds of changes of the optimum solution, i.e., large Tr$\{\mathbf{Q}\}$. For slow and medium speeds we observe a mismatch up to 2 dB similarly to the stationary scenario depicted in Fig. 3.7(a).

Scenario (a)	Scenario (b)	Scenario (c)	Scenario (d)
μ_k	$\mu_k/10$	μ_k	$\mu_k/10$
$\sigma^2_{v,k}$	$\sigma^2_{v,k}$	$\sigma^2_{v,k}/10$	$\sigma^2_{v,k}/10$

Table 3.2: Settings of the scenarios simulated in Fig. 3.7. The settings of the reference scenario (a) are described in Fig. 3.5.

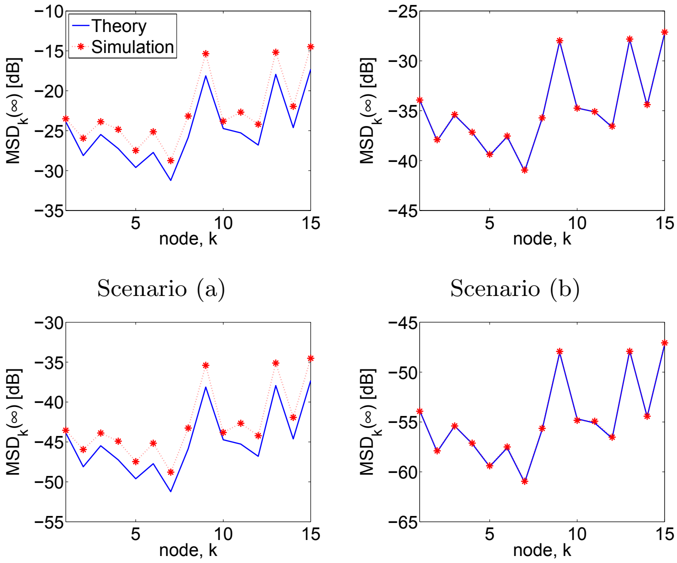


Figure 3.7: Comparison between theoretical model (blue solid line) and empirical performance (red dotted line with star markers).

3.5.2 Stationary performance of D-ATC with adaptive combiners

Before comparing the performance of D-ATC and ATC with adaptive combiners we study in Fig. 3.9 the sensitivity of the proposed combiner learning rules, APA and LS, with respect to their parameters. Note that in this stationary scenario we have introduced an abrupt change in the value of $\mathbf{w}_o$ to analyze the reconvergence of the different schemes. We observe that there is a trade-off between convergence/reconvergence speed and steady-state performance in the selection of these parameters. In fact, we can conclude that the influences of different parameters are coupled among them.

Regarding the forgetting factor γ in the LS rule, note that, when it is correctly chosen (see Fig. 3.9(b)), we can obtain a large steady-state enhancement without dramatically affecting the convergence. This was not the case with the rectangular window [45], where instability issues prevented us from using a very small regularization constant, and imposed a limit on the length the number of useful window sizes, causing degradation in steady-state performance.

Next, we compare our D-ATC scheme with adaptive combiners, with other state-

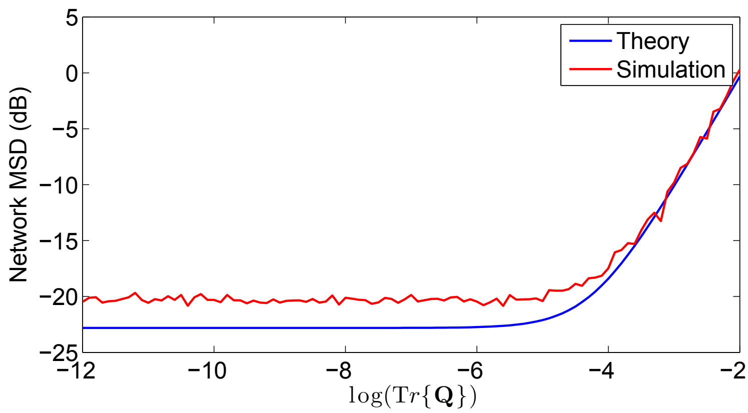
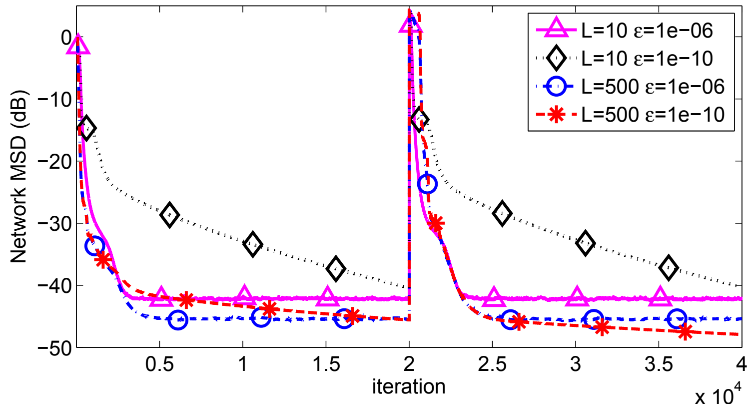


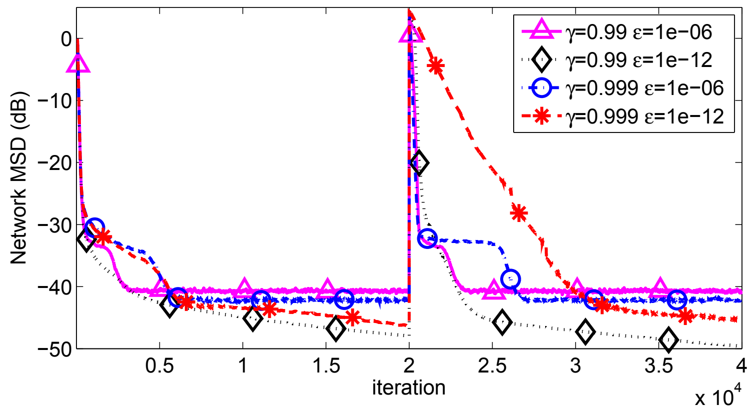
Figure 3.8: Comparison between the analysis and the simulation in a tracking scenario as a function of the logarithm of Tr$\{\mathbf{Q}\}$.

of-the-art ATC algorithms with adaptive combiners: 1) ATC with adaptive combiners proposed by Takahashi *et al.* [114], and 2) a newer approach by Tu *et al.* [107, 116]. We also include a baseline network where the nodes do not combine their estimates. The free parameters of all algorithms are chosen to maximize the steady-state performance while keeping a similar convergence rate and, for reproducibility, are given in Table 3.3.

In Fig. 3.10, we can see that D-ATC with both adaptive rules (APA and LS) significantly outperforms standard ATC. In Section 3.2.3 we saw that both ATC and D-ATC can reach a similar performance provided that the combination weights are correctly chosen. Consequently, our rules seem to be more effective in learning the combination weights for this setup. Note that adaptive rules for learning the combination weights for standard ATC [114, 116] are derived for homogeneous networks, i.e., considering that only the noise variance changes among the nodes. That explains most of the gap between both approaches. Regarding the convergence rate, the gain of the proposed scheme is not so significant but it still outperforms standard ATC with adaptive combiners. This is not surprising, as their suboptimality in terms of convergence has been made clear before, even for homogeneous networks [41, 44, 135].



(a) D-ATC with APA adaptive combiners



(b) D-ATC with LS adaptive combiners

Figure 3.9: Influence of the parameters of the algorithm used for learning the adaptive combiners in a D-ATC diffusion scheme. (a) APA adaptive combiners. (b) LS adaptive combiners.

3.5.3 Tracking performance of D-ATC with adaptive combiners

In this section, we analyze the tracking performance of our scheme with the two proposed combination rules. We compare in Fig. 3.11 the performance of D-ATC and standard ATC, both with adaptive combiners, for two speeds of change of the optimal solution. The parameters of these simulations, shown in Table 3.3, are again selected to optimize the steady-state performance, trying to keep the convergence rate of all algorithms as close as possible. In fact the steady-state NMSD of D-ATC

	D-ATC		ATC	
	APA	LS	ACW 1 [114]	ACW 2 [116]
Stationary. Fig 3.10	$L = 500$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.99$ $\epsilon = 10^{-12}$	$\alpha = 0.2$ $\epsilon = 10^{-6}$	$\nu = 0.1$
Tracking. Fig 3.11.(a)	$L = 10$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.9999$ $\epsilon = 10^{-10}$	$\alpha = 0.05$ $\epsilon = 10^{-6}$	$\nu = 0.2$
Tracking. Fig 3.11.(b)	$L = 500$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.99$ $\epsilon = 10^{-10}$	$\alpha = 0.2$ $\epsilon = 10^{-6}$	$\nu = 0.1$

Table 3.3: Parameters of the adaptive combiners algorithms.

can be further improved but at the expense of a degradation on the convergence rate. This is not true for ATC, whose steady-state error cannot be lowered for different parameters.

From Fig. 3.11, we can conclude that D-ATC outperforms both ATC techniques in terms of convergence and steady state, highlighting the behavior of the LS-based algorithm in the fast tracking scenario ($\mathsf{Tr}\{\mathbf{Q}\} = 10^{-4}$).

3.5.4 Performance under node failures

In the previous subsections, we have assumed that all nodes receive the estimates from all neighbors at every iteration. Here, we present a set of experiments allowing for node failure with a fixed probability. As a result of the failure of a node, its estimation is not transmitted to neighbors, and nodes carry out the combination step at every iteration by using the last received estimate from each neighbor.

In Fig. 3.12, we depict the network MSD for different failure rates: (a) 30%, (b) 50%, and (c) 80%, considering a stationary scenario with the settings of Fig. 3.5. In the light of these results, we can conclude that both ATC and D-ATC —with adaptive combiners— remain quite robust to node failures in terms of steady-state performance. However, while D-ATC convergence remains unaffected, ATC convergence deteriorates, being even slower than the non-cooperative strategy.

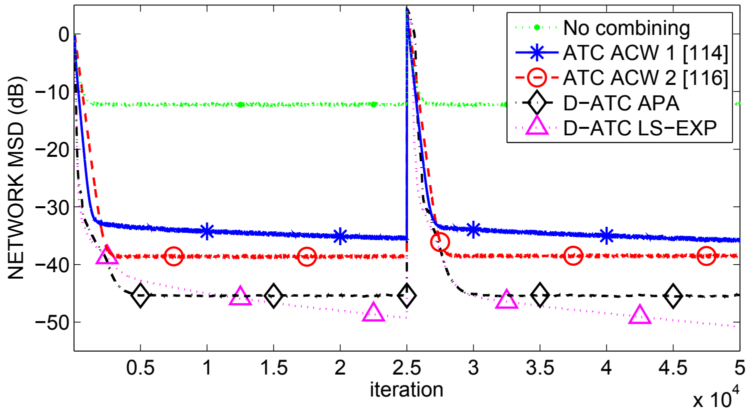
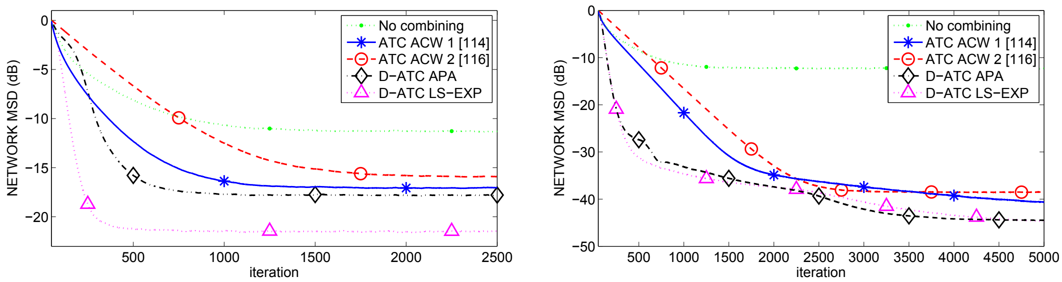


Figure 3.10: Network MSD performance for a stationary estimation problem.



