University of Padova

Department of Information Engineering

Ph.D. School in Information Engineering

Information Science and Technology

XXVI Class

Compression vs Transmission Tradeoffs for Energy Harvesting Sensor Networks

Ph.D. candidate:

Davide Zordan

Supervisor:

Prof. Michele Rossi

Course coordinator:

Prof. Carlo Ferrari

Ph.D. School director:

Prof. Matteo Bertocco

Academic Year 2013-2014

Abstract

The operation of Energy Harvesting Wireless Sensor Networks (EHWSNs) is a very lively area of research. This is due to the increasing inclination toward green systems, in order to reduce the energy consumption of human activities at large and to the desire of designing networks that can last unattended indefinitely (see, e.g., the nodes employed in Wireless Sensor Networks, WSNs). Notably, despite recent technological advances, batteries are expected to last for less than ten years for many applications and their replacement is often prohibitively expensive. This problem is particularly severe for urban sensing applications, think of, e.g., sensors placed below the street level to sense the presence of cars in parking lots, where the installation of new power cables is impractical. Other examples include body sensor networks or WSNs deployed in remote geographic areas. In contrast, EHWNs powered by energy scavenging devices (renewable power) provide potentially maintenance-free perpetual network operation, which is particularly appealing, especially for highly pervasive Internet of Things.

Lossy temporal compression has been widely recognized as key for Energy Constrained Wireless Sensor Networks (WSN), where the imperfect reconstruction of the signal is often acceptable at the data collector, subject to some maximum error tolerance. The first part of this thesis deals with the evaluation of a number of lossy compression methods from the literature, and the analysis of their performance in terms of compression efficiency, computational complexity and energy consumption. Specifically, as a first step, a performance evaluation of existing and new compression schemes, considering linear, autoregressive, FFT-/DCT- and Wavelet-based models is carried out, by looking at their performance as a function of relevant signal statistics. After that, closed form expressions for their overall energy consumption and signal representation accuracy are obtained through numerical fittings. Lastly, the benefits that lossy compression methods bring about in interference-limited multi-hop networks are evaluated. In this scenario the channel access is a source of inefficiency due to collisions and transmission scheduling. The results reveal that the DCT-based schemes are the best option in terms of compression efficiency but are inefficient in terms of energy consumption. Instead, linear methods lead to substantial savings in terms of energy expenditure by, at the same time, leading to satisfactory compression ratios, reduced network delay and increased reliability performance.

The subsequent part of the thesis copes with the problem of energy management for EHWSNs where sensor batteries are recharged via the energy harvested through a solar panel and sensors can choose to compress data before transmission. A scenario where a single node communicates with a single receiver is considered. The task of the node is to periodically sense some physical signal and report the measurements to the receiver (sink). We assume that this task is delay tolerant, i.e., the sensor can store a certain number of measurements in the memory buffer and send one or more packets of data after some time. Since most physical signals exhibit strong temporal correlation, the data in the buffer can often be compressed by means of a lossy compression method in order to reduce the amount of data to be sent. Lossy compression schemes allow us to select the compression ratio and trade some accuracy in the data reconstruction at the receiver for more energy savings at the transmitter. Specifically, our objective is to obtain the policy, i.e., the set of decision rules that describe the node behavior, that jointly maximizes throughput and reconstruction fidelity at the sink while meeting some predefined energy constraints, e.g., the battery charge level should never go below a guard threshold. To obtain this policy, the system is modeled as a Constrained Markov Decision Process (CMDP), and solved through Lagrangian Relaxation and Value Iteration Algorithm. The optimal policies are then compared with heuristic policies in different energy budget scenarios. Moreover the impact of the delay on the knowledge of the Channel State Information is investigated.

Two more parts of this thesis deal with the development of models for the generation of space-time correlated signals and for the description of the energy harvested by outdoor photovoltaic panels. The former are very useful to prove the effectiveness of the proposed data gathering solutions as they can be used in the design of accurate simulation tools for WSNs. In addition, they can also be considered as reference models to prove theoretical results for data gathering or compression algorithms. The latter are especially useful in the investigation and in the optimization of EHWSNs. These models will be presented at the beginning and then intensively used for the analysis and the performance evaluation of the schemes that are treated in the remainder of the thesis.

Sommario

Quello delle Energy Harvesting Wireless Sensor Networks (EHWSNs) è attualmente un campo di ricerca molto attivo. Ciò è principalmente dovuto al crescente interesse dimostrato verso i sistemi ”green”, con l’obiettivo di ridurre il consumo energetico delle attività umane in generale e il desiderio di progettare reti autosufficienti che possono durare indefinitamente (si pensi, ad esempio , ai nodi impiegati in reti di sensori wireless , WSNs). In particolare, nonostante i recenti progressi tecnologici, per molte applicazioni le batterie si dimostrano durare meno di dieci anni e il costo per la loro sostituzione è spesso proibitivo. Questo problema è particolarmente grave per le applicazioni di rilevamento urbano, si pensi ad esempio ad uno scenario in cui dei sensori sono posizionati al di sotto del manto stradale per il rilevamento della presenza di auto nei parcheggi, dove l’installazione di nuovi cavi di alimentazione o la sostituizione delle batterie non sono praticabili . Altri esempi includono le ”body sensor networks” o le reti di sensori distribuite in aree geografiche remote o inaccessibili. Al contrario, EHWSNs alimentate da dispositivi di energy scavenging (energia rinnovabile) possono dare vita a reti perpetue e potenzialmente esenti da manutenzione, che sono particolarmente attraenti, soprattutto per il nuovo concetto altamente pervasivo di Internet of Things.

La compressione temporale con perdite (lossy temporal compression) è ampiamente riconosciuta come componente fondamentale per il funzionamento delle reti di sensori con energia limitata, dove la ricostruzione imperfetta del segnale al punto di raccolta è spesso accettabile, fino ad un certo limite massimo sulla tolleranza di errore. Una parte di questa tesi tratta la valutazione prestazionale di un significativo numero di metodi di compressione con perdita tratti dalla letteratura, e l’analisi delle loro prestazioni in termini di efficienza di compressione, complessità computazionale e consumo energetico. In dettaglio, come primo passo, viene proposta una valutazione delle prestazioni di sistemi di compressione esistenti e nuovi, tra cui: modelli lineari, autoregressivi, basati su FFT/DCT e Wavelet, individuando le loro prestazioni in funzione delle statistiche dei segnali rilevanti. Dopo di che, attraverso interpolazione numerica, verranno derivate delle espressioni in forma chiusa per il consumo globale di energia e la precisione di rappresentazione del segnale. Infine, verranno valutati i benefici che i metodi di compressione con perdita possono portare in reti multi-hop con interferenze limitate. In questo scenario il canale di accesso diventa fonte di inefficienza attraverso collisioni e metodo di accesso al mezzo. I risultati rivelano che le tecniche basate su DCT sono la scelta migliore in termini di efficienza di compressione, ma non risultano efficienti in termini di consumo energetico. Al contrario, metodi lineari possono dar luce a notevoli risparmi in termini di dispendio energetico, e al tempo stesso, portare a rapporti di compressione soddisfacenti, ritardi di rete ridotti e migliore affidabilità.

La parte successiva di questa tesi affronta il problema della gestione energetica per EHWSNs nelle quali le batterie dei nodi sensore vengono ricaricate attraverso l'energia raccolta da un pannello solare e sensori possono scegliere di comprimere i dati prima della trasmissione. A tal fine viene considerato uno scenario in cui un singolo nodo comunica con un singolo ricevitore. L'attività del nodo è quella di campionare periodicamente qualche segnale fisico e riportare le misurazioni al ricevitore (sink). Tale attività viene assunta essere tollerante al ritardo, ovvero, il sensore pu memorizzare un certo numero di misurazioni nel buffer di memoria e inviare uno o più pacchetti di dati aggregati dopo un certo tempo. Poiché la maggior parte dei segnali fisici manifestano una forte correlazione temporale, i dati nel buffer possono eventualmente essere compressi mediante un metodo di compressione con perdita al fine di ridurre la quantità di dati da inviare. Attraverso metodi di compressione con perdita di dati che permettono di selezionare il rapporto di compressione è possibile scambiare un po' di accuratezza nella ricostruzione dei dati al ricevitore per ottenere maggiori risparmi di energia al trasmettitore. In dettaglio, l'obiettivo di questa parte della tesi è quello di ottenere la politica, cioè l'insieme di regole decisionali che descrivono il comportamento del nodo sensore, che massimizza il throughput unitamente alla fedeltà di ricostruzione al punto di raccolta soddisfacendo al tempo stesso alcuni vincoli energetici predefiniti, ad esempio, il livello di carica della batteria non deve mai scendere al di sotto di una soglia di guardia. Per ottenere la politica ottima, il sistema verrà modellato attraverso un Processo Decisonale di Markov Vincolato (Constrained Markov Decision Process, CMDP), e risolto attraverso il

rilassamento lagrangiano e l'algoritmo di Value Iteration. Le politiche ottimali verranno poi confrontate con alcune politiche euristiche in diversi scenari di bilancio energetico. Verrà inoltre studiato l'impatto del ritardo sulla conoscenza dello stato del canale.

Due ulteriori parti di questa tesi riguardano lo sviluppo di modelli per la generazione di segnali correlati nello spazio e nel tempo, e per la descrizione statistica dell'energia raccolta da pannelli fotovoltaici esterni. I primi sono utili per testare l'efficacia di algoritmi di raccolta dati e possono venire impiegati nella progettazione di accurati strumenti di simulazione per reti di sensori. Inoltre possono venire impiegati come modelli di riferimento per dimostrare risultati teorici per algoritmi di raccolta dati o di compressione. Gli ultimi sono particolarmente utili per lo studio e l'ottimizzazione delle EHWSNs. Entrambi i modelli verranno introdotti nella parte iniziale della tesi e successivamente utilizzati per tutto il corpo centrale della stessa.

List of Acronyms

AR	Auto-Regressive
ARMA	Auto-Regressive Moving Average
A-ARMA	Adaptive Auto-Regressive Moving Average
ACF	Auto-Correlation Function
ACK	Acknowledgment
AST	Apparent Solar Time
CC	Constant Compression
CMDP	Constrained Markov Decision Process
CS	Compressive Sensing
CSI	Channel State Information
CTS	Clear-To-Send
DC	Direct Current
DCT	Discrete Cosine Transform
DE	Delta Encoding
DPCM	Differential Pulse Coding Modulation
DQPSK	Differential Quadrature Phase Shift Keying

EC	Entropy Coding
EHWSN	Energy Harvesting Wireless Sensor Network
FFT	Fast Fourier Transform
FWT	Fast Wavelet Transform
GMA	Greenwich Meridian Angle
iid	Independent and Identically Distributed
KS	Kernel Smoothing
LEC	Lossless Entropy Compression
LPF	Low Pass Filter
LTC	Lightweight Temporal Compression
LTE	Long Term Evolution
M-AAR	Modified Adaptive Auto-Regressive
MAC	Medium Access Control
MPPT	Maximum Power Point Tracking
RMSE	Root Mean Square Error
NC	No Compression
PDR	Packet Delivery Ratio
PE	Power Exponential
PLA	Piecewise Linear Approximation
PLAMLiS	Piecewise Linear Approximation with Minimum number of Line Segment
PR	Polynomial Regression
PV	Photo-Voltaic

RLE	Run Length Encoding
RQ	Rational Quadratic
RTS	Request-To-Send
S-MAC	Synchronous Medium Access Control
SNR	Signal-to-Noise Ratio
UTC	Universal Time Coordinated
VI	Value Iteration
WSN	Wireless Sensor Network

Contents

1 Introduction

A Wireless Sensor Network (WSN) consists of sensor nodes deployed over a geographical area to monitor physical phenomena like temperature, humidity, seismic events, polluting agents concentration and so on. Typically, a sensor node is a small device that is composed of three main parts: a sensing subsystem responsible for collecting data from the environment, a processing subsystem dedicated to data handling and local data processing and a communication subsystem for data transmission. In addition to that, sensor nodes are usually powered by small, and generally disposable, batteries. Therefore, protocols and applications for WSNs must be carefully designed in order to make the most efficient use of the limited resources available in terms of computation capabilities and energy consumption.

In recent years, researchers have focused on the design of energy efficient protocols for data gathering [1], data aggregation and in–network compression [2], as well as Medium Access Control (MAC) [3] and Routing protocols [4]. The main direction followed by most of the works in the literature is that of trading off communications for computational complexity. In particular, since usually local computation consumes significantly less energy than communication, the proposed solutions often aimed at reducing the amount of data transmission in order to save energy. To this end, various algorithms and protocols have been proposed, claiming energy efficiency as the main objective but often neglecting the computational cost in the performance evaluation and assuming it as a small fraction of the communication cost. This however is not always the case, since the processing unit of the sensor nodes is also energy, memory and power constrained and the number of operations needed to perform some signal processing may lead to non negligible energy consumptions.

In this thesis we focus on the study of lossy compression techniques for WSNs. In particular, our aim is to carry out a thorough performance evaluation of new as well as previously proposed compression algorithms in terms of compression capabilities and, most importantly, computational complexity and energy consumption. This is done in Chapter 4, where we first consider the scenario of a single hop communication link and evaluate the performance and the energy efficiency of the selected lossy compression schemes. After that, we also analyze the case of a multi hop scenario where a realistic MAC protocol is considered, and we verify how the results obtained in the single hop case extend to the multi hop case.

The optimization objective drastically changes when we consider the possibility of harvesting energy from the environment through some energy scavenging device and use this energy to replenish the nodes batteries. In this scenario, known as Energy Harvesting Wireless Sensor Network (EHWSN), the main objective is to redesign the protocols and algorithms to properly manage the available energy preventing both the discharge of the batteries and the waste of energy due to excessively conservative approaches. The energy production process in EHWSN can be modeled as a random process, and the proposed networking / signal processing solutions should adapt to the statistical characteristic of this process.

In this thesis, after characterizing the gain that data compression can offer in energy constrained WSNs scenarios, and investigating the tradeoffs between compression and transmission energy for several lossy compression methods (see Chapter 4), we look at the optimization of the compression policy for a solar powered sensor node. In an EHWSN, in fact, the dynamics of the energy buffer provide additional challenges for the optimization of compression policies.

1.1 Structure of the thesis and main contributions

In the following we detail the thesis organization and the main contributions of each chapter.

Chapter 2 This chapter considers the problem of modeling space–time correlated signal and reproducing them. The models arising from the work in this chapter are used in Chapter 4 and Chapter 5 in order to obtain signals with desired characteristics for the performance evaluation of data compression schemes. In particular, the main contributions of the chapter are:

- the description of a method to generate synthetic signals with tunable correlation characteristics; with this method it is possible to extract the statistical characteristics of a signal from real datasets and use them to generate statistically similar signals;
- the validation of the proposed model through standard analytical techniques and through comparison against real and large data sets;

Chapter 3 This chapter presents a methodology and a tool to derive simple but accurate stochastic Markov processes for the description of the energy scavenged by outdoor solar sources. In particular, the main contributions of this chapter are:

- the description of a methodology to translate solar radiation data into the input energy process for a solar panel powered sensor node;
- the validation of the model against real datasets in terms of first– and second–order statistics.