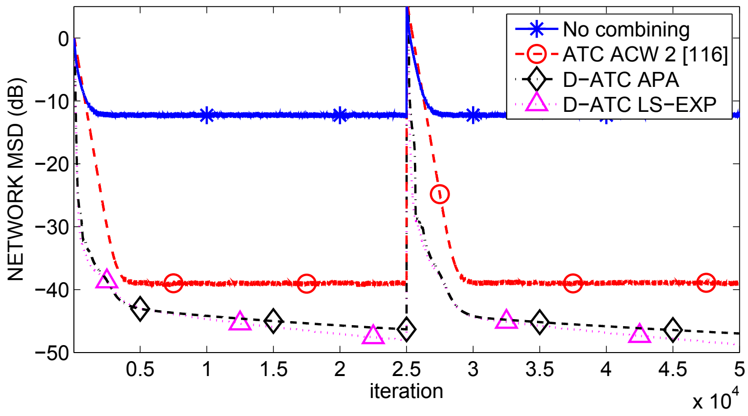
(a)Tracking a fast system, $\text{Tr}\{\mathbf{Q}\} = 10^{-4}$.

(b)Tracking a slow system, $\text{Tr}\{\mathbf{Q}\} = 10^{-8}$.

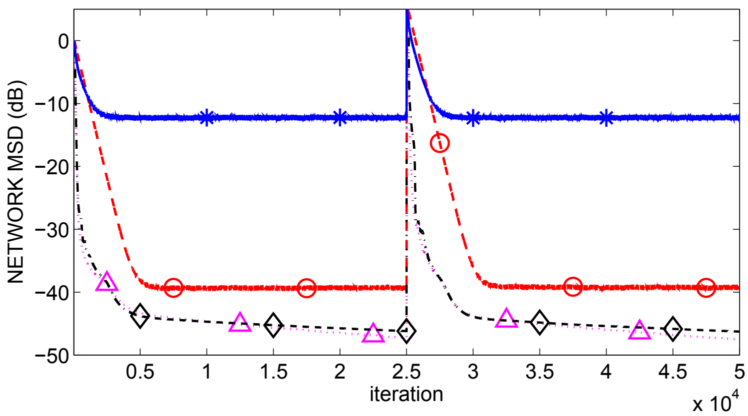
Figure 3.11: Network MSD performance for a tracking problem.

3.6 Summary

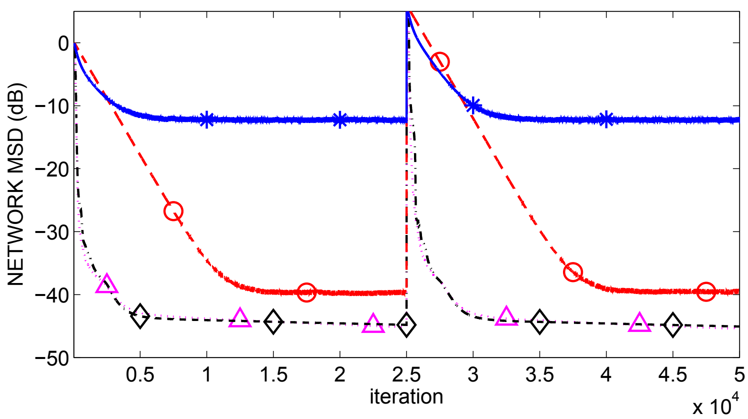
In this chapter, we have proposed a novel diffusion scheme which is specially suitable for heterogeneous networks. In this scheme, adaptation and combination are decoupled and nodes keep a fully local estimation. Although the optimal performances of D-ATC and ATC are similar, the proposed APA and LS rules, in conjunction with the diffusion scheme, seem to be very effective at optimizing network MSD. As a result, in all the presented experiments, the proposed D-ATC diffusion scheme outperforms standard ATC when both schemes use adaptive rules to learn their combiners. Finally, we have showed the robustness of the proposed scheme to communication failures. A more extensive discussion of the presented results, can be found in Chapter 5.



(a) error rate 30%



(b) error rate 50%



(c) error rate 80%

Figure 3.12: Network MSD for the complex network when nodes are subject to random failures at every iteration, with failure rate: (a) 30%, (b) 50%, and (c) 80%.

4

Censoring in adaptive diffusion networks

In the previous chapters, we have presented two complementary contributions. Firstly, we proposed a censoring strategy for energy management in harvesting sensor nodes. Secondly, we introduced a diffusion scheme for adaptive distributed estimation that decoupled the adaptation and combination phases, thus being suitable for networks whose nodes do not share a common clock signal. Therefore, it is interesting to combine these two contributions in a common scenario: a Wireless Sensor Network (WSN) composed of harvesting nodes with finite batteries solving an adaptive distributed estimation problem.

There are few works in the literature of diffusion networks that explicitly take the energy costs into account. A notable exception is [54] where game theory is use to find an activation mechanism in diffusion networks. The algorithm in [54] works in two timescales and an explicit utility of the communications in terms of energy must be defined —which is very difficult, specially in the harvesting case. When using the MDP model of Chapter 2, we can work at only one timescale and no explicit "price" of energy, but just an importance measure, needs to be defined. As far as we know, the only similar scheme in the literature is [8], where they propose a censoring strategy for standard Adapt-then-Combine (ATC) algorithm and non-rechargeable WSNs.

Therefore, in this chapter, we present an energy-aware variation of the D-ATC algorithm for WSNs composed of harvesting nodes. In order to do so, an importance measure suitable for diffusion networks is proposed and a censoring algorithm based on the the MDP methods in Chapter 2 is added on top of the estimation algorithms. The presented preliminary numerical results show the potential of this combined approach.

The rest of the chapter is organized as follows. In Section 4.1, we redefine the signal and energy model taking into account the particularities of the distributed estimation problem. In Section 4.2, the problem of how to assign the importance to estimations transmitted by the nodes is studied, and an importance function is proposed. Then, in Section 4.3, the chosen censoring algorithm is reviewed and the whole Censored Decoupled Adapt-them-Combine (CD-ATC) scheme is summarized. Finally, the chapter is closed with a section of numerical experiments to evaluate the performance of this technique.

4.1 Signal and energy Model

Let's assume a network of N nodes connected in some topology. Then, at each time step n, each node k receives the previous estimations of the neighbor nodes, $\mathbf{w}_k(n-1)$, and access to some local data, $\{d_k(n), \mathbf{u}_k(n)\}$, provided that it has enough battery. With these data, it adapts the local estimation $\boldsymbol{\psi}_k(n)$, computes the importance $x_k(n)$ of the current message, and decides whether it transmits ($a_k(n) = 1$) or censors ($a_k(n) = 0$) it. Then it combines $\boldsymbol{\psi}_k(n)$, using some possibly time-varying combination weights, with the estimations received from its neighboring nodes. It must be taken into account that some of the neighbors could have censored their estimations. These tasks consume energy, which we will model using the following energy costs:

- $b_{0,k}(n)$. Energy consumed by node k in slot n when sensing some new data. It also contains the processing energy for the adaptation step. It is consumed at all time steps n.

- $\Delta_k(n)$. Extra energy consumed by node k in slot n when transmitting an estimation to the neighboring nodes. This transmission is assumed to be a broadcast message, so that this value contains the cost of communication with all the neighbors. This is only consumed if the message is not censored.

In addition, as the node is equipped with a harvesting device, it can harvest some energy from the environment. We will assume that the node harvests some energy $h_k(n)$ in the slot from time n to $n+1$ with probability p_h. A node k with empty battery cannot measure data, adapt its estimation or communicate it with its neighbors. In such cases, as we did in the experiments in Subsection 3.5.4, the neighbors of k assume that the estimation have not changed, i.e., $\mathbf{w}_k(n) = \mathbf{w}_k(n-1)$, in order to adjust their combination weights.

Once defined the signal and the energy models, in the next section we propose an importance measurement for this scenario. Then the whole proposed scheme is summarized in Section 4.3.

4.2 Assignment of importance

As stated in Chapter 2, it is not trivial to decide how to measure the importance of the information shared to the network. Different importance functions have been defined in the literature for related detection and estimation problems in non-rechargeable sensor scenarios. For instance, in the ATC censoring scheme of [8] the product of local combination weight and distance between measurements is proposed as importance. Similarly, the difference between different measurements has been proposed as importance in a tracking scenario based on data aggregation [92]. Finally, we proposed in [46] to use as the importance function for decentralized detection problems the difference of posterior probabilities given the current measurement.

In this thesis, we propose as importance function the decrement of the neighborhood estimation error, defined as

$$x_k(n) = \max\left\{\frac{1}{N_k}\sum_{j\in\mathcal{N}_k} J_j(n) - J_k(n), 0\right\} \tag{4.1}$$

where $J_k(n)$ is a local sample-based estimation of Mean-Squared Error (MSE) that can be computed as

$$J_k(n) = (1 - \alpha_x) \cdot J_k(n-1) + \alpha_x \cdot \check{\xi}_k^2(n-1) \tag{4.2}$$

where $\check{\xi}_k(n) = d_k(n) - \mathbf{w}_k^T(n)\mathbf{u}_k(n)$, and $\alpha_x \in [0, 1]$ is a smoothing constant. This importance can be understood as an approximation to the decrement in the Mean-Squared Error in the neighborhood that the combined estimation of node k, $\mathbf{w}_k(n-1)$, would achieve. When the estimation of node k, $\mathbf{w}_k(n)$, is good, the smoothed squared error $J_k(n)$ will be lower than the average error of the neighbors $\frac{1}{N_k}\sum_{j\in\mathcal{N}_k} J_j(n)$, and this estimation is important and probably should be shared. However, we identify two drawbacks associated to this importance function:

1. Assumption **A2.1** in Section 2.3 is no longer true, as $x_k(n)$ depends not only on $x_k(n-t)$ for $t > 0$, but also on the decisions of other nodes in the network. Part of these correlations could be taken into account augmenting the state to include for example $x_k(n-1)$, but in this chapter we choose to follow a simpler approach.
2. Nodes have to share an additional scalar value, $J_k(n)$. We do not considered this as a problem since nodes can communicate it together with $\mathbf{w}_k(n)$ which could have a large amount of coefficients. In addition, note that some diffusion schemes with adaptive combiners (e.g. [116]) assume a similar increment in communication.

Although this is just an heuristic and further study of assignment of importance is needed, this chapter is proposed as a proof of concept and we follow this simple approach to show the potential of this combined strategy.

4.3 Censoring algorithm

In order to keep things simple, we use as censoring algorithm the Adaptive Balanced Transmitter (ABT) presented in Section 2.5.3. Remind that this scheme is suboptimal when the battery size is finite, but it is a computationally cheap adaptive censoring algorithm. The basis of the algorithm is the computation of a constant threshold

CD-ATC Scheme

INPUTS: Initial battery $e_k(0)$ and η for all k

Initialize $\tau_k(n) = 0$, $n_{0,k} =$, $\bar{b}_{0,k} = 0$, $n_{1,k} = 0$, $\bar{b}_{1,k} = 0$ for all k.