Chapter 4 This chapter deals with the evaluation of a number of selected lossy compression methods from the literature, and the extensive analysis of their performance in terms of compression efficiency, computational complexity and energy consumption. In detail the main contributions of this chapter are:

- an in depth performance evaluation of selected lossy compression algorithms for time series, accounting for linear, autoregressive models, Fourier and Wavelet transforms. At first, we focus on interference-free single- and multi-hop networks, where the Medium Access Control (MAC) layer is idealized, i.e., besides transmission and reception, it does not introduce further energetic inefficiencies due to collisions and idle times for floor acquisition. For this scenario, we assess whether signal compression actually helps in the reduction of the overall energy consumption, depending on the compression algorithm, the chosen reconstruction fidelity, the signal statistics and the hardware characteristics;
- the derivation of formulas, obtained through numerical fittings and validated against real datasets, to gauge the computational complexity, the overall energy consumption and the signal representation accuracy of the best performing compression algorithms as a function of the most relevant system parameters. These

formulas can be used to generalize the results obtained here to other WSN architectures and will also be used in the optimization framework of Chapter 5 to describe the reward (related to the representation accuracy) and the cost (energy consumption) functions;

- the analytical characterization of a multi hop scenario along with the assessment of the performance improvements that are brought about by different lossy compression schemes in the presence of collisions and idle times for floor acquisition at the MAC.

Chapter 5 This chapter addresses the problem of optimizing the compression policy for an Energy Harvesting sensor node. The aim is to obtain the compression/transmission policy that jointly maximizes throughput and reconstruction fidelity at the data collection point, while meeting some predefined energy constraints, i.e., the battery charge level should never go below a predefined guard threshold. In addition, the impact of perfect vs delayed Channel State Information (CSI) at the transmitter and of power control over a multi-path channel is studied. The contributions of this chapter are:

- the definition, through a Constrained Markov Decision Problem (CMDP), of a model for the transmission and energy dynamics of a sensor node implementing practical lossy compression methods;
- the design of an algorithm to numerically evaluate optimal compression/transmission policies, using a Lagrangian relaxation approach combined with a dichotomic search for the optimal Lagrangian;
- the derivation of theoretical results on the structure of the optimal policy, demonstrating that under suitable but realistic assumptions it is non decreasing in each of the system state components;
- a thorough numerical evaluation of optimal as well as heuristic policies for different scenarios, gauging the impact of perfect vs delayed CSI and power control.

Chapter 6 This chapter concludes the thesis and discusses some avenues for future research work.

2

Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields

In the past few years, a large number of networking protocols for data gathering through aggregation, compression and recovery in Wireless Sensor Networks (WSNs) have utilized the spatio-temporal statistics of real world signals in order to achieve good performance in terms of energy savings and improved signal reconstruction accuracy. However, very little has been said in terms of suitable spatio-temporal models of the signals of interest. These models are very useful to prove the effectiveness of the proposed data gathering solutions as they can be used in the design of accurate simulation tools for WSNs. In addition, they can also be considered as reference models to prove theoretical results for data gathering algorithms. In this chapter, we present a mathematical model for real world signals that are correlated in space and time. We thus describe a method to reproduce synthetic signals with tunable correlation characteristics and we verify, through analysis and comparison against large data sets from real world testbeds, that our model is accurate in reproducing the signal statistics of interest. The model presented here will thus be exploited in Chapter 4 and Chapter 5 in order to generate the signals used in the simulations.

2.1 Introduction and Related Work

The temporal and spatial correlations are key statistical features of real signals, which are effectively exploited by many networking applications in the Wireless Sensor Network (WSN) domain. The temporal correlation captures the time evolution of the signal, making it possible to find appropriate sampling intervals for its accurate reconstruction. During these intervals, sensor nodes may go into a low power state, thus saving energy. The spatial correlation can instead be exploited in the deployment phase of, e.g., WSNs for environmental monitoring, to obtain suitable sensor densities as well as good deployment strategies [5]. Moreover, these features can be directly exploited in the design of networking protocols and signal compression techniques that make use of signal statistics. As an example, the authors of [6] design a distributed and collaborative Medium Access Control (MAC) protocol for WSNs that utilizes the spatial correlation of the monitored signal and exploits the fact that a sensor node can act as a representative node for other sensors in its neighborhood. [7] seeks to minimize the energy consumption of WSNs through the use of suitable spatio-temporal sampling rates. The objective of this work is to adapt the sleeping and spatial sampling behavior of the sensor nodes (readings from closely located sensors are almost equivalent when the signal is correlated in space) as a function of the signal statistics. The aim is to reduce the number of sensors that sample the signal per unit of time, while still allowing its accurate reconstruction. The problem of designing a data gathering tree over a WSN is addressed, e.g., in [8], where the authors exploit the spatial correlation of the signal to design a proper gathering tree. [9] presents a system for data handling in WSNs, which takes into account the spatio-temporal statistics of the signal, incorporating long-term storage, multi-resolution data access and spatio-temporal pattern mining.

In addition, recent techniques for in-network aggregation and distributed data compression in WSNs use theoretical tools such as Compressive Sensing (CS) [10, 11]. In CS, convex optimization techniques exploit the sparsity of the data to achieve distributed compression. As shown in [1, 12], the spatio-temporal statistics of the signal can be used to design an ad hoc sparsification basis that allows the effective use of CS for reconstruction through random sampling.

Although many approaches in the WSN literature utilized the spatio-temporal characteristics of real world signals, very little attention has been paid to the definition of simple yet accurate models, including lightweight, fast and accurate tools for the reproduction of signals with the desired spatio-temporal statistics. We believe that these models are instead very much needed to prove performance limits of data gathering and distributed signal processing solutions, as well as to carry out their performance evaluation systematically.

In this chapter, we address this gap by developing a framework to statistically characterize real world signals in space and time. This framework allows the accurate reproduction of the spatio-temporal behavior of such signals, obtaining synthetic models that can be effectively generated and subsequently used for protocol design and testing. These models are thus extensively validated against real world data, gathered from our indoor testbed at DEI [13], as well as climate data from [14]. The signal generation tool so obtained can be tuned to generate spatially and temporally correlated signals, where temporal and spatial correlations can be independently set. We remark that the usage of this tool is not limited to the field of WSN protocol optimization. As an example, the shadowing that affects radio transmissions can also be generated through our model. Note that in most previous works only the spatial correlation has been kept into account, see [15].

Previous work to generate spatially correlated signals for WSNs [16] accounted for the spatial correlation of real world signals through variogram functions, by however neglecting the temporal correlation. Similar approaches can also be found in the field of geostatistics [17]. However, the studies within this field are mostly centered around finding optimal predictors and interpolators for spatio-temporal varying signals, e.g., Kriged Kalman filtering [18], rather than giving simple, fast and accurate models for their generation. A further and very recent application is in cooperative cognitive radio sensing [19].

In this chapter, we describe a general approach that can be used in the aforementioned applications. In detail, our main contributions are the following:

- we describe a method to generate synthetic signals with tunable correlation characteristics; in this way it is possible to extract the statistical characteristics from real datasets and use them to generate statistically similar signals;
- we verify with standard analytical techniques and through comparison against real and large data sets that these characteristics are preserved;
- we provide a useful simulation tool that can be applied in all the above fields [20].

The rest of the chapter is organized as follows. In Section 2.2 we discuss the general correlation models for real world signals and introduce their properties. The model for space-time correlated signals is presented in detail in Section 2.3 and is validated against real world measurements in Section 2.4. Section 2.5 concludes our work.

2.2 Characterization of Signals from Sensor Network Fields

In this section we want to capture the relevant statistics from real environmental signals, in order to exploit this information for the generation of new samples with similar features. In the following, we present the details of our mathematical model for space-time correlated signals, introducing our notation, reported in Tab. I, as well as our basic assumptions. We focus on time varying two-dimensional (2D) fields of real-valued measurements; with x and y we indicate the space coordinates, whereas $\mathcal{D} = [-x_D, x_D] \times [-y_D, y_D]$ is the space domain. We consider that time is slotted where the slot time has a fixed duration $\Delta T > 0$. Thus, the time index is $t = i\Delta T$ with $i = 0, 1, 2, 3, \dots$, and the corresponding time domain is denoted by $\mathcal{T}$. A point in space $\boldsymbol{p} \in \mathcal{D}$ is indicated by the pair $\boldsymbol{p} = (x, y)$. With $z(\boldsymbol{p}, t) : \mathcal{D} \times \mathcal{T} \to \mathbb{R}$ we indicate the multidimensional random process that represents the space-time correlated random field (which is the objective of our analysis). When we fix a specific point with coordinates $\boldsymbol{p}_o$ in space and t_o in time, $z(\boldsymbol{p}_o, t_o)$ represents a random variable (r.v.) describing the value of this field at the specific point considered, with mean $\mu_z(\boldsymbol{p}_o, t_o)$ and variance $\sigma_z^2(\boldsymbol{p}_o, t_o)$. We assume that z is a stationary random process (weak-sense stationarity) in both space and time, so that $\mu_z(\boldsymbol{p}_o, t_o) = \mu_z$ and $\sigma_z^2(\boldsymbol{p}_o, t_o) = \sigma_z^2$, $\forall\, \boldsymbol{p} \in \mathcal{D}$ and $\forall\, t \in \mathcal{T}$. Moreover, we assume that the correlation function of the considered signal is separable in the temporal correlation and the spatial correlation:

$$\rho(\boldsymbol{p}_1, t_1, \boldsymbol{p}_2, t_2) = \rho_S(\boldsymbol{p}_1, \boldsymbol{p}_2)\rho_T(t_1, t_2)\,. \tag{2.1}$$

This is a general assumption used in the meteorology and geology fields, e.g., see [21, 22]. In detail, we consider the following models to study separately the temporal and spatial characteristics of the real signals.

2.2.1 Spatial correlation

$\rho_S(\boldsymbol{p}_1, \boldsymbol{p}_2)$ is the spatial correlation function between any two points $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{D}$. Formally, $\forall\, t \in \mathcal{T}$:

$$\rho_S(\boldsymbol{p}_1, \boldsymbol{p}_2) = \frac{\mathrm{cov}(z(\boldsymbol{p}_1, t), z(\boldsymbol{p}_2, t))}{\sigma_z(\boldsymbol{p}_1, t)\sigma_z(\boldsymbol{p}_2, t)}\,, \tag{2.2}$$

where $\mathrm{cov}(\cdot)$ is the covariance function. The weak-sense stationarity assumption made to define our multidimensional process implies also the correlation stationarity, i.e., for the spatial correlation, if the Euclidean distance $d_{1,2} = |\boldsymbol{p}_1 - \boldsymbol{p}_2|$ is equal to $d_{3,4} = |\boldsymbol{p}_3 - \boldsymbol{p}_4|$,

Symbol	Meaning
$\mathcal{D}$	Spatial domain
$\mathcal{T}$	Time domain
$\mathcal{F}$	Frequency domain
$\boldsymbol{p}$	Point in space $\boldsymbol{p} = (x, y) \in \mathcal{D}$
$\boldsymbol{\omega}$	Point in frequency $\boldsymbol{\omega} = (u, v) \in \mathcal{F}$
$z(\boldsymbol{p}, t)$	Space-time correlated field
$\rho_T(\Delta T)$	Temporal correlation coefficient of z
d	Euclidean distance between two points in $\mathcal{D}$
$\rho_S(d)$	Spatial correlation function of z between two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ with distance $d = |\boldsymbol{p}_1 - \boldsymbol{p}_2|$
$R_S(\boldsymbol{\omega})$	2D Fourier transform of $\rho_S(\boldsymbol{p})$
$w(\boldsymbol{p}, t)$	i.i.d. random Gaussian field at time t
$W(\boldsymbol{\omega}, t)$	2D Fourier transform of $w(\boldsymbol{p}, t)$
$\varepsilon(\boldsymbol{p}, t)$	i.i.d. random Gaussian noise field

Table 2.1. *Notation used in the analysis.*

then $\rho_S(\boldsymbol{p}_1, \boldsymbol{p}_2) = \rho_S(\boldsymbol{p}_3, \boldsymbol{p}_4)$. For this reason, the spatial correlation function can be defined as a function of the distance between two points, that is a scalar $d \in [0, d_M]$, where $d_M = \sqrt{(2x_D)^2 + (2y_D)^2}$, i.e., the maximum distance between two points in $\mathcal{D}$. In the following, with an abuse of notation we can write $\rho_S(\boldsymbol{p}_1, \boldsymbol{p}_2) = \rho_S(d_{1,2})$, without loss of generality.

Moreover, there are other issues to be considered when studying a real signal. Since the signal is sensed only at specific locations, we can not evaluate the process $z(\boldsymbol{p}, t)$ at every point $\boldsymbol{p} \in \mathcal{D}$, but we can observe the process only at the points where the signal is sensed, i.e., $\boldsymbol{p}_i \in \mathcal{D}$. This translates to the fact that also $\rho_S(d)$ may not be defined $\forall\, d \in [0, d_M]$. Thus, in the literature there exist many models to capture the spatial correlations for real signals [23]. We consider the following two models:

Power Exponential (PE) model: the spatial correlation $\rho_S(d)$ is modeled with the function:

$$\widetilde{\rho}_{S_{PE}}(d) = \exp\{-(d/\zeta)^{\nu}\}\ , \quad \text{for } 0 < \nu \leqslant 2\ ; \tag{2.3}$$

$$\begin{tikzcd}
w(\boldsymbol{p},t) \arrow[d, Rightarrow, "F"] & & r_S(\boldsymbol{p}) \arrow[d, Rightarrow, "F"] & & z(\boldsymbol{p},t) \\
W(\boldsymbol{\omega},t) & \times & R_S(\boldsymbol{\omega})^{\frac{1}{2}} & = & Z(\boldsymbol{\omega},t) \arrow[u, Rightarrow, "F^{-1}"']
\end{tikzcd}$$

Figure 2.1. *Diagram for the 2D filtering procedure.*

Rational Quadratic (RQ) model: the spatial correlation $\rho_S(d)$ is modeled with the function:

$$\widetilde{\rho}_{S_{RQ}}(d) = \frac{1}{1 + (d/\zeta)^{\nu} S_{\nu}} \ , \quad \text{for } 0 < \nu \leqslant 2 \ ; \tag{2.4}$$

Both models above depend on the parameters ζ and ν, that are the correlation length and the order of the function, respectively, while $S_\nu = 20^{1/\nu} - 1$ is a scaling factor. The two algorithms set ζ and ν to best fit the correlation of the real signal considered, that is defined only for a finite set of distances d_j, with $j = 1, \ldots, J$. The best fit is obtained minimizing the Root Mean Square Error (RMSE), i.e,

$$\xi_F = \sqrt{\frac{1}{J} \sum_{j=1}^{J} \left(\widetilde{\rho}_{S_M}(d_j) - \rho_S(d_j) \right)^2} \ , \tag{2.5}$$

where $F = PE$ for the PE model and $F = RQ$ for the RQ model.

2.2.2 Temporal correlation

$\rho_T(t_1, t_2)$ is the spatial correlation function between any two time samples $t_1, t_2 \in \mathcal{T}$. Formally, $\forall\, \boldsymbol{p} \in \mathcal{D}$:

$$\rho_T(t_1, t_2) = \frac{\operatorname{cov}(z(\boldsymbol{p}, t_1), z(\boldsymbol{p}, t_2))}{\sigma_z(\boldsymbol{p}, t_1)\sigma_z(\boldsymbol{p}, t_2)} \ . \tag{2.6}$$

Similar to the case of the spatial correlation, also in this case, as a consequence of the weak-sense stationarity assumption, the stationarity in the temporal correlation holds, i.e., we can write with an abuse of notation $\rho_T(t_1, t_2) = \rho_T(t_2 - t_1) = \rho_T(i\Delta T)$, where $i\Delta T = t_2 - t_1$. Since we aim at generating a synthetic signal with an autoregressive model, as will be detailed in the next section, we only consider the one step time correlation, i.e., we are only interested in calculating the coefficient $\rho_T(\Delta T)$.

2.3 Model for Space-Time Correlated Signals

Our objective is to devise a suitable and tractable model for the generation of the colored signal $z(\boldsymbol{p}, t)$. Specifically, we want to obtain a dynamic model, evolving at discrete time instants and thus allowing the generation of $z(\boldsymbol{p}, i\Delta T)$ for $i = 0, 1, 2, 3, \dots$. This model should retain, as much as possible, the correlation characteristics of the original signal that we sample from the sensor network field. Next, we present a suitable method for the generation of such signal, whereas its accuracy is demonstrated later on in Section 2.4 by comparing the generated signals against real data measurements.