At each time step n, and for each sensor node k:

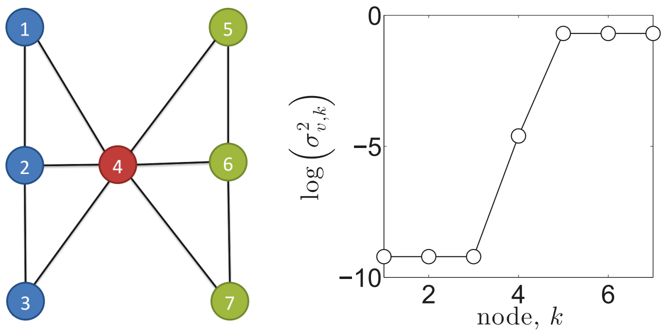
1. Sense $\{d_k, \mathbf{u}_k\}$ and receive estimations of neighbors $\{\mathbf{w}_\ell(n-1)\}_{\ell \in \mathcal{N}_k}$.
2. Harvest energy $h_k(n)$ and consume $\hat{b}_{0,k}(n)$, $(b_{0,k}(n) = \hat{b}_{0,k}(n) - h_k(n))$.
3. Compute importance $x_k(n)$ using (4.1).
4. Decide about transmitting the message:
   $a_k(n) = u\left[x_k(n) - \tau_k(n-1)\right]$.
5. Consume additional cost $\Delta_k(n)$ if $a_k(n) = 1$,
   $b_k(n) = b_{0,k}(n) + a_k(n)\Delta_k(n)$,
   $e(n+1) = \phi_B\left[e(n) - b_k(n)\right]$.
6. Update estimated costs, $\rho_k(n)$ and $\tau_k(n)$ using (4.3).
7. If $e(n+1) > 0$:
   Adapt local estimation $\boldsymbol{\psi}_k(n)$ using (3.8).
   Update combination weights $\boldsymbol{c}_k$ using Least-Squares algorithm (Section 3.4.2).
   Update $J_k(n)$ using (4.2).
   Combine received estimations $\{\mathbf{w}_\ell(n-1)\}_{\ell \in \mathcal{N}_k}$ with $\boldsymbol{\psi}_k(n)$ using $\boldsymbol{c}_k$.
   Transmit combined estimation $\mathbf{w}_k(n)$ and $J_k(n)$ to the neighbors.

Table 4.1: Description of Censoring D-ATC scheme.

that balances the energy consumed and harvested. For convenience of reference, we review here the formula for threshold computation (4.3):

$$\tau_k(n) = \tau_k(n-1) + \eta_{k,n}\big(\rho_{k,n} u\left[x_k(n) - \tau_k(n-1)\right] - (1-\rho_{k,n})u\left[\tau_k(n-1) - x_k(n)\right]\big), \quad (4.3)$$

where subindex k has been included to represent the node index and $\rho_{k,n} = \frac{\bar{b}_{1,k}}{\bar{b}_{1,k} - \bar{b}_{0,k}}$ has to be estimated as in Subsection 2.5.3 in a sample-based manner. Table 4.1 summarizes the diffusion scheme together with the censoring algorithm, named Censoring D-ATC (CD-ATC).



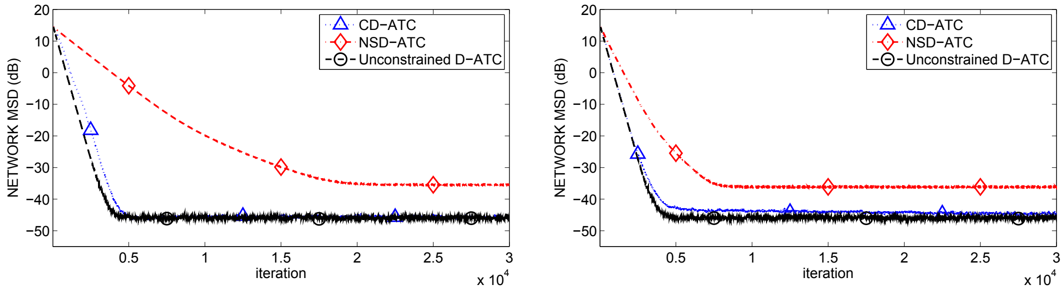
(a) (b)

Figure 4.1: (a) Network topology for the simulation experiments. (b) noise power $\sigma^2_{v,k}$ at each node in the network (in log-scale).

4.4 Simulation Results

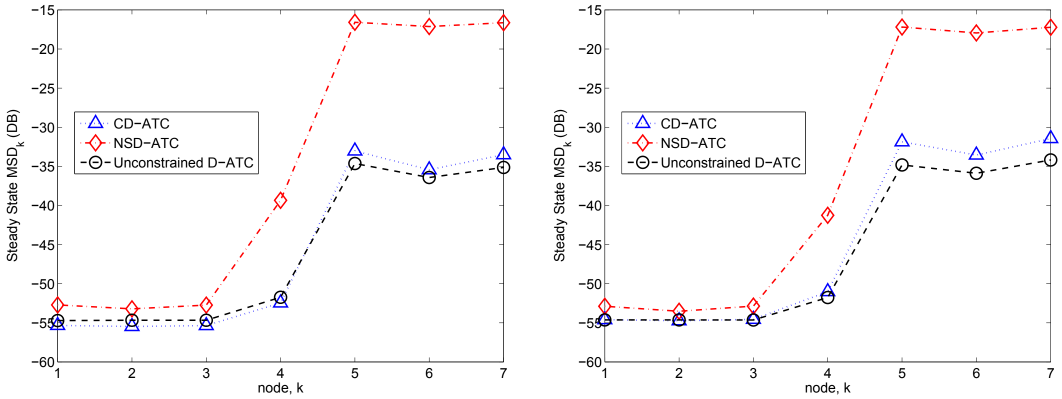
In this section, we will show simulation results to evaluate the potential of using a censoring algorithm in diffusion schemes. In order to do so, we recur to the network topology in Fig 4.1. All the nodes use an NLMS algorithm in the adaptation phase with a common step size, $\mu = 0.1$. The signal power and $\mathbf{w}_o$ are the same as in Section 3.5, the only difference among the nodes being their noise variances, shown in Fig. 4.1.(b). In this topology, we have two subnetworks connected through node 4. Nodes $\{1, 2, 3\}$ are less noisy ($\sigma^2_{v,\{1,2,3\}} = 10^{-4}$), while nodes $\{5, 6, 7\}$ are much noisier ($\sigma^2_{v,\{5,6,7\}} = 0.5$) and their steady-state performance is expected to be worse. Node 4 is a bridge between both subnets and has an intermediate noise variance $\sigma^2_{v,4} = 0.01$. Consequently, node 4 should not be very selective so that the information flows from the left to the right side.

Regarding the energy parameters, all the nodes have the same characteristics: Battery Size $B = 500$, the sensing consumption $b_{0,k}(n) = 1$, the transmission cost $\Delta_k(n) = 2$, and $h_k(n)$ is uniformly distributed in the range $[2, 4]$. Finally the probability of harvesting any energy takes two different values $p_h = \{0.4, 0.8\}$. Fig 4.2 displays the Network MSD while Fig. 4.3 represents the steady-state MSD, $\text{MSD}_k(\infty)$ as defined in (3.10), for the two different harvesting probabilities p_h. The simulated schemes are a non-selective D-ATC (NSD-ATC), the D-ATC scheme without censoring any information and the proposed CD-ATC. Note that in both schemes, whenever



(a) $p_h = 0.4$ (b) $p_h = 0.8$

Figure 4.2: Network MSD performance for two different harvesting scenarios: (a) $p_h = 0.4$ and (b) $p_h = 0.8$.

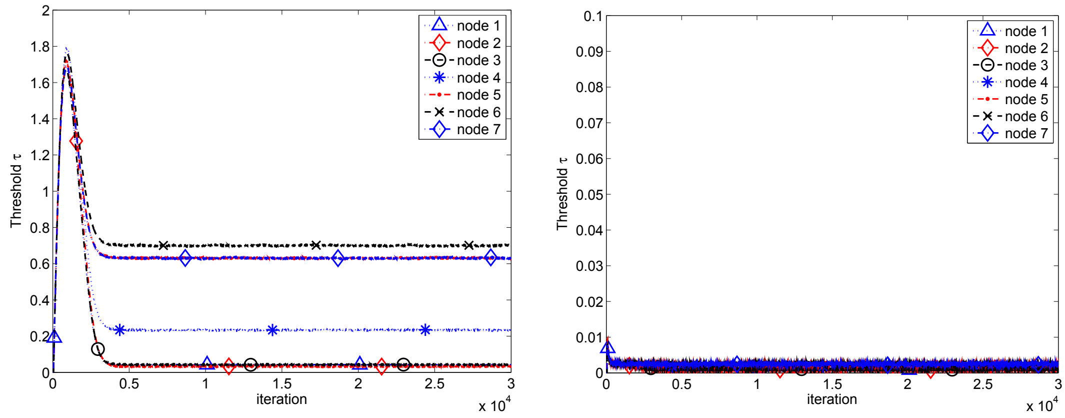


(a) $p_h = 0.4$ (b) $p_h = 0.8$

Figure 4.3: Steady-state MSD for two different harvesting scenarios: (a) $p_h = 0.4$ and (b) $p_h = 0.8$.

the battery of a node is depleted it drops the estimation. In addition we also display, as a baseline, the performance of the standard D-ATC in the unconstrained scenario, i.e., infinite amount of energy refill.

From both figures, we can conclude that censoring provides an obvious gain both in convergence and steady-state performance. As expected the gain is larger when the harvesting probability is lower, because in such case the NSD-ATC battery is zero most of the time. In Fig. 4.3 we can see that the steady-state MSD of the nodes tends to be more similar in CD-ATC than in the non-selective case where the lack of energy degrades the combined estimation.



(a) $p_h = 0.4$ (b) $p_h = 0.8$

Figure 4.4: Censoring threshold evolution for all the nodes in the network in two different harvesting scenarios: (a) $p_h = 0.4$ and (b) $p_h = 0.8$.

In order to understand the behavior of the censoring scheme, we plot in Fig. 4.4 the evolution of the thresholds $\tau_k(n)$. In case (b), $p_h = 0.8$, very little censoring —small thresholds τ_k— is needed to compensate the energy consumption, and improve the network performance. The evolution of $\tau_k(n)$ in case (a) is more interesting, where two phases can be observed. In the transient, the thresholds converge to a similar value —as convergence rate of the nodes does not depend on their noise variance— and the slight difference among them depends just on the node degree. Then, when nodes are about to reach the steady-state regime, all the thresholds quickly converge to values that mostly depend on the noise variance.

4.5 Summary

In conclusion, we have proposed a censoring scheme for diffusion networks with nodes equipped with harvesting devices. The good performance achieved by the combined scheme suggests that a better design of the importance function or a more involved decisions scheme, e.g., random policies, could eventually improve the performance of standard D-ATC even in the unconstrained case. In some way, this connects with the design of sparse combination schemes for diffusion networks, a topic that, as far as we know, has not been deeply studied in the literature.

5

Conclusions and future work

In this chapter, we summarize the main contributions of this thesis and discuss the presented results. Then, we close the dissertation describing some of the future research lines that this thesis opens.

5.1 Summary of contributions

The contributions of this thesis can be grouped into two main lines. In the first one, we focus on proposing censoring schemes for sensor networks composed of harvesting sensors. In the second one, we explore new strategies of distributed estimation based on diffusion schemes. Finally, both approaches are assembled in a common setup.

Regarding the first part of the thesis, the main contribution is the design, analysis and evaluation of censoring schemes for Wireless Sensor Networks (WSNs) with harvesting devices. Going into further detail:

- We model the problem using an infinite-horizon Markov Decision Process (MDP) that optimizes the expected aggregate reward.

- Under some convenient assumptions, mainly stationarity, we obtain a threshold-based optimal policy. Although this solution is useful, the policy itself is difficult to solve. As a result, we analyze the optimal policy in a simplified scenario where we are able to compute it using Value Iteration.

- We approximate the optimal policy using Stochastic Approximation techniques to obtain an implementable algorithm, Stochastic Approximate Policy (SAP). This strategy is compared with a simpler algorithm, Adaptive Balanced Transmitter (ABT), similar to other approaches in the literature and based on the computation of an energy-independent threshold.

- The proposed method (SAP) is analyzed in a large number of scenarios including scenarios where some of the assumptions under which it is designed (i.e stationarity, single-hop networks) do not hold. It is also compared with state-of-the-art approaches based on Q-learning, showing in all cases a significant performance improvement.