The correlated signal $z(\boldsymbol{p}, t)$ is attained through the following procedure:

S1 At time $t_o = 0$, we generate an i.i.d. random Gaussian field $w(\boldsymbol{p}, t_o) : \mathcal{D} \times \mathcal{T} \rightarrow \mathbb{R}$, which for any specific location is a Gaussian r.v. with zero mean and unit variance, $\mathcal{N}(0, 1)$. $w(\boldsymbol{p}, t_o)$ is a stationary process (strict sense) by construction.

S2 $z(\boldsymbol{p}, t_o)$ is obtained by coloring $w(\boldsymbol{p}, t_o)$ through a 2D filtering procedure. In detail, we first obtain $W(\boldsymbol{\omega}, t_o)$ from $w(\boldsymbol{p}, t_o)$ using the 2D Fourier transform $F[\cdot]$, i.e., $W(\boldsymbol{\omega}, t_o) = F[w(\boldsymbol{p}, t_o)]$, where $\boldsymbol{\omega} = (u, v) \in \mathcal{F}$ and $\mathcal{F}$ is the frequency domain. Given the reference point $\boldsymbol{p}_o = (0, 0)$, for any point $\boldsymbol{p} \in \mathcal{D}$ we enforce the correlation $\rho_S(|\boldsymbol{p} - \boldsymbol{p}_o|)$ and compute its Fourier transform, $R_S(\boldsymbol{\omega}) = F[\rho_S(|\boldsymbol{p} - \boldsymbol{p}_o|)]$.

S3 Thus, we multiply $W(\boldsymbol{\omega}, t_o)$ by $R_S(\boldsymbol{\omega})^{1/2}$ and compute the inverse Fourier transform of the result so obtained to attain the wanted colored random field $z(\boldsymbol{p}, t_o)$. Note that z is still Gaussian and stationary. This procedure is sketched in the diagram of Fig. 2.1. This (S1–S3) is a known method for coloring a random Gaussian field, whose proof as well as further mathematical details can be found in [24]. The field z so obtained is correlated in space and its spatial correlation can be controlled through a *valid* correlation function $\rho_S(|\boldsymbol{p} - \boldsymbol{p}_o|)$ [23].

S4 In order to enforce a temporal correlation as well, we adopt an autoregressive filter as follows:

$$\begin{aligned} w(\boldsymbol{p}, t_o + \Delta T) &= \rho_T(\Delta T) w(\boldsymbol{p}, t_o) \\ &+ \sqrt{1 - \rho_T(\Delta T)^2} \varepsilon(\boldsymbol{p}, t_o + \Delta T) \,, \end{aligned} \tag{2.7}$$

where $\varepsilon(\boldsymbol{p}, t_o + \Delta T)$ is an i.i.d. random Gaussian noise, $\mathcal{N}(0, 1)$. $\rho_T(\Delta T)$ is a temporal correlation coefficient that we enforce in the model. Thus, the procedure S1–S4 is iterated for all future time steps to calculate $w(\boldsymbol{p}, t_o + i\Delta T)$ from $w(\boldsymbol{p}, t_o + (i-1)\Delta T)$, for $i = 1, 2, 3, \ldots$. Note that $\rho_T(\Delta T)$ is obtained from actual field measurements, as further discussed in Section 2.4. Also, note that $w(\boldsymbol{p}, t)$ is still i.i.d. in the space domain, whereas thanks to (2.7), it is time-correlated with correlation coefficient $\rho(\Delta T)$. As we demonstrate shortly, $w(\boldsymbol{p}, t)$ is again a stationary and Gaussian random process which maintains the same mean and variance of the original process $w(\boldsymbol{p}, t_o)$.

In what follows, we show that the autoregressive model that we superimpose to the spatially correlated signal preserves the statistical properties of the random field $w(\boldsymbol{p}, t)$, i.e., the autoregressive filtering is stable and the signal that it generates is time-stationary. To this end, we need to show that both mean and variance of $w(\boldsymbol{p}, t)$ are preserved using (2.7), and we do it using standard statistical techniques for continuous time processes, e.g., see [25].

Conservation of the mean of $w(\boldsymbol{p}, t)$. For the mean, we have that:

$$\mu_w(\boldsymbol{p}, t + \Delta T) = \mathbb{E}[w(\boldsymbol{p}, t + \Delta T)] = 0 \ , \tag{2.8}$$

that is obtained using (2.7) and the linearity of the expectation. It is easy to verify (2.8) for $t = t_o$, then the result is proven inductively for all time steps. Finally, note that this result holds $\forall\, \boldsymbol{p} \in \mathcal{D}$.

Conservation of the variance of $w(\boldsymbol{p}, t)$. For the variance, we have:

$$\begin{aligned} &\sigma_w^2(\boldsymbol{p}, t + \Delta T) = \\ &\qquad = \mathbb{E}[(w(\boldsymbol{p}, t + \Delta T) - \mu_w(\boldsymbol{p}, t + \Delta T))^2] = 1 \ , \end{aligned} \tag{2.9}$$

where we use the definition of autoregressive filter in (2.7) and a mathematical reasoning similar to the one used in (2.8). Again, the result in (2.9) holds for $t = t_o$, so we have that $\sigma_w^2(\boldsymbol{p}, t_o + \Delta T) = 1$ by construction, and the result $\forall\, t \in \mathcal{T}$ follows inductively.

Conservation of the temporal correlation of $z(\boldsymbol{p}, t)$. In what follows, we prove that the random process $z(\boldsymbol{p}, t)$ that we obtain through steps S1–S4 is also correlated in time and that its correlation coefficient is $\rho(\Delta T)$ at all time instants and for all points $\boldsymbol{p} \in \mathcal{D}$. Without loss of generality (the result also holds in the continuous case), we consider our spatial signal as being sampled from a rectangular sensor grid of $N \times M$ points evenly spaced in $\mathcal{D}$. Owing to this assumption, let the space point $\boldsymbol{p} = (x, y)$ be defined with $x \in \{1, 2, \ldots, N\}$ and $y \in \{1, 2, \ldots, M\}$. Also, let us define $r_S(\boldsymbol{p})$ as:

$$r_S(\boldsymbol{p}) = F^{-1}\left[F[\rho_S(|\boldsymbol{p} - \boldsymbol{p}_o|)]^{\frac{1}{2}}\right] = F^{-1}\left[R_S(\boldsymbol{\omega})^{\frac{1}{2}}\right] . \tag{2.10}$$

From this definition and the diagram of Fig. 2.1, we see that $z(\boldsymbol{p}, t)$ is obtained in the domain $\mathcal{D}$ as the spatial convolution between $w(\boldsymbol{p}, t) = w(x, y, t)$ and $r_S(\boldsymbol{p}) = r_S(x, y)$, i.e.,

$$z(\boldsymbol{p}, t) = \sum_{i=1}^{N}\sum_{j=1}^{M} w\,(i, j, t)\, r_S(x - i, y - j) , \tag{2.11}$$

where $i \in \{1, 2, \ldots, N\}$ and $j \in \{1, 2, \ldots, M\}$. We now compute the mean of process $z(\boldsymbol{p}, t)$, that is $\mu_z(\boldsymbol{p}, t) = \mathbb{E}[z(\boldsymbol{p}, t)]$:

$$\begin{aligned}\mu_z(\boldsymbol{p}, t) &= \mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{M} w(i, j, t) r_S(x - i, y - j)\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{E}\left[w(i, j, t)\right] r_S(x - i, y - j) = 0 \,,\end{aligned} \tag{2.12}$$

where the result follows from the linearity of the expectation, the fact that $r_S(\boldsymbol{p})$ is a deterministic function (as it is directly derived from the spatial correlation function, which is a known quantity) and the result in (2.8).

We are now ready to calculate the numerator of (2.6), which is given by (2.13): in this equation, equality (a) follows from (2.12), whereas (b) follows from (2.11) and (c) follows from (2.7). Also, (d) follows as the temporal correlation coefficient $\rho_T(\Delta T)$ and $r_S(\boldsymbol{p})$ are a constant and a deterministic function, respectively. For (e), note that $\mathbb{E}[w(i, j, t)\varepsilon(k, q, t + \Delta T)] = 0$, as process ε is independent of w by construction and its mean is zero. Moreover, when indices $i \neq k$ or $j \neq q$, we have $\mathbb{E}[w(i, j, t)w(k, q, t)] = \mathbb{E}[w(i, j, t)]\mathbb{E}[w(k, q, t)] = 0$, since $w(\boldsymbol{p}, t)$ is i.i.d. in the space domain and $\mathbb{E}[w(\boldsymbol{p}, t)] = 0$, $\forall\, \boldsymbol{p} \in \mathcal{D}$. Thus, for the only non zero case in which $i = k$ and $j = q$, we have $\mathbb{E}[w(i, j, t)w(k, q, t)] = \mathbb{E}[w(i, j, t)^2] = \sigma_w^2$.

The denominator of (2.6) is computed in (2.14). From (2.11), it follows that the variance of $z(\boldsymbol{p}, t)$ does not depend on t, since $w(\boldsymbol{p}, t)$ is stationary and $r_S(\boldsymbol{p})$ is a deterministic mapping function that only depends on the spatial correlation. Hence, equality (a) follows as $\sigma_{z(\boldsymbol{p},t)} = \sigma_{z(\boldsymbol{p},t+\Delta T)}$ and $\mu_z(\boldsymbol{p}, t) = 0$, see (2.12). The remaining equalities follow using the same reasonings as those done above for Eq. (2.13).

Taking the ratio of (2.13) and (2.14), we have proved the conservation of the temporal correlation, $\forall\, t \in \mathcal{T}$ and $\forall\, \boldsymbol{p} \in \mathcal{D}$.

$$
\begin{aligned}
&\mathrm{cov}(z(\boldsymbol{p},t), z(\boldsymbol{p},t+\Delta T)) \overset{(a)}{=} \mathbb{E}[z(\boldsymbol{p},t)z(\boldsymbol{p},t+\Delta T)]\\
&\overset{(b)}{=} \mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{M} w(i,j,t) r_S(x-i,y-j) \sum_{k=1}^{N}\sum_{q=1}^{M} w(k,q,t+\Delta T) r_S(x-k,y-q)\right]\\
&\overset{(c)}{=} \mathbb{E}\Bigg[\sum_{i=1}^{N}\sum_{j=1}^{M} w(i,j,t) r_S(x-i,y-j) \sum_{k=1}^{N}\sum_{q=1}^{M} \Big(\rho_T(\Delta T) w(k,q,t) + \\
&\qquad\qquad \sqrt{1-\rho_T(\Delta T)^2}\,\varepsilon(k,q,t+\Delta T)\Big) r_S(x-k,y-q)\Bigg]\\
&\overset{(d)}{=} \sum_{i,k=1}^{N}\sum_{j,q=1}^{M} \rho_T(\Delta T)\mathbb{E}\Big[w(i,j,t)w(k,q,t)\Big] + \\
&\qquad\qquad \sqrt{1-\rho_T(\Delta T)^2}\,\mathbb{E}\Big[w(i,j,t)\varepsilon(k,q,t+\Delta T)\Big] r_S(x-i,y-j) r_S(x-k,y-q)\\
&\overset{(e)}{=} \rho_T(\Delta T)\sigma_w^2 \sum_{i=1}^{N}\sum_{j=1}^{M} r_S(x-i,y-j)^2 \qquad (2.13)
\end{aligned}
$$

$$
\begin{aligned}
\sigma_z(\boldsymbol{p},t)\sigma_z(\boldsymbol{p},t+\Delta T) &\overset{(a)}{=} \mathbb{E}[z(\boldsymbol{p},t)^2]\\
&= \mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{M} w(i,j,t) r_S(x-i,y-j) \sum_{k=1}^{N}\sum_{q=1}^{M} w(k,q,t) r_S(x-k,y-q)\right]\\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{E}\left[w(i,j,t)^2\right] r_S(x-i,y-j)^2\\
&= \sigma_w^2 \sum_{i=1}^{N}\sum_{j=1}^{M} r_S(x-i,y-j)^2 \qquad (2.14)
\end{aligned}
$$

Discussion. The method presented in S1–S4 above allows one to obtain a stationary signal which is correlated in space and time according to an arbitrary spatial correlation function $\rho_S(\boldsymbol{p})$ and a temporal correlation coefficient $\rho_T(\Delta T)$. These parameters are tunable and can be fit to any real data set.

From (2.12) we see that the random process so obtained has zero mean; however, we observe that any non-zero mean can be enforced and all the results obtained here still hold.

Finally, from (2.14) and recalling that $\sigma_w^2 = 1$, we see that the variance of $z(\boldsymbol{p},t)$ equals $\sigma_z^2(\boldsymbol{p},t) = \sum_{i=1}^{N}\sum_{j=1}^{M} r_S(x-i,y-j)^2$. Thus, the variance follows directly from the spatial correlation properties of the signal. However, we can tune the amplitude of this variance through the multiplication of $z(\boldsymbol{p},t)$ by a constant. Also in this case the properties discussed above remain unchanged.

In the model above we do not keep into account the mean and variance characteristics of the signal. From our previous discussion, we see that these can be promptly accounted for by our model in case the application is also influenced by them.

2.4 Results

In this section we present the real datasets that we have used to validate our model. We want to show that the signal model of Section 2.3 can effectively capture the correlation characteristics of real signals. To this end, we first compute the correlation characteristics of the real data. These are thus used to tune the spatial correlation models of Section 2.2 and obtain the temporal correlation coefficient, which are subsequently used with the procedure of Section 2.3 to obtain the synthetic signals with the desired correlation properties. These are finally examined to check how well they can reproduce the correlation characteristics of the real data sets.

2.4.1 Real Signals analyzed

We hereby consider two types of real signals, both available online. The former is a dataset of global climate data from the Center for Climatic Research (CCR), Department of Geography, University of Delaware, available at [14]. This dataset is obtained interpolating the observations from about 10^4 climatic stations, and sampling uniformly the sensed field into a total of 85794 different points. The maximum distance between any two points is $d_M \simeq 2{\cdot}10^4$ km, while the time step is $\Delta T = 1$ month. Within the large set of data available in the website, we selected eight representative signals, namely, Air Temperature (CT), Total Precipitation (CP), EvapoTranspiration (CE), the Difference among Precipitation and EvapoTranspiration (CD), Snow Cover (CC), Snow Melt (CM), Soil Moisture (CS) and Moisture Indices (CI). For more information about these signals, see [14].

For the latter dataset, we considered a number of indoor signals gathered from the environmental monitoring WSN testbed deployed on the ground floor of the Department of Information Engineering (DEI), University of Padova, Italy [13]. The data from this dataset were sampled from $N = 68$ IEEE 802.15.4–compliant TelosB wireless nodes deployed according to an irregular topology, with maximum distance among any two sensors of $d_M \simeq 36$

m. The time step in this case is $\Delta T = 360$ s. These sensors can measure five different signals: temperature, humidity, luminosity in two different ranges ($\mathrm{DL}_1 : 320 - 730$ and $\mathrm{DL}_2 : 320 - 1100$ nm), and their battery voltage. For the performance analysis we considered temperature (DT) and humidity (DH), which have high spatial and temporal correlation, and luminosity in the two ranges (DL_1 and DL_2).