In the second part of the thesis, we introduce a novel strategy for distributed estimation in WSN, Decoupled Adapt-then-Combine (D-ATC), based in the popular diffusion scheme for adaptive networks. The main contributions on this research line are:

- We propose a novel diffusion scheme where the adaptation and combination steps are decoupled. In this structure nodes keep an estimation, which is only adapted from local data, and combine it with the combined estimations sent by nodes in its neighborhood.

- We compare the new strategy with the standard ATC scheme, showing that both schemes can perform similarly if suitable combiners are chosen.

- We theoretically analyze D-ATC using a variety of analysis tools: including classical analysis approach based on the computation of cross-covariance matrices, and the more recent energy conservation approach. This is, as far as we know, the only work where an adaptive network is studied under these two analysis approaches. In addition these analyses are valid both for stationary and tracking scenarios, which are not usually considered in the analyses of adaptive networks.

- Due to the need of time-varying combiners, we present two adaptive rules to learn the combination weights for our D-ATC scheme. One of these rules is based in a least-squares (LS) approach, and the other in affine projection algorithm (APA).

- The performance of the diffusion strategy and the adaptive combination rules are evaluated through numerical simulations, showing a significant gain with respect to other state-of-the-art approaches. Besides, the proposed scheme shows additional robustness with respect to previous approaches when we consider failures or asynchrony in the nodes operation.

Finally, based in the good behavior of D-ATC under asynchronous communications, the previous approaches are put together in a common setup. In order to apply the censoring scheme in the diffusion setup we have to choose a suitable importance value for the messages (combined estimates) generated by the nodes. As a result, we propose a sensible importance function based on smoothed local and neighboring squared error. Some preliminary simulations are performed showing the potential of censoring schemes in diffusion networks when energy is a limitation.

5.2 Discussion

Our first contribution is a censoring scheme for energy harvesting WSNs. We have used an infinite-horizon MDP and proposed a model-based stochastic approximation scheme, SAP, which works better than current standard strategies, such as the balanced scheme or Q-learning, in relevant scenarios. The main limitation of this approach is that it is model-based, and when the assumptions of the model do not hold the obtained policies will be suboptimal. This means that we do not have any guarantee of optimality in scenarios with characteristics such as time-correlated energy refill, correlated data importance, multi-hop networks, etc. Although simulations show that the proposed algorithm works well in some of these scenarios, an approach that explicitly models this phenomena is expected to work better.

Regarding the diffusion scheme, we have proposed D-ATC as an alternative to standard diffusion schemes, such as ATC. Our scheme is specially suitable for heterogeneous networks and networks with asynchronous communications. However, there are some issues that have not been solved in the present work. First of all, an explicit consideration of the time-varying combination coefficients in the energy conservation analysis could help to design new adaptive algorithms to learn the coefficients, similarly to [116, 140]. In addition, incorporating the asynchronous communications or the dynamic topology in the design and analysis of the diffusion strategy could also provide intuitions that improve the performance of the algorithm in those situations. Finally, the ATC algorithm has been analyzed in a number of problems different to distributed estimation, e.g., multi-task learning, dictionary learning, etc. In this work, we focus on the linear estimation problem and, consequently, do not provide any guarantee of performance for those cases.

Finally, in Chapter 4 we have presented some preliminary results to show the potentials of incorporating a censoring algorithm into a diffusion scheme. Although the presented numerical simulations serve as a justification for this approach, we know that the proposed solution has a number of limitations. The censoring scheme that we used, ABT, can perform poorly in some scenarios as shown in 2.6. Moreover, although we believe that the chosen importance function is a sensible approach, it is just a heuristic. We would expect a larger gain from a more formal approach where both the diffusion strategy and the censoring scheme are optimized together.

5.3 Future work

A number of future research lines can be extracted from the discussion above. We provide below a list of some of them grouped by the different contributions of the thesis. Firstly, regarding the paths suggested by the censoring scheme, we identify the following:

- **Data correlation.** In Section 2.3, we assume i.i.d. importance values. As we saw in Chapter 4, this could be not true in some meaningful scenarios.

Proposing models that account for these dependencies, for example augmenting the state with more variables, could improve the performance of the censoring schemes but would increase their complexity. Consequently, designing low-complexity censoring algorithms that take data correlation into account constitutes an interesting and challenging research path.

- **Energy Correlation.** A similar extension related to energy time-dependencies could also be explored. One can think of several scenarios where the energy refill has some seasonal component, for example solar energy. In such cases, more complex energy models, e.g., Markov models, etc. [60, 84], could be proposed; but the complexity would also increase significantly. Finding strategies that reduce this complexity is also an appealing research direction.

- **Multi-hop networks.** The presented algorithms are designed for single-hop networks. However they have also shown good performance in networks with multi-hop topologies. Therefore, it seems reasonable to try search for optimal censoring schemes at a network level. We have already proposed a censoring scheme that is close to the optimal for non-rechargeable multi-hop networks [43]. However, even in the non-rechargeable case, its implementation is difficult and the extension to the harvesting case (where the optimal policy is no longer constant) is non-trivial. Therefore, significant research could be done in this direction.

- **Incomplete state observation.** A different approach to the previous problems is assuming the the actual state of the node —or the network in the multi-hop case— is not observable, i.e., we assume that only a part of the state or a related variable is observed. This kind of approach is also valid for example to model more complex battery models. In these cases, the state-observation model is a Hidden Markov Model (HMM) [98] and the MDP becomes a Partially Observable Markov Decision Processes (PO-MDPs) [128]. This approach is interesting because the optimization in this kind of models would produce more sophisticated algorithms that could keep the complexity under control.

- **Alternative stochastic schemes.** Regarding the approximated schemes, a possible approach is to develop more complex Reinforcement Learning algorithms [113]. As the state is continuous and the optimal policy has some identifiable shape —decreasing and smooth—, algorithms of reinforcement learning with function approximation [128] seem to be a sensible approach. In those methods, the value function is modeled using a parametric function and reinforcement learning schemes could be used to learn those parameters.

- **Explore other WSNs tasks.** Finally, in this work we have proposed a reasonable importance function for a particular WSN task: distributed estimation. The applicability of this scheme to other tasks demands the definition of importance functions that represent the relevance of the communications on that setup. In that sense, in [46] we presented some preliminary work on the detection problem and defined a importance function for single-hop detection networks. Since the assignment of the importance value is not a trivial design decision, future work should include a deep analysis of each problem to be able to apply the presented schemes in more scenarios.

If we focus now in the diffusion scheme, this work also opens a number of research lines worth exploring.

- **Theory for asynchronous adaptation.** Strict synchronization is an unfeasible constraint in WSNs. As a consequence, we consider the robustness of our scheme to this lack of synchrony a very valuable feature. However, it would be really interesting to understand this behavior from a theoretical point of view. In the case of standard ATC some results [139] have been published, and a similar approach could be followed for our scheme.

- **New adaptive combination rules.** Studying novel combination rules can be useful to further improve the performance of D-ATC. For example, the simulations in Section 4.4 may suggest that incorporating some sparsity in the combination coefficients could be beneficial.

- **Bayesian interpretation and combination.** There is a growing interest in bringing ideas from *Bayesian* or *probabilistic machine learning* to adaptive filtering problems [48, 77, 123]. As the Bayesian modeling provides not only a point estimate but a full probability distribution —and consequently an uncertainty measurement—, those interpretations could help for example in the determination of the combination weights in adaptive networks.

- **Generalize to more complex distributed processing tasks.** In principle, the extension of our strategy to more general distributed convex optimization problems is straightforward. However, we do not have guarantee about its performance anymore, and the conclusions of this thesis cannot be directly applied. The study of the performance of D-ATC in these cases or more complex ones, e.g., non-convex optimization, optimization with different minimizers in each nodes, multi-task learning, etc, is a very interesting research path.

- **Other areas.** Here we have presented diffusion networks in the context of WSNs but this is not at all their only potential area of application. In that sense, the good features of D-ATC may make it suitable for applications such as social networks, smart grid, cognitive networks, etc. Research about that specific problems is needed in other to apply our strategy to them.

Finally, there are some research paths that are common for all the thesis:

- **Censoring diffusion networks.** We have proposed in this thesis a first approach of this problem but further study is needed. The main open question is how much we can improve the performance of the network by using censoring schemes. In order to do that, the effect of importance assignment, the use of different censoring algorithms, and the effect of different combination schemes has to be studied.

- **Real-world tasks.** In this thesis we focus on algorithms more that applications but it is also important to evaluate these schemes in the resolution of real tasks. We hope that our contribution helps to further develop the applicability of these networks.

- **Implementation in real sensors.** In this thesis, we have focused on theoretically analyzing and developing schemes and algorithms that take into account some real limitations of WSNs. Nevertheless, the implementation of the algorithms in real WSNs is a challenging task by itself and we have to consider it as future work. In addition, the problems derived from the real implementation are also a useful guide for future theoretical work that should not be underestimated.

A

Proofs and Derivations of Chapter 2

A.1 Proof of Theorem 1

Using (2.3), and for $\mathbf{s} = (e, x)$,

$$\mathbb{E}\{r(n)|a(n) = a, \mathbf{s}(n) = \mathbf{s}\} = axW(e, x) \tag{A.1}$$

where $W(e, x) = \mathbb{E}\{w(n)|e(n) = e, x(n) = x\}$.

Also, using (2.2) and taking into account that $x(n)$ is an i.i.d. sequence and independent of $e(n)$ (A2.1), and $b(n)$ is independent of $x(n)$ and $e(n)$ given $a(n)$ (A2.2), we have that

$$\begin{aligned}
&\mathbb{E}\{V_{\pi^*}(\mathbf{s}(n+1))|a(n) = a, e(n) = e, x(n) = x\} \\
&\quad =\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e - b(n)), x(n+1))||a(n) = 1, x(n) = x\} \\
&\quad =a\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e - b(n)), x(n+1))|a(n) = 1\} \\
&\qquad + (1-a)\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e - b(n)), x(n+1))|a(n) = 0\}.
\end{aligned} \tag{A.2}$$

Joining (2.6), (A.1) and (A.2), and using as3),

$$
\begin{aligned}
V_{\pi^*}(\mathbf{s}) = \max_a \{ & axW(e,x) \\
& + a\gamma\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e-b(n)), x(n+1))|a(n)=1\} \\
& + (1-a)\gamma\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e-b(n)), x(n+1))|a(n)=0\}\}.
\end{aligned} \tag{A.3}
$$

Defining the threshold function

$$
\begin{aligned}
\tau(e) = \gamma(& \mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e-b(n)), x(n+1))|a(n)=0\} \\
& - \mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e-b(n)), x(n+1))|a(n)=1\}).
\end{aligned} \tag{A.4}
$$

the optimal policy is $a^* = \pi^*(e,x) = u(xW(e,x) - \tau(e))$, which is equivalent to (2.7)

Note also that (A.4) can be written as

$$
V_{\pi^*}(\mathbf{s}) = \gamma\mathbb{E}\{V_{\pi^*}(\boldsymbol{\phi}_B(e-b(n)), x(n+1))|a(n)=0\} + [xW(e,x) - \tau(e)]^+ . \tag{A.5}
$$

Defining $\lambda(e)$ as in (2.10), we get (2.8) and (2.9).