For each signal in both CCR and DEI databases, we considered about 500 time samples, we calculated the spatial correlation function using (2.2) and the one step temporal correlation coefficient using (2.6). Thus, we fit the spatial correlation of each signal through the PE model of (2.3) and the RQ model of (2.4). In the columns of Tab. II we show: 1. all the signals considered, 2-7. the parameters inferred and the values of the corresponding RMSE for the two correlation models (PE and RQ), and 8. the value of the temporal correlation $\rho_T(\Delta T)$ estimated from the real data. Boldface text is used in the table to indicate the best fitting model for each signal.

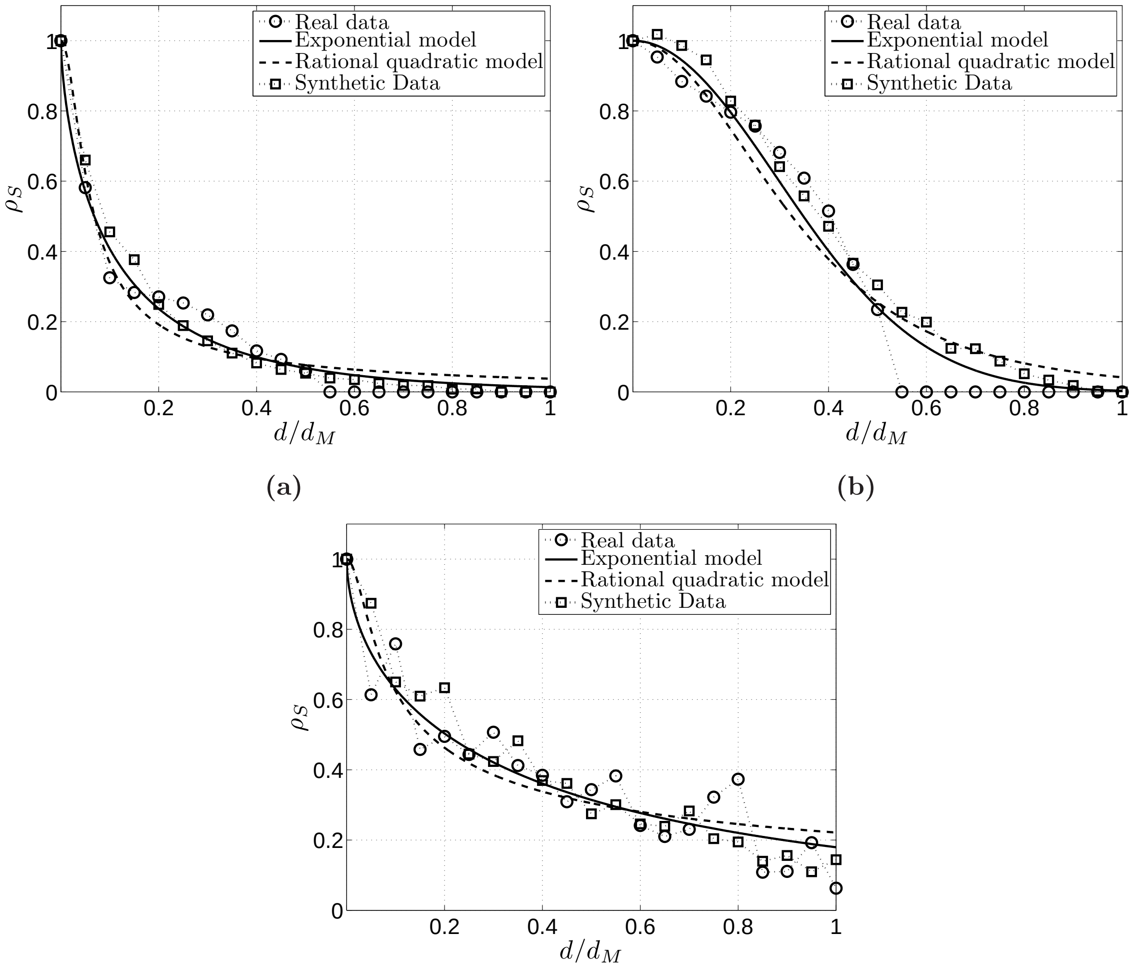
In order to calculate the spatial correlation (shown in Fig. 2.2 and discussed shortly), we had to consider each pair of points in the dataset. Since this number for the CCR dataset is very large, we picked at random only a subset of the total number of pairs (10^5 pairs, randomly selected, were considered for the results we present here), while for the DEI dataset we considered all possible pairs. For each pair of points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, we calculated the distance $d_{i,j}$ and the spatial correlation $\rho_S(d_{i,j})$, using (2.2). For representation purposes, we considered the maximum distance d_M and we divided it into 20 intervals. For each interval and for each pair of points whose distance falls within such interval, we calculated the average spatial correlation. In this way, we obtained one spatial correlation value for each interval (associating it with the center of the interval), similarly to the procedure adopted in [16] for the variogram calculation.

2.4.2 Model Validation

In Fig. 2.2 we show on the y-axis the spatial correlation $\rho_S(d)$ for three selected signals *vs* the normalized distance $d/d_M \in [0, 1]$. For comparison, in this figure we plot the empirical correlation obtained from the real data, the autocorrelation function obtained from the PE ($\widetilde{\rho}_{S_{PE}}(d)$) and RQ ($\widetilde{\rho}_{S_{RQ}}(d)$) models, as well as the spatial correlation obtained from the synthetic signal. Note that the synthetic signals are generated through the procedure of

Signal	Spatial Correlation						Temporal
	PE Model			RQ Model			Corr.
	ζ	ν	ξ_{PE}	ζ	ν	ξ_{RQ}	$\rho_T(\Delta T)$
CD	0,12	0,65	**0,042**	0,91	0,47	0,058	0,85
CP	0,11	0,68	**0,034**	0,71	0,52	0,050	0,90
CI	0,25	0,96	**0,046**	1,05	0,71	0,073	0,88
CM	0,06	0,53	0,040	0,49	0,51	**0,034**	0,27
CE	0,24	1,00	**0,057**	0,87	0,80	0,085	0,85
CT	0,42	2,00	**0,073**	0,94	2,00	0,103	0,98
CC	0,31	0,96	0,121	1,62	0,60	**0,117**	0,99
CS	0,05	0,62	0,022	0,33	0,60	**0,019**	0,94
DT	272	0,18	**0,119**	12916	0,13	0,146	0,99
DH	11640	0,31	**0,008**	9273	0,17	0,024	0,99
DL_1	0,38	0,56	**0,081**	25,46	0,23	0,095	0,98
DL_2	0,43	0,58	**0,081**	28,47	0,23	0,089	0,99

Table 2.2. *Correlation of Real signals*



(a)

(b)

(c)

Figure 2.2. *Spatial correlation for the real data, the two correlation fitting models (PE and RQ), and the correlation of the synthetic data generated according to the best fitting model, for the following signals: (a) global precipitation, CP; (b) global temperature, CT; and (c) indoor luminosity, DL_1.*

Section 2.3, using the best fitting model among PE and RQ, in terms of ξ_F. As shown in this figure, the chosen correlation models both nicely fit empirical correlation values. Moreover, the spatial correlation obtained from the synthetic signal is also very close to the empirical one.

In Figs. 2.2-(a) and 2.2-(b) we respectively show ρ_S for the Total Precipitation (CP) and the Air Temperature (CT) signals from the CCR dataset. In Fig. 2.2-(c) we plot ρ_S for the Luminosity signal (DL_1) from the DEI dataset. From these figures we see that our model

Signal	Model	ξ_F	ξ_S	$\Delta\rho_T$
CD	PE	0,042	0,021	3,92E-3
CP	PE	0,034	0,021	2,71E-3
CI	PE	0,046	0,030	1,18E-2
CM	RQ	0,034	0,035	3,82E-3
CE	PE	0,057	0,040	2,60E-3
CT	PE	0,073	0,035	2,16E-3
CC	RQ	0,117	0,069	2,14E-6
CS	RQ	0,019	0,031	2,60E-4
DT	PE	0,119	0,560	1,39E-4
DH	PE	0,008	0,542	1,16E-5
DL_1	PE	0,081	0,040	6,62E-4
DT_2	PE	0,081	0,072	2,24E-4

Table 2.3. *Correlation of Synthetic signals*

can very nicely fit the spatial characteristics of the real signals for both PE and RQ models. Furthermore, with the proposed method we are able to generate a synthetic signal that also follows with good accuracy the real signal correlation. Tab. 2.3 shows the results of the fitting of the synthetic signal for all the considered datasets. In particular, in the columns of this table we represent: 1. the considered signal, 2. the best fitting model among PE and RQ, 3. the corresponding RMSE ξ_F for the best model, 4. ξ_S, the RMSE between the chosen fitting model and the spatial correlation of the synthetic signal, 5. the relative error $\Delta\rho_T$ among the temporal correlation $\rho_T(\Delta T)$ and $\widetilde{\rho}_T(\Delta T)$, of the real and the synthetic signal, respectively. $\Delta\rho_T$ is calculated as follows:

$$\Delta\rho_T = \frac{|\rho_T(\Delta T) - \widetilde{\rho}_T(\Delta T)|}{\rho_T(\Delta T)} \ . \tag{2.15}$$

From Tab. 2.3, we observe that for all signals: (a) $\Delta\rho_T$ is sufficiently small, so we represent with high accuracy the temporal correlation, (b) both fitting models PE and RQ accurately reproduce the spatial correlation of the real signals. Furthermore, (c) the synthetic data very nicely follows the real spatial characteristics for all the CCR signals and for the DEI luminosity signals DL_1 and DL_2, (d) we introduce some error while representing the spatial characteristics for the DEI signals DT and DH. This is due to the fact that these two signals have a correlation length ζ, see Tab. 2.2, that is very large compared to the size of the network d_M. In this case, we are not able to reproduce with high accuracy the actual correlation of the signal due to border effects.

2.5 Conclusions

In this chapter we have presented a model for the statistical characterization of real world signals that are correlated in space and time. Our model allows the efficient generation of synthetic signals with the desired correlation properties, where spatial and temporal correlations can be independently set and fit to those of the real signals of interest. The accuracy of the proposed model has been verified through comparison against real data sets from large sensor testbeds. Future extensions of our work include the application of our statistical model to other types of signals, such as sensor data from smart grids, e.g., to model the distributed space and time dependent energy production and consumption process.

3

Modeling Photovoltaic Sources through Stochastic Markov Processes

In this chapter, we present a methodology and a tool to derive simple yet accurate stochastic Markov processes for the description of the energy scavenged by outdoor solar sources. In particular, we target photovoltaic panels with small form factors, as those exploited by embedded communication devices such as wireless sensor nodes or, concerning modern cellular system technology, by small-cells. Our models are especially useful for the theoretical investigation and the simulation of energetically self-sufficient communication systems including these devices.

The Markov models that we derive are obtained from extensive solar radiation databases, that are widely available online. Basically, from hourly radiance patterns, we derive the corresponding amount of energy (current and voltage) that is accumulated over time, and we finally use it to represent the scavenged energy in terms of its relevant statistics. Toward this end, two clustering approaches for the raw radiance data are described and the resulting Markov models are compared against the empirical distributions.

Our results indicate that Markov models with just two states provide a rough characterization of the real data traces. While these could be sufficiently accurate for certain applications, slightly increasing the number of states to, e.g., eight, allows the representation of the real energy inflow process with an excellent level of accuracy in terms of first and second order statistics.

3.1 Introduction

The use of renewable energy is very much desirable at every level of the society, from industrial / manufacturing activities to smart cities, public buildings, etc. Being able to capture any sort of renewable energy source is in fact very useful to power up, e.g., sensing equipment and electric apparatuses that surround us in our daily life, from automatic doors, to sensor systems for traffic control, intrusion detection, alarms, pollution reporting, etc. According to the paradigm of Smart Cities and the Internet of Things [26,27], these “resource constrained” small sensing devices are expected to be deployed massively. Of course, as an immediate advantage, self-sufficient (also referred to as *perpetual*) networks that will live unattended, just thanks to the energy they scavenge from the environment, would cut down their maintenance cost. Moreover, these systems will contribute to the reduction of the energy absorbed from the power grid (which is usually obtained from carbon fossil or nuclear power plants), thus benefiting the environment.

Similarly, Cellular Networks are expecting a tremendous grow of the traffic demand in the next years with a consequent increase in terms of cost and energy consumption. A challenging but promising solution is represented by the deployment of Base Stations (BSs) employing renewable energy sources [28]. Note, however, that the mere integration of a solar panel into existing electrical apparatuses, such as macro BSs, is often not sufficient as keeping these devices fully operational at all times would demand for unrealistically large solar modules [29]. To overcome this, the energy coming from the renewable sources should be wisely used, predicting future energy arrival and the energy consumption that is needed by the system to remain operational when needed. This calls for complex optimization approaches that will adapt the behavior of modern systems to the current application needs as well as to their energy reserves and the (estimated) future energy inflow [30].

A large body of work has been published so far to mathematically analyze these facts, especially in the field of wireless sensor networks. However, often researchers have tested their ideas considering deterministic [31,32], iid distributed across time slots [33] or time-correlated Markov models [34]. While these contributions are valuable for the establishment of the theory of energetically self-sufficient networks; seldom, the actual energy production process in these papers has been linked to that of real solar sources, to estimate the effectiveness of the proposed strategies under realistic scenarios.

The work presented in this chapter aims at filling this gap, by providing a methodology and a tool to obtain simple yet accurate stochastic Markov processes for the description of the energy scavenged by outdoor solar sources. In this study, we focus on solar modules as those that are installed in wireless sensor networks or small-LTE cells, by devising suitable Markov processes with first- and second-order statistics that closely match that of real data traces. Our Markov models allow the statistical characterization of solar sources in simulation and theoretical developments, leading to a higher degree of realism.

The rest of the chapter is organized as follows. In Section 3.2 we detail the system model and in particular how the raw radiance data is processed to estimate the corresponding instantaneous harvested power. This requires the combination of several building blocks, including an astronomical model (Section 3.2.1) to estimate the actual irradiance that hits the solar module, given the inclination of the sun during the day and the module placement, an electrical model of photovoltaic cells (Section 3.2.2) and a model for the DC/DC power processor (Section 3.2.3), which is utilized to maximize the amount of power that is collected. Hence, in Section 3.2.4 we describe the Markov model that we use to statistically describe the energy inflow, according to two clustering approaches for the raw data. The results from this Markov model are shown in Section 3.3, whereas our conclusions are presented in Section 3.4.

3.2 System Model

In this section, we describe the source model that we have adopted to statistically describe the energy inflow for a solar powered embedded device, see also [35]. To facilitate our description, we consider the diagram of Fig. 3.1 where we identify the key building blocks for our study: the solar source (indicated as I_{sun}), the photovoltaic panel (PV), the DC/DC power processor and the energy buffer (i.e., a rechargeable battery). In Section 3.2.1 we start with the characterization of the effective solar irradiance, I_{eff}, that in general depends on the geographical coordinates of the installation site, the season of the year and the hour of the day. Hence, I_{eff} is translated by the PV module into some electrical power and a DC/DC power processor is used to ensure that the maximum power is extracted from it.

3.2.1 Astronomical Model

The effective solar radiance that hits a photovoltaic module, I_{eff}, depends on physical factors such as its location, the inclination of the solar module, the time of the year and the hour of the day. Solar radiation databases are available for nearly all locations around the Earth and their data can be used to obtain the statistics of interest. An astronomical model is typically utilized to translate the instantaneous solar radiance I_{sun} (expressed in W/m^2) into the effective sunlight that shines on the solar module. According to [36], the effective solar radiance that hits the solar module, I_{eff}, is proportional to $\cos\Theta$, where $\Theta \in [-90^\circ, 90^\circ]$ is the angle between the sunlight and the normal to the solar module surface[1]. Astronomical models can be found in, e.g., [36] and Chapter 8 of [37].