A.2 Derivation of Transition probability matrix

The entries of the transition matrix $\mathbf{P}$ for an arbitrary transmission policy based on a generic threshold function $\tau(e)$ [cf. (2.23)] can be found (using the abbreviated notation $P\{j|i,\ldots\}$ instead of $P\{e(n)=j|e(n-1)=i,\ldots\}$) as

$$
\begin{aligned}
p_{ij} = & P\left\{ j|i, x(n-1) \geq \frac{\tau(i)}{W(i)} \right\} \cdot \left(1 - F_X\left(\frac{\tau(i)}{W(i)}\right)\right) \\
& + P\left\{ j|i, x(n-1) < \frac{\tau(i)}{W(i)} \right\} \cdot F_X\left(\frac{\tau(i)}{W(i)}\right) \\
= & \left[\begin{array}{ll}
(1-F_{b_1}(i))(1-F_X) + (1-F_{b_0}(i))F_X & j=0 \\
P_{b_1}(i-j)(1-F_X) + P_{b_0}(i-j)F_X & 0<j<B \\
F_{b_1}(i-B)(1-F_X) + F_{b_0}(i-B)F_X & j=B
\end{array} \right.
\end{aligned} \tag{A.6}
$$

where P_{b_0} and P_{b_1} are the conditional probability mass functions of b given actions $a=0$ and $a=1$, respectively, and F_{b_0} and F_{b_1} the respective cumulative conditional probability functions. With some abuse of notation, we have abbreviated $F_X = F_X(\tau(i)/W(i))$.

A.3 Derivation of the stochastic algorithm

Let us first define functions

$$\alpha(e) = \mathbb{E}\{\lambda(\boldsymbol{\phi}_B(e - b_0(n)))\} \tag{A.7}$$

$$\beta(e) = \mathbb{E}\{\lambda(\boldsymbol{\phi}_B(e - b_0(n) - \Delta(n)))\}. \tag{A.8}$$

The solution (2.7) of the MDP can then be written as

$$a(n) = u\left[W(e(n))x(n) - \gamma\left(\alpha(e(n)) - \beta(e(n))\right)\right], \tag{A.9}$$

$$\lambda(e) = \gamma\alpha(e) + \mathbb{E}\{(W(e, x(n))x(n) - \gamma(\alpha(e) - \beta(e)))^{+}\}. \tag{A.10}$$

To derive the proposed algorithm as an instance of the Robbins-Monro algorithm [134], we represent functions in vector notation.

Accordingly, we define $\boldsymbol{\lambda} = (\lambda(0), \lambda(1), \ldots, \lambda(B))^{\intercal}$, and let $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ be the corresponding vectorizations of $W(e)$, $\alpha(e)$ and $\beta(e)$. We also define the vector of success indices

$$\mathbf{w}_c = (u(0-b), u(1-b), \ldots, u(B-b))^{\intercal} \tag{A.11}$$

and the transformation $\boldsymbol{\lambda}' = \mathbf{T}_b\boldsymbol{\lambda}$, such that $\lambda'_i = \lambda_{\boldsymbol{\phi}_B(i-b)+1}$. Then, we can write

$$\boldsymbol{\omega} = \mathbb{E}\{\mathbf{w}_{b_0(n)+\Delta(n)}\} \tag{A.12}$$

$$\boldsymbol{\alpha} = \mathbb{E}\{\mathbf{T}_{b_0(n)}\boldsymbol{\lambda}\} \tag{A.13}$$

$$\boldsymbol{\beta} = \mathbb{E}\{\mathbf{T}_{b_0(n)+\Delta(n)}\boldsymbol{\lambda}\} \tag{A.14}$$

$$\boldsymbol{\lambda} = \gamma\boldsymbol{\alpha} + \mathbb{E}\{(\boldsymbol{\omega}x - \gamma(\boldsymbol{\alpha} - \boldsymbol{\beta}))^{+}\}. \tag{A.15}$$

Now, defining vector

$$\mathbf{v} = (\boldsymbol{\omega}^{\intercal}, \boldsymbol{\alpha}^{\intercal}, \boldsymbol{\beta}^{\intercal}, \boldsymbol{\lambda}^{\intercal})^{\intercal} \tag{A.16}$$

and matrices

$$\mathbf{M}_{b_0,b_1} = \begin{pmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{T}_{b_0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{T}_{b_1} \\ \mathbf{0} & \gamma\mathbf{I} & \mathbf{0} & -\mathbf{I} \end{pmatrix} \tag{A.17}$$

$$\mathbf{N}_x = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ x\mathbf{I} & -\gamma\mathbf{I} & \gamma\mathbf{I} & \mathbf{0} \end{pmatrix}, \tag{A.18}$$

eqs. (A.12) to (A.15) are equivalent to

$$\mathbb{E}\left\{(\mathbf{N}_x\mathbf{v})^+ + \mathbf{M}_{b_0(n),b_0(n)+\Delta(n)}\mathbf{v} + \overline{\mathbf{w}}_{b_0(n)+\Delta(n)}\right\} = 0 \tag{A.19}$$

where $\overline{\mathbf{w}}_b = (\mathbf{w}_b^\intercal, \mathbf{0}^\intercal, \mathbf{0}^\intercal, \mathbf{0}^\intercal)^\intercal$. The Robbins-Monro algorithm that solves (A.19) then becomes [134]

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \eta(n)\left[(\mathbf{N}_x(n)\mathbf{v}(n))^+ + \mathbf{M}_{b_0,b_1}(n)\mathbf{v}(n) + \overline{\mathbf{w}}_b(n)\right], \tag{A.20}$$

which is equivalent to (2.32)-(2.35).

B

Proofs and Derivations of Chapter 3

Recurrent expressions for the cross-variance matrices

In this appendix, we obtain recurrent expressions for $\mathbf{S}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_{\ell}(n)\widetilde{\boldsymbol{\psi}}_{m}^{T}(n)\}$, $\mathbf{X}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_{\ell}(n)\widetilde{\mathbf{w}}_{m}^{T}(n)\}$, and $\mathbf{W}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\mathbf{w}}_{\ell}(n)\widetilde{\mathbf{w}}_{m}^{T}(n)\}$, assuming $\ell \neq m$ and $\ell = m$, where ℓ and m represent nodes of the network.

In order to obtain a recursion for $\mathbf{S}_{\ell m}(n)$, we must rewrite (3.4) in terms of the weight-error vector $\widetilde{\boldsymbol{\psi}}_{k}(n)$. Thus, subtracting both sides of (3.4) from $\mathbf{w}_{\mathrm{o}}(n)$ and replacing $\xi_k(n) = \mathbf{u}_k^T(n)\widetilde{\boldsymbol{\psi}}_k(n-1) + v_k(n)$, we obtain

$$\widetilde{\boldsymbol{\psi}}_k(n) = [\mathbf{I} - \widetilde{\mu}_k(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)]\,\widetilde{\boldsymbol{\psi}}_k(n-1) - \widetilde{\mu}_k(n)\mathbf{u}_k(n)v_k(n) + \mathbf{q}(n), \tag{B.1}$$

where $\mathbf{I}$ stands for the identity matrix of dimension M and

$$\widetilde{\mu}_k(n) \triangleq \frac{\mu_k}{\delta + \|\mathbf{u}_k(n)\|^2}. \tag{B.2}$$

Multiplying (B.1) with $k \leftarrow \ell$ by its transpose with $k \leftarrow m$, taking the expectations of both sides, and using the fact that $\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_k(n-1)\mathbf{q}^T(n)\} = \mathbf{0}$ since the sequence $\{\mathbf{q}(n)\}$

is i.i.d. (Assumption **A3.1**), we obtain

$$\begin{aligned}
&\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n)\widetilde{\boldsymbol{\psi}}_m^T(n)\} \approx \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\}\\
&-\overbrace{\mathbb{E}\{\widetilde{\mu}_m(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{A}}}\\
&-\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell^T(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\}}^{\boldsymbol{\mathcal{B}}}\\
&+\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)\widetilde{\mu}_m(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell^T(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{C}}}\\
&+\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)\widetilde{\mu}_m v_\ell(n)v_m(n)\mathbf{u}_\ell(n)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{D}}}\\
&-\overbrace{\mathbb{E}\{\widetilde{\mu}_m(n)v_m(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{E}}}\\
&-\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)v_\ell(n)\mathbf{u}_\ell(n)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\}}^{\boldsymbol{\mathcal{F}}}\\
&+\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)\widetilde{\mu}_m v_m(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell^T(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{G}}}\\
&+\overbrace{\mathbb{E}\{\widetilde{\mu}_\ell(n)\widetilde{\mu}_m v_\ell(n)\mathbf{u}_\ell(n)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\}}^{\boldsymbol{\mathcal{H}}}\\
&+\mathbb{E}\{\mathbf{q}(n)\mathbf{q}^T(n)\}. \qquad \text{(B.3)}
\end{aligned}$$

Now, using Assumptions **A3.2**, **A3.5** and **A3.6** (see below), we can evaluate the terms $\boldsymbol{\mathcal{A}}$-$\boldsymbol{\mathcal{H}}$ of (B.3):

$\boldsymbol{\mathcal{A}}$- Recalling that Assumption **A3.2** implies that $\widetilde{\boldsymbol{\psi}}_\ell(n-1)$ and $\widetilde{\boldsymbol{\psi}}_m(n-1)$ are independent of $\mathbf{u}_m(n)$, the term $\boldsymbol{\mathcal{A}}$ can be approximated by

$$\begin{aligned}
\boldsymbol{\mathcal{A}} &= \mathbb{E}\Big\{\mathbb{E}\Big\{\widetilde{\mu}_m(n)\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\mathbf{u}_m(n)\mathbf{u}_m^T(n)|\mathbf{u}_m(n)\Big\}\Big\}\\
&\approx \mathbb{E}\left\{\widetilde{\mu}_m(n)\mathbb{E}\left\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\right\}\mathbf{u}_m(n)\mathbf{u}_m^T(n)\right\}\\
&= \mathbf{S}_{\ell m}(n-1)\,\mathbb{E}\{\widetilde{\mu}_m(n)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\}. \qquad \text{(B.4)}
\end{aligned}$$

We must obtain an approximation for

$$\mathbb{E}\{\widetilde{\mu}_m(n)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\} = \mu_m\mathbb{E}\left\{\frac{\mathbf{u}_m(n)\mathbf{u}_m^T(n)}{\delta+\mathbf{u}_m^T(n)\mathbf{u}_m(n)}\right\}. \qquad \text{(B.5)}$$

To arrive at a simple model, we also assume that

A3.5- The number of coefficients M is large enough for each element $\mathbf{u}_m(n)\mathbf{u}_m^T(n)$ in the numerator to be approximately independent from the denominator $\sum_{l=0}^{M-1}|u(n-l)|^2$. This is equivalent to applying the averaging principle of [104], since for large M, $\|\mathbf{u}_m(n)\|^2$ tends to vary slowly compared to the individual entries of $\mathbf{u}_m(n)\mathbf{u}_m^T(n)$.

A3.6- The regressors $\mathbf{u}_k(n)$, $k = 1, 2, \ldots, N$ are formed by a tapped-delay line with Gaussian entries and $\delta = 0$. This is a common assumption in the analysis of adaptive filters and leads to reasonable analytical results [57].

Under **A3.5** and **A3.6**, (B.5) can be approximated as [31, 14]

$$\mathbb{E}\{\widetilde{\mu}_m(n)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\} \approx \frac{\mu_m}{\sigma_{u_m}^2(M-2)}\mathbf{R}_{mm}, \tag{B.6}$$

and the term $\boldsymbol{\mathcal{A}}$ as

$$\boldsymbol{\mathcal{A}} \approx \frac{\mu_m}{\sigma_{u_m}^2(M-2)}\mathbf{S}_{\ell m}(n-1)\mathbf{R}_{mm}, \tag{B.7}$$

where $\sigma_{u_m}^2$ is the variance of the input signal at node k.