In short, I_{eff} depends on many factors such as the elliptic orbit of the Earth around the sun (which causes a variation of the distance between Earth and sun across different seasons), the fact that the Earth is itself tilted on its axis at an angle of 23.45°. This gives rise to a *declination* angle γ, which is the angular distance North or South of the Earth's equator, which is obtained as:

$$\gamma(N) \simeq \sin^{-1}\left[\sin(23.45^\circ)\sin\left(D(N)\right)\right] , \tag{3.1}$$

where $D(N) = 360(N-81)/365^\circ$ and N is the day number in a year with first of January being day 1. Other key parameters are the *latitude* $La \in [0, 90^\circ]$ (positive in either hemisphere), the *longitude* Lo, the *hour angle* $\omega(t, N) \in [0, 360^\circ]$, that corresponds to the azimuth's angle of the sun's rays due to the Earth's rotation, the inclination β of the solar panel toward the sun on the horizon and the azimuthal displacement α, which is different from zero if the normal to the plane of the solar module is not aligned with the plane of the corresponding meridian, that is, the solar panel faces West or East.[2] $\omega(t, N)$ is given by $\omega(t, N) = 15(AST(t, N)-12)^\circ$, where $AST(t, N) \in [0, 24]$ hours, is the apparent solar time, which is the time based on the rotation of the Earth with respect to the sun and is obtained as a scaled version of the local standard time t (we refer to t' as t adjusted accounting for the daylight savings time) for the time zone where the solar module is installed. $AST(t, N)$ is computed as follows. Briefly, we obtain the Greenwich meridian angle, $GMA = UTC_{\text{off}} \times 15^\circ$, which corresponds to the angle between the Greenwich meridian and the meridian of the selected time zone: UTC_{off} is the time offset

[1] $\Theta = 0$ ($\Theta = \pm 90^\circ$) if the sunlight arrives perpendicular (parallel) to the module.

[2] $\alpha > 0$ if the panel faces West and $\alpha < 0$ if it faces East.

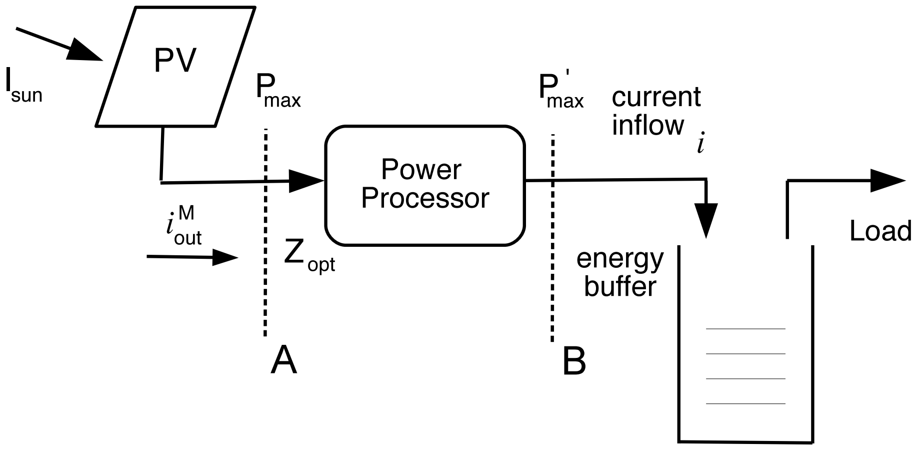


Figure 3.1. *Diagram of a solar powered device. The power processor adapts its input impedance so that it will match that of the source, Z_{opt}. This allows the extraction of the maximum power $P_{\max}$.*

between Greenwich and the time zone and 15 is the rotation angle of the Earth per hour. Thus, we compute $\Delta t = (Lo - GMA)/15^\circ$, i.e., the time displacement between the selected time zone and the time at the reference Greenwich meridian. At this point, $AST(t, N)$ is obtained as $AST(t, N) = t' + \Delta t + ET(N)$ (expressed in hours), where $ET(N)$ is known as the *equation of time*, with $ET(N) \simeq [9.87 \sin(2D(N)) - 7.53 \cos(D(N)) - 1.5 \sin(D(N))]/60$.

Finally, the power incident on the PV module depends on the angle Θ, for which we have:

$$\begin{aligned}\cos\Theta(t,N) &= \sin\gamma(N)\sin La\cos\beta - \\ &- \sin\gamma(N)\cos La\sin\beta\cos\alpha + \\ &+ \cos\gamma(N)\cos La\cos\beta\cos\omega(t,N) + \\ &+ \cos\gamma(N)\sin La\sin\beta\cos\alpha\cos\omega(t,N) \\ &+ \cos\gamma(N)\sin\beta\sin\alpha\sin\omega(t,N)\,. \qquad (3.2)\end{aligned}$$

Once an astronomical model is used to track Θ, the effective solar radiance as a function of time t is given by: $I_{\text{eff}}(t, N) = I_{\text{sun}}(t, N) \max(0, \cos\Theta(t, N))$, where the $\max(\cdot)$ accounts for the cases where the solar radiation is above or below the horizon, as in these cases the sunlight arrives from below the solar module and is therefore blocked by the Earth. The sun radiance, $I_{\text{sun}}(t, N)$, for a given location, time t and day N, has been obtained from the database at [38].

3.2.2 PV Module

A PV module is composed of a number n_{sc} of *solar cells* that are electrically connected according to a certain configuration, whereby a number n_{p} of them are connected in parallel and n_{s} in series, with $n_{\mathrm{sc}} = n_{\mathrm{p}} n_{\mathrm{s}}$. A given PV module is characterized by its I-V curve, which emerges from the composition of the I-V curves of the constituting cells. Specifically, the I-V curve of the single solar cell is given by the superposition of the current generated by the solar cell diode in the dark with the so called *light-generated* current i_ℓ [39], where the latter is the photo-generated current, due to the sunlight hitting the cell. The I-V curve of a solar cell can be approximated as:

$$i_{\mathrm{out}} \simeq i_\ell - i_{\mathrm{o}} \left[\exp\left(\frac{qv}{nkT} \right) - 1 \right] , \tag{3.3}$$

where q is the elementary charge, v is the cell voltage, k is the Boltzmann's constant, T is the temperature in degree Kelvin[3], $n \geqslant 1$ is the diode ideality factor and i_{o} is the *dark saturation current*. i_o corresponds to the solar cell diode leakage current in the absence of light and depends on the area of the cell as well as on the photovoltaic technology. The open circuit voltage v_{oc} and the short circuit current i_{sc} are two fundamental parameters for a solar cell. The former is the maximum voltage for the cell and occurs when the net current through the device is zero. i_{sc} is instead the maximum current and occurs when the voltage across the cell is zero (i.e., when the solar cell is short circuited). If $v_{\mathrm{oc}}^{\mathrm{M}}$ and $i_{\mathrm{sc}}^{\mathrm{M}}$ are the open circuit voltage and short circuit current for a solar module M, the single solar cell parameters are obtained as: $i_{\mathrm{sc}} = i_{\mathrm{sc}}^{\mathrm{M}}/n_{\mathrm{p}}$ and $v_{\mathrm{oc}} = v_{\mathrm{oc}}^{\mathrm{M}}/n_{\mathrm{s}}$ (considering a module composed of homogeneous cells).

The light-generated current for the single solar cell is a time varying quantity, $i_\ell(t, N)$, which depends on the amount of sunlight that hits the solar cell at time t, where N is the day number. Here, we have used the following relation: $i_\ell(t, N) = i_{\mathrm{sc}} F(t, N)$, where the *radiation rate* $F(t, N) \in [0, 1]$ is obtained as $F(t, N) = 0.001 I_{\mathrm{eff}}(t, N)$, i.e., normalizing the effective irradiance hitting the solar cell with respect to the maximum radiation of 1 kW/m^2 (referred to in the literature as "one sun" [40]). Hence, $i_\ell(t, N)$ is plugged into (3.3) to obtain $i_{\mathrm{out}}(t, N)$ for a single solar cell as a function of the time t for day N. The total current that is extracted from the solar module is: $i_{\mathrm{out}}^{\mathrm{M}}(t, N) = n_{\mathrm{p}} i_{\mathrm{out}}(t, N)$.

[3] T is given by the sum of the ambient temperature, which can be obtained from the dew point and relative humidity, and of a further factor due to the solar power hitting the panel.

3.2.3 Power Processor

Generally speaking, every voltage or current source has a *maximum power point*, at which the average power delivered to its load is maximized. For example, a Thévenin voltage source delivers its maximum power when operating on a resistive load whose value matches that of its internal impedance. However, in general the load of a generic device does not match the optimal one, which is required to extract the maximum power from the connected solar source. To cope with this, in practice the optimal load is emulated through a suitable *power processor*, whose function is that of "adjusting" the source voltage (section A of Fig. 3.1) until the power extracted from it is maximized,[4] which is also known as maximum power point tracking (MPPT). Ideally, through MPPT, the maximum output power is extracted from the solar panel under any given temperature and irradiance condition, adapting to changes in the light intensity. Commercially available power processors use "hill climbing techniques"; as an example, in [41] the authors propose advanced control schemes based on the downhill simplex algorithm, whereby the voltage and the switching frequency are jointly adapted for fast convergence to the maximum power point. See also [42] for further information on MTTP algorithms and their comparative evaluation and [43] for a low-power design targeted to wireless sensor nodes. In the present work, we have taken into account the DC/DC power processor by computing the operating point $(i_{\mathrm{out}}^{\mathrm{M}}, v^{\mathrm{M}})$ (see (3.3)) for which the extracted power in section A, $P = i_{\mathrm{out}}^{\mathrm{M}} v^{\mathrm{M}}$, is maximized. Note that, if i_{out} and v are the output current and the voltage of the single solar cell, we have $i_{\mathrm{out}}^{\mathrm{M}} = n_{\mathrm{p}} i_{\mathrm{out}}$ and $v^{\mathrm{M}} = n_{\mathrm{s}} v$. For this procedure, we have considered the parameters of Section 3.2.1 and 3.2.2 (solar irradiance, rotation of the Earth, etc.) and also the fact that i_{sc} and v_{oc} change as a function of the environmental temperature, which affects the shape of the I-V curve (3.3) (see, e.g., the dependence of i_ℓ on i_{sc}). Hence, we have computed the extracted power in two steps: step 1) we have obtained the (ideal) maximum power P_{MPP} that would be extracted by the panel at the MPP by an ideal tracking system:

$$P_{\mathrm{MPP}} = \max_{v}\{i_{\mathrm{out}}^{\mathrm{M}} v^{\mathrm{M}}\} = n_{\mathrm{p}} n_{\mathrm{s}} \max_{v}\{i_{\mathrm{out}} v\}\,, \tag{3.4}$$

where i_{out} is given by (3.3). Step 2) the power available after the power processor (section B in Fig. 3.1) is estimated as $P'_{\max} = \eta P_{\mathrm{MPP}}$, where $\eta \in (0, 1)$ is the power processor conversion

[4] This corresponds to adapting the input impedance of the power processor to $Z_{\mathrm{opt}} = Z_{\mathrm{source}}^{*}$, where * indicates the complex conjiugate.

efficiency, which is usually defined as the ratio $P'_{\max}/P_{\text{MPP}}$ and can be experimentally characterized for a given MPP tracking circuitry [43]. $P'_{\max}$ is the power that is finally transferred to the energy buffer.

3.2.4 Semi-Markov Model for Stochastic Energy Harvesting

The dynamics of the energy harvested from the environment is captured by a continuous time Markov chain with $N_{\text{s}} \geqslant 2$ states. This model is general enough to accommodate different clustering approaches for the empirical data, as we detail shortly.

Formally, we consider an energy source that, at any given time, can be in any of the states $x_s \in \mathcal{S} = \{0, 1, \ldots, N_{\text{s}} - 1\}$. We refer to t_k, with $k \geqslant 0$, as the time instants where the source transitions between states, and we define $\tau_k = t_{k+1} - t_k$ as the time elapsed between two subsequent transitions. In what follows, we say that the system between t_k and t_{k+1} is in *cycle* k.

Right after the k-th transition to state $x_s(k)$, occurring at time t_k, the source remains in this state for τ_k seconds, where τ_k is governed by the probability density function (pdf) $f(\tau|x_s)$, with $\tau \in [\tau_{\min}(x_s), \tau_{\max}(x_s)]$. At the next transition instant, t_{k+1}, the source moves to state $x_s(k+1) \in \mathcal{S}$ according to the probabilities $p_{uv} = \text{Prob}\{x_s(k+1) = v | x_s(k) = u\}$, with $u, v \in \mathcal{S}$. When the source is in state $x_s(k)$, an input current i_k is fed to the rechargeable battery, where i_k is drawn from the pdf $g(i|x_s)$, with $i \geqslant 0$. That is, when a state is entered, the input current i and the permanence time τ are respectively drawn from $g(i|x_s)$ and $f(\tau|x_s)$. Then, the input current remains constant until the next transition, that occurs after τ seconds. In this work, we assume that the voltage at the energy buffer (section B of Fig. 3.1) is constant, as typically considered when a rechargeable battery is used. Given that, there is a one-to-one mapping between instantaneous harvested power and harvested current.

3.2.5 Estimation of Energy Harvesting Statistics

Based on our models of Sections 3.2.1-3.2.3, we have mapped the hourly irradiance patterns obtained from [38] into the corresponding operating point, in terms of power $P'_{\max}$ and current i after the power processor (section B of Fig. 3.1). Thus, we have computed the statistics $f(\tau|x_s)$ and $g(i|x_s)$ of Section 3.2.4 from these data according to the two approaches that we describe next. These differ in the adopted clustering algorithm, in the number of states N_s and in the structure of the transition probabilities p_{uv}, $u, v \in \mathcal{S}$.

Night-day clustering: we have collected all the data points in [38] from 1991 to 2010 and grouping them by month. Thus, for each day in a given month we have classified the corresponding points into two states $x_s \in \{0, 1\}$, i.e., a low- ($x_s = 1$) and a high-energy state ($x_s = 0$). To do this, we have used a current threshold i_th, which is a parameter set by the user, corresponding to a small fraction of the maximum current in the dataset. According to the resulting value of i_th, we have classified all the points that fall below that threshold as belonging to state 0 (i.e., night) and those points above the threshold as belonging to state 1 (day). After doing this for all the days in the dataset, we have estimated the probability density function (pdf) of the duration τ, $f(\tau|x_s)$, and that of the input current i (after the power processor), $g(i|x_s)$, for each state and for all months of the year. For the estimation of the pdfs we have used the kernel smoothing technique see, e.g., [44]. The transition probabilities of the resulting semi-Markov chain are $p_{10} = p_{01} = 1$ and $p_{00} = p_{11} = 0$ as a night is always followed by a day and vice versa.

Slot-based clustering: as above, we have collected and classified the irradiance data by month. Then, we subdivided the 24 hours in each day into a number $N_\mathrm{s} \geqslant 2$ of time slots of constant duration, equal to T_i hours, $i = 1, \dots, N_\mathrm{s}$. Each slot is a state x_s of our Markov model. Hence, for each state x_s we computed the pdf $g(i|x_s)$ for each month of the year, considering the empirical data that has been measured in slot x_s for all days in the dataset for the month under consideration. Again, the kernel smoothing technique has been utilized to estimate the pdf. For the statistics $f(\tau|x_s)$, being the slot duration constant by construction, we have that: $f(\tau|x_s) = \delta(\tau - T_{x_s})$, for all states $x_s \in \mathcal{S}$, where $\delta(\cdot)$ is the Dirac's delta. The transition probabilities of the resulting Markov chain are $p_{uv} = 1$, when $u \in \mathcal{S}$ and $v = (u + 1)$ mod N_s, and $p_{uv} = 0$ otherwise. This reflects the temporal arrangement of the states.