$\boldsymbol{\mathcal{B}}$- Analogously, we obtain for $\boldsymbol{\mathcal{B}}$

$$\boldsymbol{\mathcal{B}} \approx \frac{\mu_\ell}{\sigma_{u_\ell}^2(M-2)}\mathbf{R}_{\ell\ell}\,\mathbf{S}_{\ell m}(n-1). \tag{B.8}$$

$\boldsymbol{\mathcal{C}}$- Under **A3.2**, it holds that

$$\boldsymbol{\mathcal{C}} = \mathbb{E}\left\{\widetilde{\mu}_\ell(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell^T(n)\mathbf{S}_{\ell m}(n-1)\widetilde{\mu}_m(n)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\right\}. \tag{B.9}$$

Note that if the regression data of nodes ℓ and m are spatially independent (Assumption **A3.2**) and under **A3.5** and **A3.6**, (B.9) reduces to

$$\boldsymbol{\mathcal{C}} \approx \frac{\mu_\ell\mu_m}{\sigma_{u_\ell}^2\sigma_{u_m}^2(M-2)^2}\,\mathbf{R}_{\ell\ell}\mathbf{S}_{\ell m}(n-1)\mathbf{R}_{mm}. \tag{B.10}$$

For $m = \ell$, $\mathbf{R}_{mm} = \mathbf{R}_{\ell\ell} \neq \mathbf{0}$ and therefore, (B.9) reduces to (see, e.g., [31, 14])

$$\boldsymbol{\mathcal{C}} \approx \frac{\mu_\ell^2}{\sigma_{u_\ell}^4(M-2)(M-4)}\Big[2\mathbf{R}_{\ell\ell}\mathbf{S}_{\ell\ell}(n-1)\mathbf{R}_{\ell\ell} + \mathrm{Tr}(\mathbf{R}_{\ell\ell}\mathbf{S}_{\ell\ell}(n-1))\mathbf{R}_{\ell\ell}\Big]. \tag{B.11}$$

$\boldsymbol{\mathcal{D}}$- Under Assumption **A3.2** and for $\ell \neq m$, $\boldsymbol{\mathcal{D}} \approx \mathbf{0}$. On the other hand, for $m = \ell$, we get [106]

$$\boldsymbol{\mathcal{D}} \approx \frac{\mu_\ell^2}{\sigma_{u_\ell}^4(M-2)(M-4)}\sigma_{v_\ell}^2\mathbf{R}_{\ell\ell}. \tag{B.12}$$

$\boldsymbol{\mathcal{E}}$-$\boldsymbol{\mathcal{H}}$- Under Assumption **A3.2**, all the terms $\boldsymbol{\mathcal{E}}$ to $\boldsymbol{\mathcal{H}}$ are $M \times M$ null matrices.

From the previous results, (B.3) reduces to (B.13). Similarly, for $m = \ell$, we arrive at (B.14).

$$\begin{aligned}\mathbf{S}_{\ell m}(n) \approx\ & \mathbf{S}_{\ell m}(n-1) - \overline{\mu}_m \mathbf{S}_{\ell m}(n-1)\,\mathbf{R}_{mm} - \overline{\mu}_\ell \mathbf{R}_{\ell\ell}\,\mathbf{S}_{\ell m}(n-1) \\ & + \overline{\mu}_\ell \overline{\mu}_m \mathbf{R}_{\ell\ell}\mathbf{S}_{\ell m}(n-1)\mathbf{R}_{mm} + \mathbf{Q},\ (\ell \neq m) \qquad \text{(B.13)}\\ \mathbf{S}_{\ell\ell}(n) \approx\ & \mathbf{S}_{\ell\ell}(n-1) - \overline{\mu}_\ell \Big[\mathbf{S}_{\ell\ell}(n-1)\,\mathbf{R}_{\ell\ell} + \mathbf{R}_{\ell\ell}\,\mathbf{S}_{\ell\ell}(n-1)\Big] \\ & + \overline{\mu}_\ell^2 \frac{M-2}{M-4}\Big[2\mathbf{R}_{\ell\ell}\mathbf{S}_{\ell\ell}(n-1)\mathbf{R}_{\ell\ell} + \mathbf{R}_{\ell\ell}\mathrm{Tr}(\mathbf{S}_{\ell\ell}(n-1)\mathbf{R}_{\ell\ell})\Big] \\ & + \overline{\mu}_\ell^2 \frac{M-2}{M-4}\sigma_\ell^2 \mathbf{R}_{\ell\ell} + \mathbf{Q}, \qquad \text{(B.14)}\end{aligned}$$

In order to obtain a recurrent expression for $\mathbf{X}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\}$, we first add $\mathbf{q}(n)$ to both sides of (3.16) with $k \leftarrow m$ and transpose the resulting equation, which leads to

$$\widetilde{\mathbf{w}}_m^T(n) \approx c_{mm}(n)\widetilde{\boldsymbol{\psi}}_m^T(n-1) + \sum_{p\in\bar{\mathcal{N}}_m} c_{pm}(n)\widetilde{\mathbf{w}}_p^T(n-1) + \mathbf{q}^T(n). \qquad \text{(B.15)}$$

Then, we multiply (B.15) by $\widetilde{\boldsymbol{\psi}}_\ell(n)$ from the left, using (B.1) with $k \leftarrow \ell$ to multiply the right-hand side. Taking expectations on both sides and using assumptions **A3.1-A3.3**, we arrive at

$$\begin{aligned}\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\} \approx\ & \mathbb{E}\{c_{mm}(n)\}\mathbb{E}\{[\mathbf{I}-\widetilde{\mu}_\ell(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell(n)]\}\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\} \\ & + \sum_{p\in\bar{\mathcal{N}}_m}\mathbb{E}\{c_{pm}(n)\}\mathbb{E}\{[\mathbf{I}-\widetilde{\mu}_\ell(n)\mathbf{u}_\ell(n)\mathbf{u}_\ell(n)]\}\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\mathbf{w}}_p^T(n-1)\}+\mathbf{Q}. \qquad \text{(B.16)}\end{aligned}$$

Under Assumptions **A3.5** and **A3.6** [see Eq. (B.6)], (B.16) reduces to

$$\begin{aligned}\mathbf{X}_{\ell m}(n) \approx\ & \mathbb{E}\{c_{mm}(n)\}\left[\mathbf{I} - \frac{\mu_\ell}{\sigma_{u_\ell}^2(M-2)}\mathbf{R}_{\ell\ell}\right]\mathbf{S}_{\ell m}(n-1) \\ & + \left[\mathbf{I} - \frac{\mu_\ell}{\sigma_{u_\ell}^2(M-2)}\mathbf{R}_{\ell\ell}\right]\sum_{p\in\bar{\mathcal{N}}_m}\mathbb{E}\{c_{pm}(n)\}\mathbf{X}_{\ell p}(n-1) + \mathbf{Q}. \qquad \text{(B.17)}\end{aligned}$$

Using the definition of $\overline{\mu}_\ell$ [Eq. (B.21)] in (B.17), we arrive at (B.18).

$$\begin{aligned}\mathbf{X}_{\ell m}(n) \approx\ & \mathbb{E}\{c_{mm}(n)\}[\mathbf{I} - \bar{\mu}_\ell \mathbf{R}_{\ell\ell}]\mathbf{S}_{\ell m}(n-1) \\ & + [\mathbf{I} - \bar{\mu}_\ell \mathbf{R}_{\ell\ell}] \sum_{p \in \bar{\mathcal{N}}_m} \mathbb{E}\{c_{pm}(n)\}\mathbf{X}_{\ell p}(n-1) + \mathbf{Q}.\end{aligned} \tag{B.18}$$

Finally, to obtain a recurrent expression for $\mathbf{W}_{\ell m}(n) \triangleq \mathbb{E}\{\widetilde{\mathbf{w}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\}$, we multiply (3.16) with $k \leftarrow \ell$ by its transpose with $k \leftarrow m$ from the right and take the expectations of both sides. After some algebraic manipulations under Assumptions **A3.1** and **A3.3**, we arrive at

$$\begin{aligned}\mathbb{E}\{\widetilde{\mathbf{w}}_\ell(n)\widetilde{\mathbf{w}}_m^T(n)\} - \mathbf{Q} \approx\ & \mathbb{E}\{c_{\ell\ell}(n)c_{mm}(n)\}\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\} \\ & + \sum_{p \in \bar{\mathcal{N}}_\ell} \sum_{r \in \bar{\mathcal{N}}_m} \mathbb{E}\{c_{p\ell}(n)c_{rm}(n)\}\mathbb{E}\{\widetilde{\mathbf{w}}_p(n-1)\widetilde{\mathbf{w}}_r^T(n-1)\} \\ & + \sum_{r \in \bar{\mathcal{N}}_m} \mathbb{E}\{c_{\ell\ell}(n)c_{rm}(n)\}\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_\ell(n-1)\widetilde{\mathbf{w}}_r^T(n-1)\} \\ & + \sum_{p \in \bar{\mathcal{N}}_\ell} \mathbb{E}\{c_{p\ell}(n)c_{mm}(n)\}\mathbb{E}\{\widetilde{\mathbf{w}}_p(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\}.\end{aligned} \tag{B.19}$$

Noting that $\mathbb{E}\{\widetilde{\mathbf{w}}_p(n-1)\widetilde{\boldsymbol{\psi}}_m^T(n-1)\} = \mathbf{X}_{mp}^T(n-1)$, (B.19) can be rewritten as (B.20).

$$\begin{aligned}\mathbf{W}_{\ell m}(n) \approx\ & \mathbb{E}\{c_{\ell\ell}(n)c_{mm}(n)\}\mathbf{S}_{\ell m}(n-1) \\ & + \sum_{p \in \bar{\mathcal{N}}_\ell} \sum_{r \in \bar{\mathcal{N}}_m} \mathbb{E}\{c_{p\ell}(n)c_{rm}(n)\}\mathbf{W}_{pr}(n-1) \\ & + \sum_{r \in \bar{\mathcal{N}}_m} \mathbb{E}\{c_{\ell\ell}(n)c_{rm}(n)\}\mathbf{X}_{\ell r}(n-1) \\ & + \sum_{p \in \bar{\mathcal{N}}_\ell} \mathbb{E}\{c_{p\ell}(n)c_{mm}(n)\}\mathbf{X}_{mp}^T(n-1) + \mathbf{Q},\end{aligned} \tag{B.20}$$

where we have defined

$$\overline{\mu}_k \triangleq \frac{\mu_k}{\sigma_{u,k}^2(M-2)}, \tag{B.21}$$

with $\sigma_{u,k}^2$ being the variance of the input signal at node k, with $k = 1, 2, \ldots, N$.

C

Acronyms and abbreviations

- **ABT.** Adaptive Balanced Transmitter.
- **ACW.** Adaptive Combination Weights.
- **APA.** Affine Projection Algorithm.
- **ATC.** Adapt-then-Combine.
- **BT.** Bluetooth.
- **C2.** Command and Control.
- **CD-ATC.** Censoring Decoupled Adapt-then-Combine.
- **CTA.** Combine-then-Adapt.
- **D-ATC.** Decoupled Adapt-then-Combine.
- **DARPA.** Defense Advanced Research Projects Agency.

- **DP.** Dynamic Programming.
- **IEEE.** Institute of Electrical and Electronic Engineers.
- **IoT.** Internet of the Things.
- **LMS.** Least Mean Squares.
- **LS.** Least Squares.
- **MAC.** Medium Access Control.
- **MDP.** Markov Decision Process.
- **MSD.** Mean Square Deviation.
- **MSE.** Mean Square Error.
- **NLMS.** Normalized Least Mean Squares.
- **NMSD.** Network Mean Square Deviation.
- **NMSE.** Network Mean Square Error.
- **NS.** Non selective.
- **NSD-ATC.** Non Selective Decoupled Adapt-then-Combine.
- **QoI.** Quality of Information.
- **QoS.** Quality of Service.
- **SAP.** Stochastic Approximate Policy.
- **SNR.** Signal-to-noise ratio.
- **WSN.** Wireless Sensor Network.

D

List of Publications

In this appendix we list the publications where work related to this thesis has been presented.

Journal Publications

1. **J. Fernandez-Bes**, M. T. M. Silva, J. Arenas-García, and L. A. Azpicueta-Ruiz. Decoupled Adapt-then-Combine diffusion networks with adaptive combiners (UNDER REVIEW). [available at `http://arxiv.org/abs/1504.01982v1`]

2. **J. Fernandez-Bes**, R. Arroyo-Valles, J. Cid-Sueiro. Asymptotic Analysis of Cooperative Censoring Policies in Sensor Networks. Ad Hoc Networks, 29(0):63-77, Jun.2015.

3. **J. Fernandez-Bes**, J. Cid-Sueiro, A. G. Marques. An MDP Model for Censoring in Harvesting Sensors: Optimal and Approximated Solutions. IEEE Journal on Selected Areas in Communications, PP(99):1-1, 2015.

4. **J. Fernandez-Bes**, L. A. Azpicueta-Ruiz, J. Arenas-García, and M. T. M. Silva. Distributed estimation in diffusion networks using affine least-squares combiners . Digital Signal Processing. 36:1-14, Jan. 2015.

Conference Publications

1. **J. Fernandez-Bes**, V. Elvira, and S. Van Vaerenbergh. A probabilistic Least-Mean-Squares Filter. In International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, Brisbane (Australia). May 2015.

2. **J. Fernandez-Bes**, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-García. Estimación distribuida con redes de difusión: algoritmo con combinadores adaptativos y agentes que preservan estimaciones locales . In XXIX Simposiun Nacional de la Unión Científica Internacional de Radio (URSI), 2014, Valencia (Spain). Sep. 2014.

3. **J. Fernandez-Bes**, J. Arenas-García, and Ali H. Sayed. Adjustment of Combination Weights Over Adaptive Diffusion Networks. In International Conference on Acustics, Speech and Signal Processing (ICASSP), 2014, Firenze (Italy). May 2014.

4. **J. Fernandez-Bes**, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-García. Improved least-squares-based combiners for diffusion networks. In International Symposium on Wireless Communication Systems (ISWCS), 2013, Ilmenau (Germany). Aug. 2013.

5. **J. Fernandez-Bes**, J. Cid-Sueiro, A. G. Marques. Battery-Aware Selective Transmitters in Energy-Harvesting Sensor Networks: Optimal Solution and Stochastic Dual Approximation. In International Symposium on Wireless Communication Systems (ISWCS), 2013, Ilmenau (Germany). Aug. 2013.

6. **J. Fernandez-Bes**, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-García. A Novel Scheme for Diffusion Networks with Least-Squares Adaptive Combiners. In Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on, Santander (Spain). Sep. 2012.

7. **J. Fernandez-Bes**, and J. Cid-Sueiro. Decentralized Detection with Energy-Aware Greedy Selective Sensors. In Cognitive Information Processing (CIP), 2012 Third International Workshop. Baiona (Spain). May 2012.

8. **J. Fernandez-Bes**, R. Arroyo-Valles, and J. Cid-Sueiro. Cooperative data censoring for energy-efficient communications in sensor networks. In Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on pages 1–6. Beijing (China). Sep. 2011.

Bibliography

[1] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, May 2005.

[2] I. F. Akyildiz and M. C. Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.

[3] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. A cross-layer protocol for wireless sensor networks. In *40th Annual Conference on Information Sciences and Systems*, pages 1102–1107. IEEE, 2006.

[4] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal. Data aggregation in wireless sensor networks-exact and approximate algorithms. In *Workshop on High Performance Switching and Routing (HPSR)*, pages 241–245. IEEE, 2004.

[5] M. A. Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin. Markov decision processes with applications in wireless sensor networks: A survey. *arXiv preprint arXiv:1501.00644*, 2015.

[6] S. Appadwedula, V. V. Veeravalli, and D. L. Jones. Energy-efficient detection in sensor networks. *IEEE Journal on Selected Areas Communications*, 23(4):693–702, Apr. 2005.

[7] S. Appadwedula, V.V. Veeravalli, and D.L. Jones. Decentralized detection with censoring sensors. *IEEE Transactions on Signal Processing*, 56(4):1362–1373, Apr. 2008.

[8] R. Arroyo-Valles, S. Maleki, and G. Leus. A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks. In *14th IEEE*

International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pages 146–150, 2013.

[9] R. Arroyo-Valles, A. G. Marques, and J. Cid-Sueiro. Optimal Selective Transmission under Energy Constraints in Sensor Networks. *IEEE Transactions on Mobile Computing*, 8(11):1524–1538, Nov. 2009.

[10] R. Arroyo-Valles, A. G. Marques, and J. Cid-Sueiro. Optimal selective forwarding for energy saving in wireless sensor networks. *IEEE Transactions on Wireless Communicationss*, 10(1):164 –175, Jan. 2011.

[11] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, and J. Arenas-García. Least-squares adaptation of affine combinations of multiple adaptive filters. In *Proc. of IEEE International Symposium on Circuits and Systems*, pages 2976–2979, Paris, France, 2010.

[12] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer Communications*, 30(7):1655–1695, May 2007.

[13] R. Bellman. On the theory of dynamic programming. *Proc. of the National Academy of Sciences of the United States of America*, 38(8):716, Aug. 1952.

[14] N. J. Bershad and J. C. M. Bermudez. Mean-square stability of the Normalized Least-Mean Fourth algorithm for white Gaussian inputs. *Digital Signal Processing*, 21(6):694–700, Dec. 2011.

[15] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995.

[16] D. P. Bertsekas. *Dynamic Programming and Optimal Control, 3rd edition, volume II.* Athena Scientific, 2011.

[17] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability.* Athena Scientific, 2002.

[18] A. Bielsa. Smart City Project in Santander Monitors Environmental Parameters and Parking Slots — Libelium. http://www.libelium.com/smart_santander_smart_parking. Accessed: 2015-02-25.

[19] P. Blasco, D. Gunduz, and M. Dohler. A learning theoretic approach to energy harvesting communication system optimization. *IEEE Transactions on Wireless Commun.*, 12(4):1872–1882, Apr. 2013.

[20] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. of 44th IEEE Conference on Decision and Control European Control Conference (CDC-ECC'05)*, pages 2996–3000, Seville, Spain, 2005.

[21] R. S. Blum and B. M. Sadler. Energy efficient signal detection in sensor networks using ordered transmissions. *IEEE Transactions on Signal Processing*, 56(7):3229–3235, Jul. 2008.

[22] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, Jan.-Mar. 2004.

[23] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. L.P. Corrêa. From wireless sensors to field mapping: Anatomy of an application for precision agriculture. *Computers and Electronics in Agriculture*, 58(1):25–36, Aug. 2007.

[24] J. A. Carta, P. Ramírez, and S. Velázquez. A review of wind speed probability distributions used in wind energy analysis: Case studies in the Canary islands. *Renewable & Sustainable Energy Reviews*, 13:933–955, Jun. 2009.

[25] M. Castro, A. J. Jara, and A. F. G. Skarmeta. Smart lighting solutions for smart cities. In *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 1374–1379. IEEE, 2013.

[26] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 56(5):1865–1877, May 2008.

[27] F. S. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048, Mar. 2010.

[28] J. Chen, C. Richard, and A. H. Sayed. Multitask diffusion adaptation over networks. *IEEE Transactions on Signal Processing*, 62(16):4129–4144, Aug. 2014.

[29] S. Chouvardas, K. Slavakis, and S. Theodoridis. Adaptive robust distributed learning in diffusion sensor networks. *IEEE Transactions on Signal Processing*, 59(10):4692–4707, Oct. 2011.

[30] R. W. Claeys and G. S. Decamp. System for verifying and recording drug administration to a patient, August 1 1989. US Patent 4,853,521.

[31] M. C. Costa and J. C. M. Bermudez. An improved model for the normalized LMS algorithm with Gaussian inputs and large number of coefficients. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume II, pages 1385–1388, Orlando, FL, 2002.

[32] T. Dang, S. Frolov, N. Bulusu, W. Feng, and A. Baptista. Near optimal sensor selection in the Columbia river (CORIE) observation network for data assimilation using genetic algorithms. In *Distributed Computing in Sensor Systems*, pages 253–266. Springer, 2007.

[33] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed. Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks. *IEEE Transactions on Signal Processing*, 61(12):3183–3197, Jun. 2013.

[34] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[35] P. S. R. Diniz. *Adaptive filtering: algorithms and practical implementation*, volume 694. Springer Verlag, 2008.

[36] C. Donovan, A. Dewan, D. Heo, and H. Beyenal. Batteryless, wireless sensor powered by a sediment microbial fuel cell. *Environmental science & technology*, 42(22):8591–8596, Oct. 2008.

[37] G. L. Duckworth, J. E. Barger, and D. C. Gilbert. Acoustic counter-sniper system, January 23 2001. US Patent 6,178,141.

[38] European Commision. ICT 2015 - Information and Communications Technologies: Internet of Things and Platforms for Connected Smart Objects. `http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/9091-ict-30-2015.html`. Accessed: 2014-10-29.

[39] European Commision. Smart Cities — Digital Agenda for Europe — European Commission. `http://ec.europa.eu/digital-agenda/en/smart-cities`. Accessed: 2015-02-25.

[40] P. Fergus, K. Kifayat, S. Cooper, M. Merabti, and A. El Rhalibi. A framework for physical health improvement using wireless sensor networks and gaming. In *3rd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1–4. IEEE, 2009.

[41] J. Fernandez-Bes, J. Arenas-Garcia, and A. H. Sayed. Adjustment of combination weights over adaptive diffusion networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6409–6413, Florence, Italy, 2014. IEEE.

[42] J. Fernandez-Bes, J. Arenas-García, M. T. M. Silva, and L. A. Azpicueta-Ruiz. Decoupled Adapt-then-Combine diffusion networks with adaptive combiners. *arXiv preprint arXiv:1504.01982*, 2015.

[43] J. Fernandez-Bes, R. Arroyo-Valles, and J. Cid-Sueiro. Asymptotic analysis of cooperative censoring policies in sensor networks. *Ad Hoc Networks*, 29(0):63–77, Jun. 2015.

[44] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-García, and M. T. M. Silva. Distributed estimation in diffusion networks using affine least-squares combiners. *Digital Signal Processing*, 36:1–14, Jan. 2015.

[45] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-Garcıa. A novel scheme for diffusion networks with least-squares adaptive combiners.

In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2012.

[46] J. Fernandez-Bes and J. Cid-Sueiro. Decentralized detection with energy-aware greedy selective sensors. In *3rd International Workshop on Cognitive Information Processing (CIP)*, pages 1–6, 2012.

[47] J. Fernandez-Bes, J. Cid-Sueiro, and A. G. Marques. An MDP Model for Censoring in Harvesting Sensors: Optimal and Approximated Solutions. *IEEE Journal on Selected Areas in Communications*, PP(99):1–1, 2015.

[48] J. Fernandez-Bes, V. Elvira, and S. Van Vaerenbergh. A Probabilistic Least-Mean-Squares-Filter. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Briebane, Australia, 2015. IEEE.

[49] J. Fernandez-Bes, A. G. Marques, and J. Cid-Sueiro. Battery-aware selective transmitters in energy-harvesting sensor networks: Optimal solution and stochastic dual approximation. In *Proc. of the 10th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5. VDE, 2013.

[50] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel. Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Computing*, 10(1):28–39, Jan.-Mar. 2011.

[51] K. Gatsis, A. Ribeiro, and G. J. Pappas. Optimal power management in wireless control systems. In *American Control Conference (ACC), 2013*, pages 1562–1569. IEEE, 2013.

[52] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries. *IEEE Transactions on Wireless Communications*, 9(2):581–593, Feb. 2010.

[53] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems 6*. Citeseer, 1994.

[54] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin. Distributed energy-aware diffusion least mean squares: Game-theoretic learning. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):821–836, Oct. 2013.

[55] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi. Designing intelligent energy harvesting communication systems. *IEEE Communications Magazine*, 52(1):210–216, Jan. 2014.

[56] G. P. Hancke and G. P De Carvalho e Silva, B. Hancke Jr. The role of advanced sensing in smart cities. *Sensors*, 13(1):393–425, Dec. 2012.

[57] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, 4th edition, 2002.

[58] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, and T. F. Abdelzaher. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):1–38, Feb. 2006.