3.3 Numerical Results

For the results in this section, we have used as reference the commercially available micro-solar panels from Solarbotics, selecting the Solarbotics's SCC-3733 Monocrystalline solar

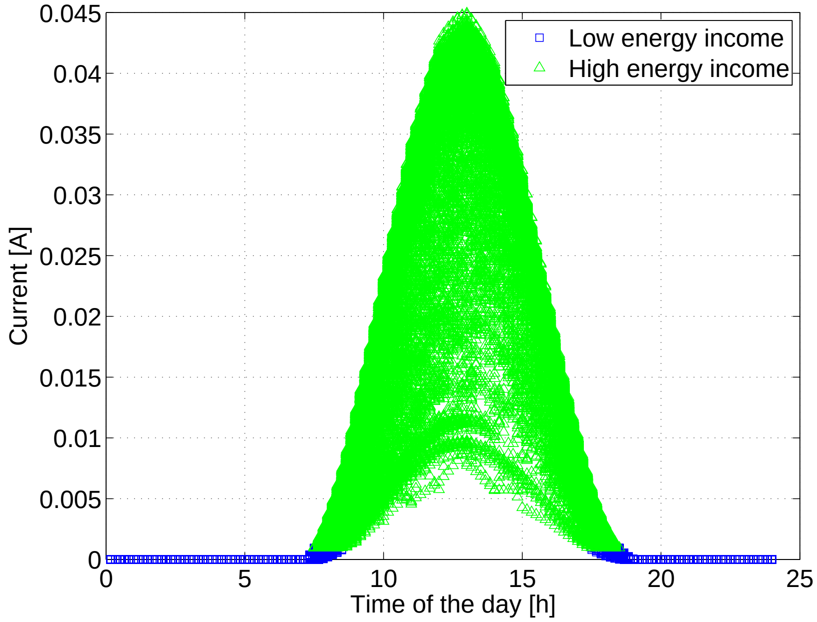


Figure 3.2. *Result of the night-day clustering approach for the month of July considering the radiance data from years* $1999-2010$*.*

technology [45]. For this product, the single cell area is about 1 square centimeters, the solar cells have an efficiency of 21.1%, $i_{\rm sc} = 5$ mA and $v_{\rm oc} = 1.8$ V. For the DC/DC power processor we have set $\eta = 0.5$ and $V_{\rm ref} = 3$ V, which are typical values for embedded sensor nodes, see [43] and [46]. Next, we show some results on the stochastic model for the solar energy source of Section 3.2. These are obtained considering a solar module with $n_{\rm p} = 6$ and $n_{\rm s} = 6$ cells in parallel and in series, respectively. We have selected Los Angeles as the installation site, considering $\beta = 45^\circ$, $\alpha = 30^\circ$ and processing the data from [38] as described in the previous section with a cluster threshold equal to $1/50-$th of the maximum value of the current in the dataset.

Night-day clustering: a first example for the night-day clustering approach is provided in Fig. 3.2, which shows the result of the clustering process for the month of July. Two macro states are evident: a low energy state (night), during which the power inflow is close to zero, and a high energy state (day). As this figure shows, the harvested current during the day follows a bell-shaped curve. However, contrarily to what one would expect, even for the month of July the high-energy state shows a high degree of variability from day-to-day, as is

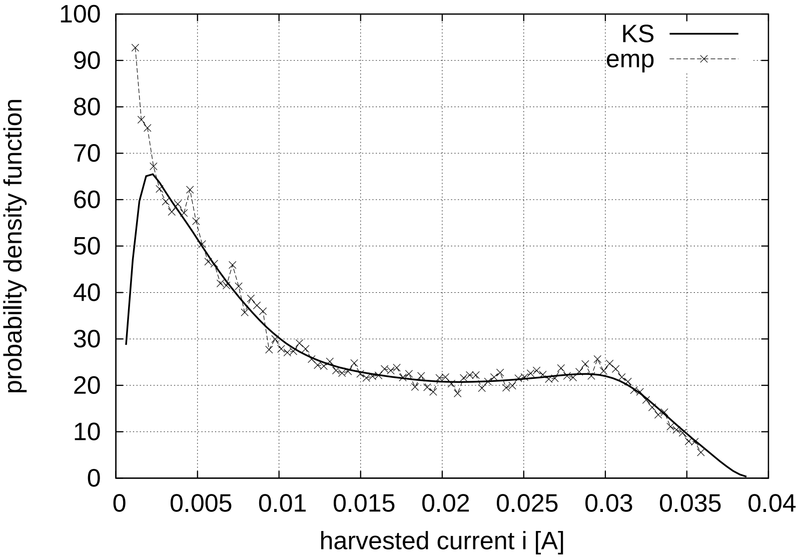


Figure 3.3. *$g(i|x_s)$ (solid line, $x_s = 0$) obtained through the Kernel Smoothing (KS) technique for the month of February, for the night-day clustering method (2-state semi-Markov model), using radiance data from years $1999 - 2010$. The empirical pdf (emp) is also shown for comparison.*

testified by the considerable dispersion of points across the y-axis. This reflects the variation in the harvested current due to diverse weather conditions. In general we have a twofold effect: (i) for different months the peak of the bell varies substantially, e.g., from winter to summer and (ii) for a given month the variability across the y-axis remains among different days. These facts justify the use of stochastic modeling, as we do in this work, to capture such variability in a statistical sense.

Another example, regarding the accuracy of the Kernel Smoothing (KS) technique to fit the empirical pdfs, is provided in Fig. 3.3, where we show the fitting result for the month of February.

In Figs. 3.4 and 3.5 we show some example statistics for the months of February, July and December. In Fig. 3.4, we plot the pdf $g(i|x_s)$, which has been obtained through the Kernel Smoothing (KS) technique for the high-energy state $x_s = 0$. As expected, the pdf for the month of July has a larger support and has a peak around $i = 0.04$ A, which means that is likely to get a high amount of input current during that month. For the months of February and December, we note that their supports shrink and the peaks move to the left to about 0.03 A and 0.022 A, respectively, meaning that during these months the energy scavenged is lower and is it more likely to get a small amount of harvested current during the day. Fig. 3.5 shows the cumulative distribution functions (cdf) obtained integrating $g(i|x_s)$ and also the corresponding empirical cdfs. From this graph we see that the cdfs obtained through KS closely match the empirical ones. In particular, all the cdfs that we have obtained through KS have passed the Kolmogorov-Smirnov test when compared against the empirical ones, for a confidence of 1%, which confirms that the obtained distributions represent a good model for the statistical characterization of the empirical data. The pdf for state $x_s = 1$ is not shown as it has a very simple shape, presenting a unique peak around $i = 0^+$. In fact, the harvested current is almost always negligible during the night.[5] Figs. 3.6 and 3.7 respectively show the pdf $f(\tau|x_s)$ obtained through KS and the corresponding cdf for the same location and months of above, for $x_s = 0$. Again, Fig. 3.6 is consistent with the fact that in the summer days are longer and Fig. 3.7 confirms the goodness of our KS estimation. Also in this case, the statistics for all months have passed the Kolmogorov-Smirnov test for a confidence of 1%. The pdfs for state $x_s = 1$ are not shown as these are specular to those of Fig. 3.6 and this is also to be expected as the sum of the duration of the two states $x_s = 0$ (daytime) and $x_s = 1$ (night) corresponds to the constant duration of a day. This means that the duration statistics of one state is sufficient to derive that of the other.

[5] Note that our model does not account for the presence of external light sources such as light poles.

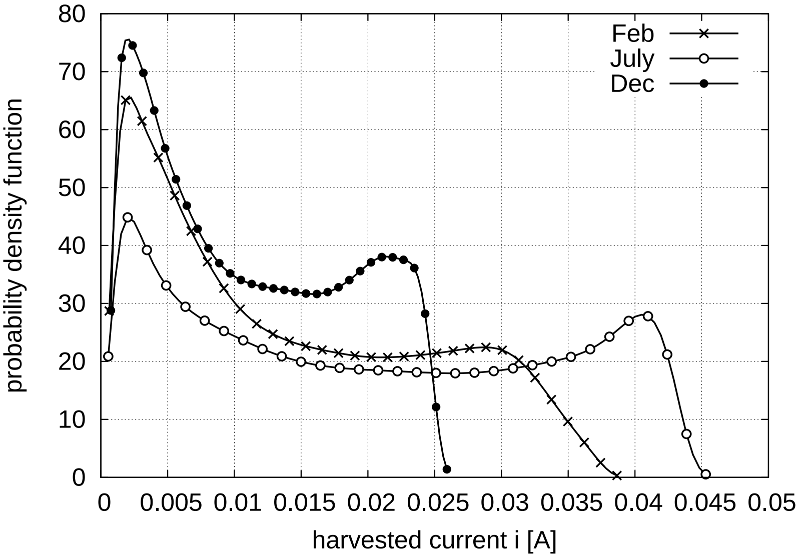


Figure 3.4. *Probability density function $g(i|x_s)$, for $x_s = 1$, obtained through Kernel Smoothing for the night-day clustering method (2-state Markov model).*

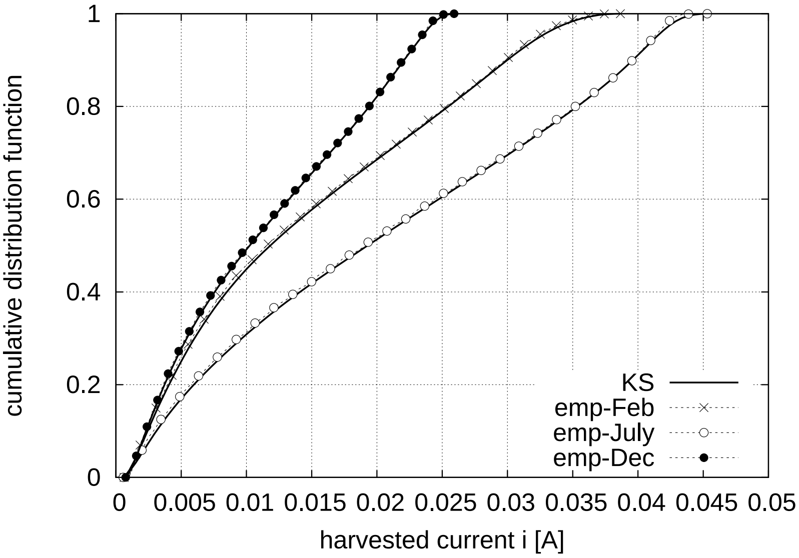


Figure 3.5. *Cumulative distribution function of the harvested current for $x_s = 1$ (solid lines), obtained through Kernel Smoothing (KS) for the night-day clustering method (2-state Markov model). Empirical cdfs (emp) are also shown for comparison.*

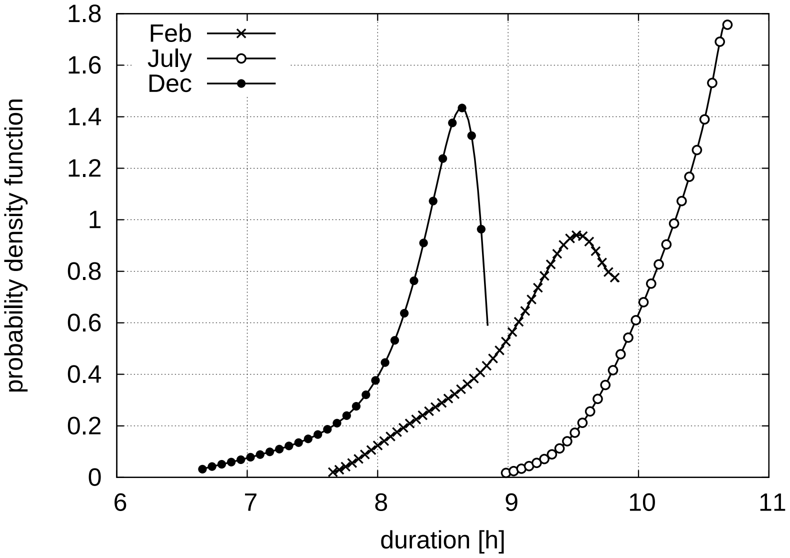


Figure 3.6. *Probability density function $f(\tau|x_s)$, for $x_s = 1$, obtained through Kernel Smoothing for the night-day clustering method (2-state Markov model).*

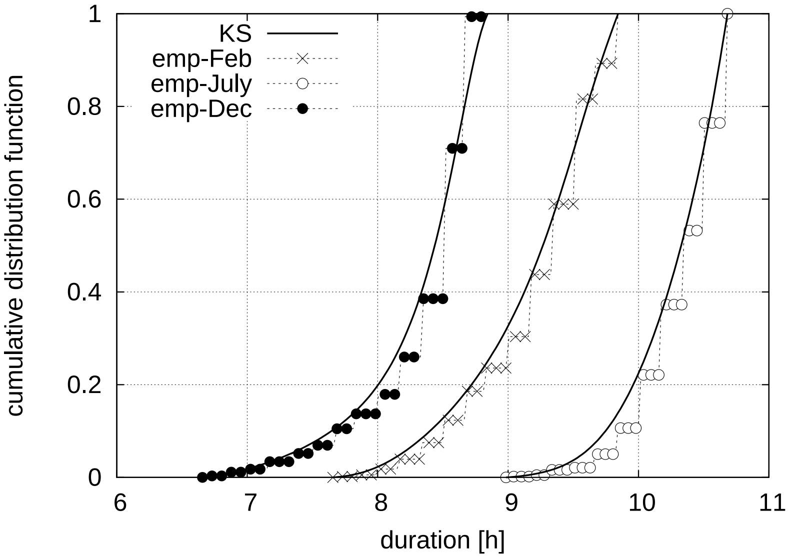


Figure 3.7. *Cumulative distribution function of the state duration for $x_s = 1$ (solid lines), obtained through Kernel Smoothing (KS) for the night-day clustering method (2-state Markov model). Empirical cdfs (emp) are also shown for comparison.*

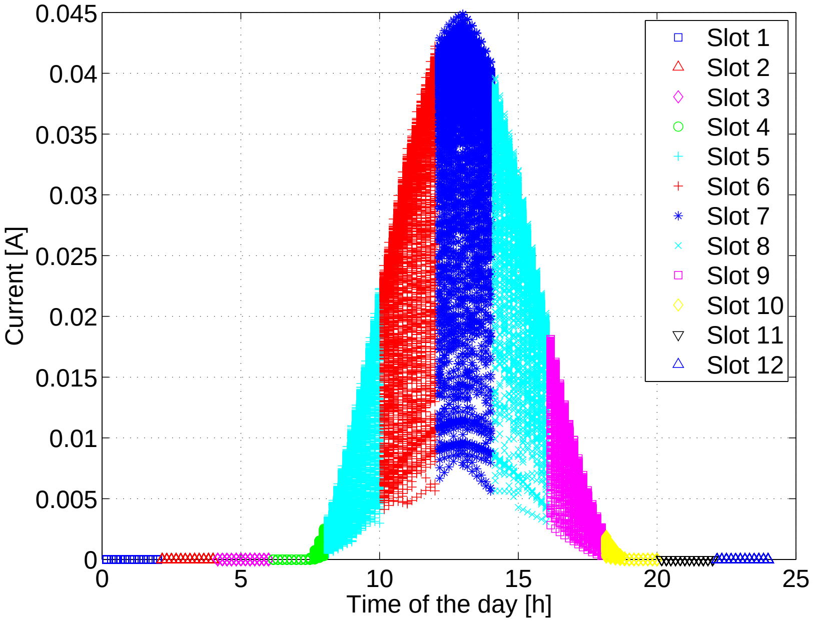


Figure 3.8. *Result of slot-based clustering considering* 12 *time slots (states) for the month of July, years* 1999 – 2010*.*

Slot-based clustering: the attractive property of the 2-state semi-Markov model obtained from the night-day clustering approach is its simplicity, as two states and four distributions suffice to statistically represent the energy inflow dynamics. Nevertheless, this model leads to a coarse-grained characterization of the temporal variation of the harvested current, especially in the high-energy state.