[59] F. Hernando-Gallego. Personal webpage. `http://www.tsc.uc3m.es/~fhernando`. Accessed: 2015-03-12.

[60] C. K. Ho, P. D. Khoa, and P. C. Ming. Markovian models for harvested energy in wireless communications. In *Communication Systems (ICCS), 2010 IEEE International Conference on*, pages 311–315. IEEE, 2010.

[61] C. K. Ho and R. Zhang. Optimal energy allocation for wireless communications with energy harvesting constraints. *IEEE Transactions on Signal Processing*, 60(9):4808–4818, Sep. 2012.

[62] R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[63] O. C. Imer and T. Basar. Optimal estimation with limited measurements. *International Journal of Systems, Control and Communications*, 2(1):5–29, 2010.

[64] Institute of Electrical and Electronics Engineers. Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). *IEEE Std. 802.15.4-2006*, pages i –305, 2006.

[65] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6(4):32, Sep. 2007.

[66] P. S. Khairnar and N. B. Mehta. Power and discrete rate adaptation for energy harvesting wireless nodes. In *IEEE International Conference on Communications*, pages 1–5. IEEE, 2011.

[67] S. Kianpisheh and N. M. Charkari. Dynamic power management for sensor node in wsn using average reward mdp. In *Wireless Algorithms, Systems, and Applications*, pages 53–61. Springer, 2009.

[68] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 254–263. IEEE, 2007.

[69] J. Lei, R. Yates, and L. Greenstein. A generic model for optimizing single-hop transmission policy of replenishable sensors. *IEEE Transactions on Wireless Communications*, 8(2):547 –551, Feb. 2009.

[70] Libelium. Waspmote - Wireless Sensor Networks 802.15.4 ZigBee Mote - Open Source Sensor Device — Libelium. `http://www.libelium.com/es/products/waspmote/`. Accessed: 2015-02-26.

[71] Libelium. Waspmote Sensors Overview - Complete List of Sensors — Libelium. `http://www.libelium.com/es/products/waspmote/sensors`. Accessed: 2015-05-22.

[72] R. Lin, Z. Wang, and Y. Sun. Wireless sensor networks solutions for real time monitoring of nuclear power plant. In *5th World Congress on Intelligent Control and Automation (WCICA)*, volume 4, pages 3663–3667. IEEE, 2004.

[73] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077, Jul. 2007.

[74] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, Jun. 2008.

[75] K. Lorincz, D. J. Malan, T. R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, Oct.–Dec. 2004.

[76] K. S. Low, W. N. N. Win, and Meng J. Er. Wireless sensor networks for industrial environments. In *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, volume 2, pages 271–276. IEEE, 2005.

[77] R. Maas, C. Huemmer, C. Hofmann, and W. Kellermann. On bayesian networks in speech signal processing. In *Proc. of 11th ITG Symposium Speech Communication*, pages 1–4, Erlangen, Germany, 2014. VDE.

[78] R. Madan, S. Cui, S. Lall, and A. Goldsmith. Cross-Layer design for lifetime maximization in Interference-Limited wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(11):3142–3152, Nov. 2006.

[79] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks*, volume 5, 2004.

[80] S. Maleki, A. Pandharipande, and G. Leus. Energy-efficient distributed spectrum sensing for cognitive sensor networks. *IEEE Sensors Journal*, 11(3):565–573, Jan. 2011.

[81] J. E. Mazo. On the independent theory of equalizer convergence. *Bell Systems Technology Journal*, 58:963–993, May.–Jun. 1979.

[82] N. Michelusi, K. Stamatiou, L. Badia, and M. Zorzi. Operation policies for energy harvesting devices with imperfect state-of-charge knowledge. In *Proc. IEEE International Conference on Communications*, pages 5782–5787, 2012.

[83] N. Michelusi, K. Stamatiou, and M. Zorzi. On optimal transmission policies for energy harvesting devices. In *IEEE Information Theory and Applications Workshop*, pages 249–254. IEEE, 2012.

[84] N. Michelusi, K. Stamatiou, and M. Zorzi. Transmission policies for energy harvesting sensors with time-correlated energy supply. *IEEE Transactions on Communications*, 61(7):2988–3001, Jul. 2013.

[85] N. Michelusi and M. Zorzi. Optimal random multiaccess in energy harvesting wireless sensor networks. In *IEEE International Conference on Communications Workshops (ICC)*, pages 463–468. IEEE, 2013.

[86] J. Minkoff. Comment on the "Unnecessary assumption of statistical independence between reference signal and filter weights in feedforward adaptive systems". *IEEE Transactions on Signal Processing*, 49:1109, May 2001.

[87] MIT Technology Review. 10 Emerging Technologies That Will Change the World. `http://www2.technologyreview.com/Infotech/13060/`, Feb. 2003. Accessed: 2014-10-29.

[88] M. Niedzwiecki. On tracking characteristics of weighted least squares estimators applied to nonstationary system identification. *IEEE Transactions on Automatic Control*, 1988.

[89] M. Niedzwiecki. *Identification of time-varying processes.* Wiley New York, 2000.

[90] K. Ota, M. Dong, Z. Cheng, J. Wang, X. Li, and X. S. Shen. ORACLE: Mobility control in wireless sensor and actor networks. *Computer Communications*, 35(9):1029–1037, 2012.

[91] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, 4(1):18–27, 2005.

[92] S. Pino-Povedano, R. Arroyo-Valles, and J. Cid-Sueiro. Selective forwarding for energy-efficient target tracking in sensor networks. *Signal Processing*, 94:557–569, 2014.

[93] H. V. Poor. *An introduction to signal detection and estimation*. Springer Science & Business Media, 1994.

[94] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, 2007.

[95] K. D. Prasad and S.V.S. Murty. Wireless Sensor Networks–A potential tool to probe for water on Moon. *Advances in Space Research*, 48(3):601–612, Aug. 2011.

[96] M.L. Puterman. *Markov Decision Processes*. Wiley-Interscience, 2005.

[97] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, Jul. 2000.

[98] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, Feb. 1989.

[99] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2):40–50, Mar. 2002.

[100] C. Rago, P. Willett, and Y. Bar-Shalom. Censoring sensors: A low-communication-rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems*, 32(2):554–568, Apr. 1996.

[101] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, Feb. 2006.

[102] Sh. Roundy, E. S. Leland, J. Baker, E. Carleton, E. Reilly, E. Lai, B. Otis, J. M. Rabaey, P. K. Wright, and V. Sundararajan. Improving power output

for vibration-based energy scavengers. *IEEE Pervasive Computing*, 4(1):28–36, Jan.–Mar. 2005.

[103] M. Saerens. Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks*, 11(6):1263–1271, Mar. 2000.

[104] C. Samson and V. U. Reddy. Fixed-point error analysis of the normalized ladder algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31:1177–1191, Oct. 1983.

[105] A. H. Sayed. *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[106] A. H. Sayed. *Adaptive Filters*. John Wiley & Sons, NJ, 2008.

[107] A. H. Sayed. Diffusion adaptation over networks. In R. Chellapa and S. Theodoridis, editors, *Academic Press Library in Signal Processing: array and statistical signal processing*, volume 3, chapter 9, pages 323–456. Academic Press, 2014 [See also arXiv:1205.4220, May 2012].

[108] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic. Diffusion strategies for adaptation and learning over networks. *IEEE Signal Processing Magazine*, 30(3):155–171, May 2013.

[109] I. D. Schizas, G. Mateos, and G. B. Giannakis. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Transactions on Signal Processing*, 57(6):2365–2382, Jun. 2009.

[110] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proc. of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 272–287. ACM, 2001.

[111] Shimmer Research Ireland. Wireless GSR Sensor — Shimmer Galvanic Skin Response Sensor for realtime GSR Biofeedback. `http://www.shimmersensing.com/shop/shimmer3-wireless-gsr-sensor`. Accessed: 2015-02-25.

[112] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys & Tutorials*, 13(3):443–461, Sep. 2011.

[113] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*. MIT Press, 1998.

[114] N. Takahashi, I. Yamada, and A. H. Sayed. Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 58:4795–4810, Sep. 2010.

[115] J. N. Tsitsiklis. Decentralized detection by a large number of sensors. *Mathematics of Control, Signals and Systems*, 1(2):167–182, Jun. 1988.

[116] S.-Y. Tu and A. H. Sayed. Optimal combination rules for adaptation and learning over networks. In *Proc. of 4th IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Process. (CAMSAP)*, pages 317–320, 2011.

[117] S.-Y. Tu and A. H. Sayed. On the influence of informed agents on learning and adaptation over networks. *IEEE Transactions on Signal Processing*, 61(6):1339–1356, March 2013.

[118] K. Tutuncuoglu and A. Yener. Communicating with energy harvesting transmitters and receivers. In *2012 Information Theory and Applications Workshop (ITA)*, pages 240–245. IEEE, 2012.

[119] K. Tutuncuoglu and A. Yener. Optimum transmission policies for battery limited energy harvesting nodes. *IEEE Transactions on Wireless Communications*, 11(3):1180–1189, Mar. 2012.

[120] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang. Energy Harvesting Wireless Communications: A Review of Recent Advances. *IEEE Journal on Selected Areas in Communications*, 33(3):360–381, Mar. 2015.

[121] L. Van Hoesel, T. Nieberg, J. Wu, and P.J.M. Havinga. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Communications*, 11(6):78–86, Dec. 2004.

[122] C. Van Phan, Y. Park, H. Choi, J. Cho, and J. G. Kim. An energy-efficient transmission strategy for wireless sensor networks. *IEEE Transactions on Consumer Electronics*, 56(2):597–605, May 2010.

[123] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría. Kernel recursive least-squares tracker for time-varying regression. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1313–1326, Jul. 2012.

[124] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors i. fundamentals. *Proc. of the IEEE*, 85(1):54–63, Jan. 1997.

[125] G. Wahba. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM Journal on Numerical Analysis*, 14(4):651–667, May 1977.

[126] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, Jan. 2001.

[127] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

[128] M. Wiering and Ma. Van Otterlo. Reinforcement learning. In *Adaptation, Learning, and Optimization*, volume 12. Springer, 2012.

[129] F. Xiangning and S. Yulin. Improvement on LEACH protocol of wireless sensor network. In *International Conference on Sensor Technologies and Applications (SensorComm)*, pages 260–264. IEEE, 2007.

[130] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis. Distributed compression-estimation using wireless sensor networks. *Signal Processing Magazine, IEEE*, 23(4):27–41, Jul. 2006.

[131] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53:65–78, Sep. 2004.

[132] J. Yang and S. Ulukus. Optimal packet scheduling in an energy harvesting communication system. *IEEE Transactions on Communication*, 60(1):220–230, Feb. 2012.

[133] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMInfocom)*, volume 3, pages 1567–1576. IEEE, 2002.

[134] G. G. Yin and H.J. Kushner. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

[135] C.-K. Yu and A. H. Sayed. A strategy for adjusting combination weights over adaptive networks. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Process. (ICASSP)*, pages 4579–4583, Vancouver, Canada, 2013.

[136] S. Zahedi and C. Bisdikian. A framework for qoi-inspired analysis for sensor network deployment planning. In *Proc. of the 3rd international conference on Wireless internet*, page 28. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[137] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proc. of the 2nd international conference on Embedded Networked Sensor Systems*, pages 227–238. ACM, 2004.

[138] X. Zhang, C. Yin, and G. Yue. An optimal power allocation algorithm in distributed sensing. In *IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2010.

[139] X. Zhao and A. H. Sayed. Asynchronous diffusion adaptation over networks. In *Proc. of 20th European Signal Processing Conference (EUSIPCO)*, pages 86–90, Bucarest, Romania, 2012. IEEE.

[140] X. Zhao and A. H. Sayed. Performance limits for distributed estimation over LMS adaptive networks. *IEEE Transactions on Signal Processing*, 60(10):5107–5124, Oct. 2012.