Slot-based clustering has been devised with the aim of capturing finer details. An example of the clustering result for this case is given in Fig. 3.8, for the month of July. All slots in this case have the same duration, which has been fixed a priori and corresponds to $24/N_s$ hours.

Fig. 3.9 shows the pdf $g(i|x_s)$ for the first three states of the day (slots $5, 6$ and 7, see Fig. 3.8) for the month of July, which have been obtained through KS. As expected, the peaks (and the supports) of the pdfs move to higher values, until reaching the maximum of 0.04 A for slot 7, which is around noon. Due to the symmetry in the solar distribution within the day, the results for the other daytime states are similar and therefore have not been reported. In Fig. 3.10 we compare the cdfs obtained through KS against the empirical ones. Also in this case, all the cdfs have passed the Kolmogorov-Smirnov test for a confidence of 1%.

A last but important results is provided in Fig. 3.11, where we plot the autocorrelation

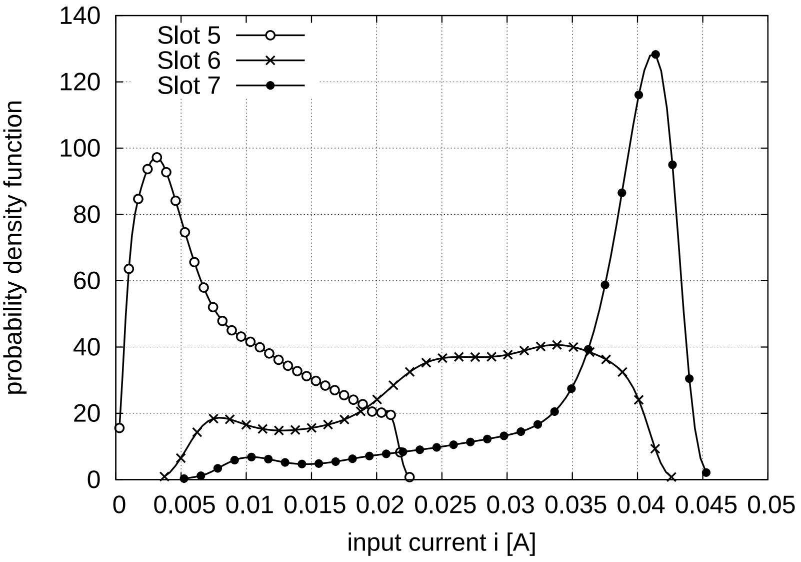


Figure 3.9. *Probability density function (pdf) $g(i|x_s)$ for $x_s = 5, 6$ and 7 for the slot-based clustering method for the month of July.*

function (ACF) for the empirical data and the Markov processes obtained by slot-based clustering for a number of states N_s ranging from 2 to 24 for the month of January. With the ACF we test how well the Markov generated processes match the empirical data in terms of second-order statistics. As expected, a 2-state Markov model poorly resembles the empirical ACF, whereas a Markov process with $N_\mathrm{s} = 12$ states performs quite satisfactorily. Note also that 5 of these 12 states can be further grouped into a single macro-state, as basically no current is scavenged in any of them (see Fig. 3.8). This leads to an equivalent Markov process with just eight states.

We highlight that our Markov approach keeps track of the temporal correlation of the harvested energy within the same day, though the Markovian energy generation process is independent of the "day type" (e.g., sunny, cloudy, rainy, etc.) and also on the previous day's type. Given this, one may expect a good fit of the ACF within a single day but a poor representation accuracy across multiple days. Instead, Fig. 3.11 reveals that the considered Markov modeling approach is sufficient to accurately represent second-order statistics. This has been observed for all months. On the other hand, one may be thinking of extending the state space by additionally tracking good (g) and bad (b) days so as to also model the temporal correlation associated with these qualities. This would amount to defining a Markov chain with the two macro-states g and b, where $p_{gb} = \mathrm{Prob}\{\text{day } k \text{ is } g | \text{ day } k-1 \text{ is } b\}$, with

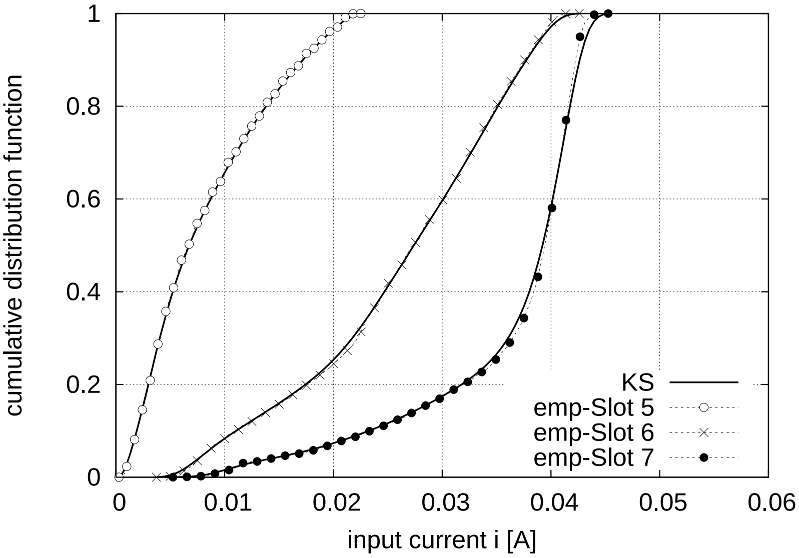


Figure 3.10. *Comparison between KS and the empirical cdfs (emp) of the scavenged current for $x_s = 5, 6$ and 7 for the slot-based clustering method for the month of July.*

$k \geqslant 1$. Hence, in each state g or b, the energy process could still be tracked according to one of the two clustering approaches of Section 3.2.4, where the involved statistics would be now conditioned on being in the macro-state. The good approximation provided by our model, see Fig. 3.11, show that this further level of sophistication is unnecessary.

Panel size and location: to conclude, we show some illustrative results for different solar panel sizes and locations. Table 3.1 presents the main outcomes for different solar cells configurations for the night-day clustering approach. Two representative months are considered: the month with the highest energy harvested, August, and the one with the lowest, December. As expected, the current inflow strongly depends on the panel size (linearly). Also, note that the day duration slightly increases for an increasing panel area as this value is obtained by measuring when the energy is above a certain (clustering) threshold. Although we scaled this threshold proportionally with an increasing harvested current, the longer duration of the day is due to the exponential behavior introduced by the scaling factor in (3.3), see the RHS of this equation.

Finally, in table 3.2 we show some energy harvesting figures for a solar panel with $n_{\rm p} = n_{\rm s} = 6$ for some representative cities.

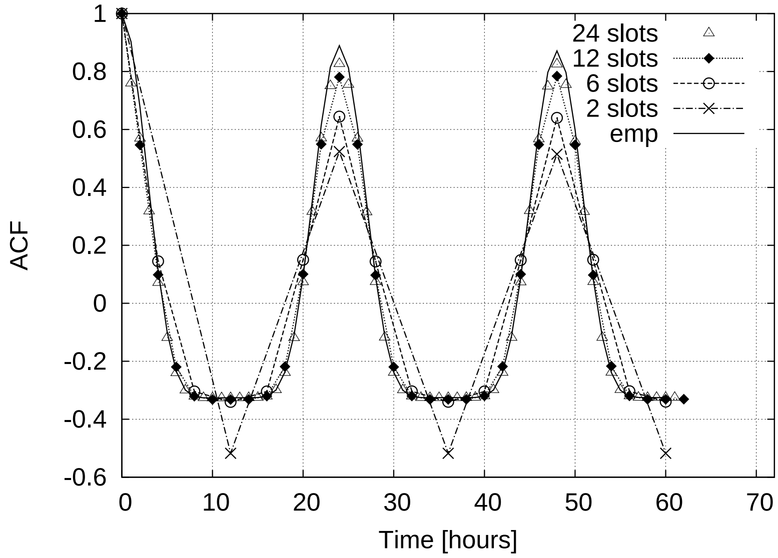


Figure 3.11. *Autocorrelation function for empirical data (solid curve) and for a synthetic Markov process generated through the night-day clustering (*2 *slots) and the slot-based clustering (*6 *and* 12 *slots) approaches, obtained for the month of January.*

		Aug					Dec				
$n_\mathrm{p} \times n_\mathrm{s}$	Size	$\bar{i}$	max(i)	$\bar{\tau}$	min(τ)	max(τ)	$\bar{i}$	max(i)	$\bar{\tau}$	min(τ)	max(τ)
	[cm^2]	[mA]	[mA]	[h]	[h]	[h]	[mA]	[mA]	[h]	[h]	[h]
2 x 2	2.99	2.163	4.524	9.73	8.17	10.17	1.110	2.484	7.74	5.00	8.33
4 x 4	11.98	9.254	19.766	10.18	9.00	10.67	4.847	11.029	8.27	6.50	8.67
6 x 6	26.96	21.292	45.561	10.26	9.17	10.67	11.189	25.666	8.38	6.67	8.83
8 x 8	47.92	38.149	82.101	10.32	9.17	10.83	20.115	46.006	8.42	6.83	8.83
10 x 10	74.88	59.967	129.194	10.34	9.17	10.83	31.650	72.437	8.44	6.83	8.83
12 x 12	107.83	86.729	186.905	10.35	9.17	10.83	45.795	104.829	8.45	6.83	9.00

Table 3.1. *Results for different solar panel configurations with night-day clustering in Los Angeles*

3.4 Concluding Remarks

In this chapter we have considered micro-solar power sources, providing a methodology to model the energy inflow as a function of time through stochastic Markov processes. The latter, find application in energy self-sustainable systems, such as, for instance, in the simulation of energy harvesting communication networks and are as well useful to extend current theoretical work through more realistic energy models. Our approach has been validated against real

Location	Aug					Dec				
	$\bar{i}$	max(i)	$\bar{\tau}$	min(τ)	max(τ)	$\bar{i}$	max(i)	$\bar{\tau}$	min(τ)	max(τ)
	[mA]	[mA]	[h]	[h]	[h]	[mA]	[mA]	[h]	[h]	[h]
Chicago, IL	17.029	46.742	10.57	8.50	11.33	5.241	16.084	6.95	4.83	8.00
Los Angeles, CA	21.292	45.561	10.26	9.17	10.67	11.189	25.666	8.38	6.67	8.83
New York, NY	17.174	44.617	10.42	8.83	11.00	6.813	18.945	7.57	5.67	8.33
Reno, NV	22.912	48.525	10.72	9.16	11.00	8.247	21.128	7.85	6.00	8.50

Table 3.2. *Results for different solar panel locations for* $n_{\mathrm{p}} = n_{\mathrm{s}} = 6$

energy traces, showing good accuracy in their statistical description in terms of first and second order statistics.

Our tool has been developed using Matlab$^{\mathrm{TM}}$and is available under the GPL license at [47].

4

On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking

In the past few years, temporal lossy compression has been widely applied in the field of wireless sensor networks (WSN), where energy efficiency is a crucial consideration due to the constrained nature of the sensor devices. Often, the common thinking among researchers and implementers is that compression is always a good choice, because the major source of energy consumption in a sensor node comes from the transmission of the data.

Lossy compression is deemed a viable solution as the imperfect reconstruction of the signal is often acceptable in WSNs, subject to some (application dependent) maximum error tolerance. In this chapter, we evaluate a number of selected lossy compression methods from the literature, and extensively analyze their performance in terms of compression efficiency, computational complexity and energy consumption. Specifically, we first carry out a performance evaluation of existing and new compression schemes, considering linear, autoregressive, FFT-/DCT- and Wavelet-based models, by looking at their performance as a function of relevant signal statistics. Second, we obtain formulas through numerical fittings, to gauge their overall energy consumption and signal representation accuracy. Third, we evaluate the benefits that lossy compression methods bring about in interference-limited multi-hop networks, where the channel access is a source of inefficiency due to collisions and transmission scheduling. Our results reveal that the DCT-based schemes are the best option in terms of compression efficiency but are inefficient in terms of energy consumption. Instead, linear methods lead to substantial savings in terms of energy expenditure by, at the same time, leading to satisfactory compression ratios, reduced network delay and increased reliability performance.

4.1 Introduction

In recent years, wireless sensors and mobile technologies have experienced a tremendous upsurge. Advances in hardware design and micro-fabrication have made it possible to potentially embed sensing and communication devices in every object, from banknotes to bicycles [48].

Wireless Sensor Network (WSN) technology has now reached a good level of maturity, as testified by the many emerging industrial standardization efforts [49]. Notable WSN application examples include environmental monitoring [50], geology [51] structural monitoring [52], smart grid and household energy metering [53, 54]. These applications often require the collection and the subsequent analysis of large amounts of data, which are to be sent through suitable routing protocols to some data collection point(s). One of the main problems of this is related to the large number of devices: if this number will keep increasing as predicted in [55], and all signs point toward this direction, the amount of data to be managed by the network will become prohibitive. Further issues are due to the constrained nature of sensor nodes in terms of limited energy resources (devices are often battery operated) and to the fact that radio activities are their main source of energy consumption. This, together with the fact that sensor nodes are required to remain unattended and operational for long periods of time, poses severe constrains.

Several strategies have been developed to prolong the lifetime of sensor nodes. These comprise processing techniques such as data aggregation [56], distributed [57] or temporal [58] compression as well as battery replenishment through energy harvesting [59]. The rationale behind data compression is that we can trade some additional energy for compression for some reduction in the energy spent for transmission. As we shall see in the remainder of this chapter, this allows some important savings.

We focus on the energy saving opportunities offered by data processing and, in particular, on the *lossy temporal compression* of data. With lossy techniques, the original data is compressed by however discarding some of the original information in it so that, at the receiver side, the decompressor can reconstruct the original data up to a certain accuracy. Lossy compression makes it possible to trade some reconstruction accuracy for some additional gains in terms of compression ratio with respect to lossless schemes. Note that these gains correspond to further savings in terms of transmission needs and that, depending on the application, some small inaccuracy in the reconstructed signal may be acceptable. Thus, lossy compression introduces some additional flexibility as one can tune the compression ratio as a function of energy consumption criteria.

We note that much of the existing literature has been devoted to the systematic study of lossless compression. [60] proposes a simple Lossless Entropy Compression (LEC) algorithm, comparing LEC with standard techniques such as gzip, bzip2, rar and classical Huffman and arithmetic encoders. A simple lossy compression scheme, called Lightweight Temporal Compression (LTC) [61], was also considered. However, the main focus of this comparison has been on the achievable compression ratio, whereas considerations on energy savings are only given for LEC. [62] examines Huffman, Run Length Encoding (RLE) and Delta Encoding (DE), comparing the energy spent for compression for these schemes. [63] treats lossy (LTC) as well as lossless (LEC and Lempel-Ziv-Welch) compression methods, but only focusing on their compression performance. Further work is carried out in [64], where the energy savings from lossless compression algorithms are evaluated for different radio setups, in single- as well as multi-hop networks. Along the same lines, [65] compares several lossless compression schemes for a StrongArm CPU architecture, showing that data compression in some cases may cause an increase in the overall energy expenditure. A comprehensive survey of practical lossless compression schemes for WSN can be found in [66]. The lesson that we learn from these papers is that lossless compression can provide some energy savings. These are however smaller than one might expect because, for the sensor hardware in use nowadays, the energy spent for the execution of the compression algorithms (CPU) may be of the same order of magnitude of that spent for transmission (radio).

Further work has been carried out for what concerns lossy compression schemes. LTC [67], PLAMLiS [68] and the algorithm of [69] are all based on Piecewise Linear Approximation (PLA). Adaptive Auto-Regressive Moving Average (A-ARMA) [67] is based on ARMA models and RACE [70] exploits Wavelet-based compression. Also, [71] presents a lightweight compression framework based on Differential Pulse Coding Modulation (DPCM) where dictionaries are selected offline through multi-objective evolutionary optimization. Nevertheless, we remark that no systematic energy comparison has been carried out so far for lossy schemes. In this case, it is not clear whether lossy compression can be advantageous in terms of energy savings and what the involved tradeoffs are in terms of compression ratio *vs* representation accuracy and yet how these affect the overall energy expenditure. In addition, it is unclear whether linear and autoregressive schemes can provide any advantages at all compared to more sophisticated techniques such as Fourier- or Wavelet-based transforms, which have been effectively used to compress audio and video signals and for which fast and computationally efficient algorithms exist. In this chapter, we fill these gaps by systematically comparing selected lossy temporal compression methods from the literature including polynomial, Fourier (FFT and DCT) and Wavelet compression schemes. We remark that alternative approaches, such as data aggregation [56] are also possible. However, these are out of the scope of our investigation, which focuses on the temporal and lossy compression of time series.

The main contributions of this chapter are:

- We consider selected lossy compression algorithms for time series, accounting for linear (e.g., LTC [67]), autoregressive (e.g., A-ARMA [67]) models, Fourier and Wavelet transforms. At first, we focus on interference-free single- and multi-hop networks, where the Medium Access Control (MAC) layer is idealized, i.e., besides transmission and reception, it does not introduce further energetic inefficiencies due to collisions and idle times for floor acquisition. For this scenario, we assess whether signal compression actually helps in the reduction of the overall energy consumption, depending on the compression algorithm, the chosen reconstruction fidelity, the signal statistics and the hardware characteristics.

- We provide formulas, obtained through numerical fittings and validated against real datasets, to gauge the computational complexity, the overall energy consumption and the signal representation accuracy of the best performing compression algorithms as a function of the most relevant system parameters. These formulas can be used to generalize the results obtained here to other WSN architectures.

- We consider interference-limited multi-hop networks where multiple nodes contend for the channel and data traverses a data collection tree until it reaches a data collection point located at its root (the WSN “sink”). Thus, we analytically characterize this second scenario by evaluating the performance improvements that are brought about by different lossy compression schemes in the presence of collisions and idle times for floor acquisition at the MAC.

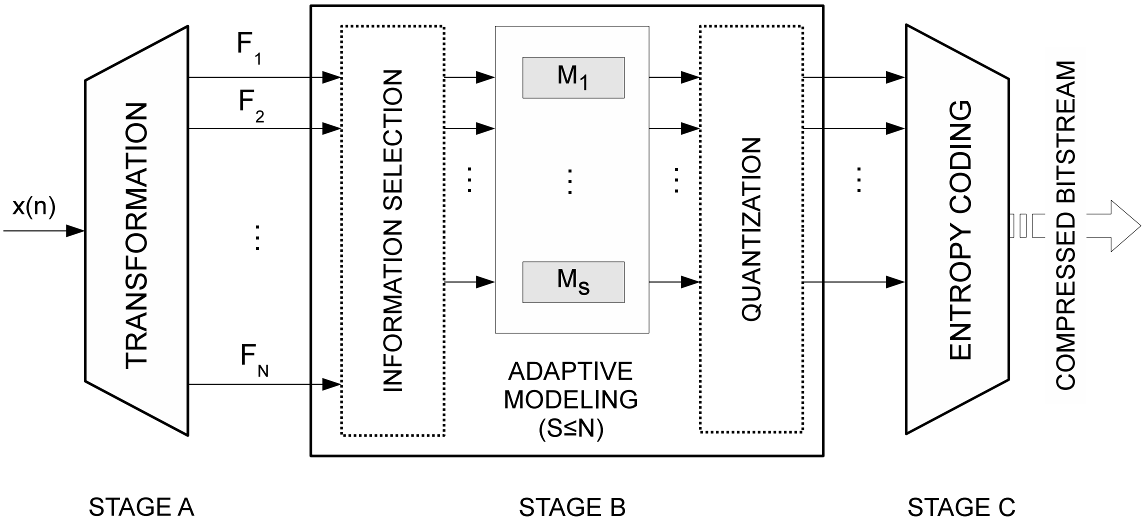


Figure 4.1. *General lossy compression diagram.*

The rest of the chapter is organized as follows. In Section 4.2 we discuss selected lossy compression algorithms from the literature, along with some other lossy compression schemes that we introduce there. In Sections 4.3 and 4.4 we carry out our performance evaluation of lossy compression for the interference-free and the interference-limited WSN scenarios, respectively. We finally draw our conclusions in Section 4.5.

4.2 Lossy Compression for Constrained Sensor Nodes

To facilitate the description of the compression schemes we considered and to identify their essential features, in Fig. 4.1 we show the diagram of a generic lossy compression algorithm, see, e.g., [72]. The following three fundamental stages are identified:

A **Transformation:** this stage entails the representation of the input signal (time series $x(n)$) into a convenient transformation domain. That is, the signal is decomposed into a number N of coefficients $\{F_1, \dots, F_N\}$ in the new domain. As an example, FFT, DCT and Wavelet transforms represent time series into the frequency domain.

B **Adaptive modeling:** a number of coefficients $S \leqslant N$ is selected so that these will be sufficient to represent the signal within a certain target accuracy. Moreover, a further adaptive modeling phase (models $\{M_1, \dots, M_S\}$) can be applied on the time series corresponding to each of the selected coefficients and, finally, a quantizer can be employed to represent the data through a finite number of levels.

C **Entropy coding:** the quantized data can be encoded using an *entropy coder* (EC) to obtain additional compression. Entropy represents the amount of information present in the data, and an EC encodes the given set of symbols with the minimum number of bits required to represent them.

As a popular example, JPG image compression [72] matches this model as follows: Stage-A: DCT, Stage-B: DPCM modeling for the DC coefficients, quantization for all coefficients, with run length encoding for null coefficients after quantization, Stage-C: huffman coding (arithmetic coding is also supported).

We remark that a specific compression algorithm does not necessary have to implement all the three stages above, but some of them can be omitted or only partially taken into account. For example, for Stage-B we could use the selection and quantization blocks, without any adaptive modeling. In WSNs, the exact combination of algorithms to use depends on the reconstruction accuracy goal as well as on the affordable computational complexity.

In the following, we briefly review the lossy signal compression methods that will be characterized later on in this chapter. Due to the contained nature of the sensor devices, these schemes only use some of the above stages. In Section 4.2.1, we discuss techniques based on Fourier and Wavelet transforms (Stage-A). In Section 4.2.2, we describe adaptive modeling techniques (Stage-B). Finally, in Section 4.2.3 we discuss a lightweight scheme based on quantization and entropy coding (Stage-C).

4.2.1 Compression Methods Based on Fourier and Wavelet Transforms (Stage-A)

For these techniques, compression is achieved through sending subsets of the FFT, DCT or Wavelet transformation coefficients. We came up with some possible methods, presented below, that differ in how the transformation coefficients are picked. These algorithms first transform the signal into a suitable domain (Stage-A) and subsequently use the *information selection* block of Stage-B.

4.2.1.1 Fast Fourier Transform (FFT)

The first method that we consider relies on the simplest way to use the Fourier transform for compression. Specifically, the input time series $x(n)$ is mapped to its frequency representation $X(f) \in \mathbb{C}$ through a Fast Fourier Transform (FFT). We define $X_{\mathcal{R}}(f) \triangleq \mathfrak{Re}\{X(f)\}$, and $X_{\mathcal{I}}(f) \triangleq \mathfrak{Im}\{X(f)\}$ as the real and the imaginary part of $X(f)$, respectively. Since $x(n)$ is a real-valued time series, $X(f)$ is Hermitian, i.e., $X(-f) = \overline{X(f)}$. This symmetry allows the FFT to be stored using the same number of samples N of the original signal. For N even we take $f \in \{f_1, \ldots, f_{N/2}\}$ for both $X_{\mathcal{R}}(\cdot)$ and $X_{\mathcal{I}}(\cdot)$, while if N is odd we take $f \in \{f_1, \ldots, f_{\lfloor N/2 \rfloor + 1}\}$ for the real part and $f \in \{f_1, \ldots, f_{\lfloor N/2 \rfloor}\}$ for the imaginary part.

The compressed representation $\hat{X}(f) \triangleq \hat{X}_{\mathcal{R}}(f) + j\hat{X}_{\mathcal{I}}(f)$ will also be in the frequency domain and it is built (for the case of N even) as follows:

1. initialize $\hat{X}_{\mathcal{R}}(f) = 0$ and $\hat{X}_{\mathcal{I}}(f) = 0$, $\forall\, f \in \{f_1, \ldots, f_{N/2}\}$;
2. select the coefficient with maximum absolute value from $X_{\mathcal{R}}$ and $X_{\mathcal{I}}$, i.e., $f_{\max} \triangleq \operatorname{argmax}_f \max\{|X_{\mathcal{R}}(f)|, |X_{\mathcal{I}}(f)|\}$ and $M \triangleq \operatorname{argmax}_{i \in \{\mathcal{R},\mathcal{I}\}}\{|X_i(f_{\max})|\}$;
3. set $\hat{X}_M(f_{\max}) = X_M(f_{\max})$ and then set $X_M(f_{\max}) = 0$;
4. if $\hat{x}(n)$, the inverse FFT of $\hat{X}(f)$, meets the error tolerance constraint continue, otherwise repeat from step (2);
5. encode the values and the positions of the harmonics stored in $\hat{X}_{\mathcal{R}}$ and $\hat{X}_{\mathcal{I}}$.

Hence, the decompressor at the receiver obtains $\hat{X}_{\mathcal{R}}(f)$ and $\hat{X}_{\mathcal{I}}(f)$ and exploits the Hermitian symmetry to reconstruct $\hat{X}(f)$.

Note that the above coefficient selection method resembles a K non-linear approximation, as usually implemented by image processing techniques see, e.g., [73]. In our case, K (the number of coefficients to be retained) is dynamically selected depending on the input signal characteristics. We emphasize that alternative selection schemes are also possible. For instance, one may select the FFT coefficients based on the maximum absolute magnitude of their complex values and then retain both the real and imaginary part of the selected coefficients. In our tests, the algorithm that we have described above has shown better performance in terms of compression capabilities and computational complexity. For this reason, further approaches are not evaluated in what follows.

4.2.1.2 Low Pass Filter (FFT-LPF)

We implemented a second FFT-based lossy algorithm, which we have termed FFT-LPF. Since the input time series $x(n)$ is a slowly varying signal in many common cases (i.e., having strong temporal correlation) with some high frequency noise superimposed, most of the significant coefficients of $X(f)$ reside in the low frequencies. For FFT-PLF, we start setting $\hat{X}(f) = 0$ for all frequencies. Thus, $X(f)$ is evaluated from f_1, incrementally moving toward higher frequencies, $f_2, f_3, \ldots$. At each iteration i, $X(f_i)$ is copied onto $\hat{X}(f_i)$ (both real and imaginary part), the inverse FFT is computed taking $\hat{X}(f)$ as input and the error tolerance constraint is checked on the so obtained $\hat{x}(n)$. If the given tolerance is met the algorithm stops, otherwise it is reiterated for the next frequency f_{i+1}.

Note that this method resembles a K linear approximation scheme, where the selection order is fixed (LPF), but the number of coefficients to be retained, K, is dynamically adjusted in order to meet a given error tolerance.

4.2.1.3 Windowing

The two algorithms discussed above suffer from an edge discontinuity problem. In particular, when we take the FFT over a window of N samples, if $x(1)$ and $x(N)$ differ substantially the information about this discontinuity is spread across the whole spectrum in the frequency domain. Hence, in order to meet the tolerance constraint for all the samples in the window, a high number of harmonics is selected by the previous algorithms, resulting in a poor compression and in a high number of operations.

To solve this issue, we implemented a version of the FFT algorithm that considers overlapping windows of $N + 2W$ samples instead of disjoint windows of length N, where W is the number of samples that overlap between subsequent windows. The first FFT is taken over the entire window and the selection of the coefficients goes on depending on the selected algorithm (either FFT or FFT-LPF), but the tolerance constraint is only checked on the N samples in the central part of the window. With this workaround we can get rid of the edge discontinuity problem and encode the information about the N samples of interest with very few coefficients as it will be seen shortly in Section 4.3. As a drawback, the direct and inverse transforms have to be taken on longer windows, which results in a higher number of operations.

4.2.1.4 Discrete Cosine Transform (DCT)

We also considered the Discrete Cosine Transform (type II), mainly for three reasons: 1) its coefficients are real, so we did not have to cope with real and imaginary parts, thus saving memory and number of operations; 2) it has a strong "energy compaction" property [74], i.e., most of the signal information tends to be concentrated in a few low-frequency components; 3) the DCT of a signal with N samples is equivalent to a DFT on a real signal of even symmetry with double length, so the DCT does not suffer from the edge discontinuity problem.

4.2.1.5 Wavelet Transform (WT)

As an alternative to Fourier schemes, several methods based upon multi-resolution analysis have been proposed in the literature. RACE [70] is a notable example: it features a compression algorithm based on the Fast Wavelet Transform (FWT) of the signal (Stage-A) followed by the selection of a number of coefficients (Stage-B) that are used to represent the input signal within given error bounds. As for DCT schemes, the compression mainly takes place in the selection step.

In [70], a Haar basis function is used for the wavelet decomposition step. The most remarkable contribution of RACE is the way in which the wavelet coefficients are selected. Most traditional compression algorithms, after the FWT, just pick the largest coefficients, i.e., the selection step is based on a threshold value, whereby all the coefficients below this threshold are discarded, whereas those above it are retained. Differently, in RACE, the Haar wavelet coefficients are arranged into a tree structure. Then, thanks to some special properties of the Haar functions, at each node of the tree, the error in the reconstruction of the signal is estimated assuming that this node (i.e., the corresponding coefficient) and all its children in the tree are omitted.

This selection method has two important properties. First, the signal representation error can be evaluated on-the-fly during the decomposition and the maximum error tolerance can be maintained under control, without having to compute any inverse wavelet transform.[1] Second, compression can be achieved in an incremental way, by descending the tree and adding nodes until the desired precision is reached (of course, the higher the number of

[1] Note that in the FFT and DCT methods of above, the error tolerance check always entails the computation of an inverse transformation at the source.

coefficients, the lower the compression performance). These facts are very important for energy constrained WSNs and, as we will see in Section 4.3, lead to a smaller energy for compression with respect to DCT and FFT schemes.

4.2.2 Compression Methods Based on Adaptive Modeling (Stage-B)

In Adaptive Modeling schemes, some signal model is iteratively updated over time, exploiting the correlation structure of the signal through linear, polynomial or autoregressive methods. Specifically, the input time series is collected and processed according to transmission windows of N samples each. At the end of each time window the selected compression method is applied, obtaining a set of model parameters that are transmitted in place of the original data. In the adaptive modeling schemes described below, information selection is not used, as they do not employ any transformation stage.

4.2.2.1 Piecewise Linear Approximations (PLA)

The idea of PLA is to use a sequence of line segments to represent an input time series $x(n)$ over pre-determined time windows (of N samples) with a bounded approximation error. For most time series consisting of environmental measures, linear approximations work well enough over short time frames. Further, since a line segment can be determined by only two end points, PLA leads to quite efficient implementations in terms of memory and transmission requirements.

The approximated signal is hereafter referred to as $\hat{x}(n)$, the error with respect to the actual value is given by the Euclidean distance $|\hat{x}(n) - x(n)|$. Most PLA algorithms use standard least squares fitting to calculate the approximating line segments. Often, a further simplification is introduced to reduce the computational complexity, which consists of forcing the end points of each line segment to be points of the original time series $x(n)$. This makes least squares fitting unnecessary as the line segments are fully identified by the extreme points of $x(n)$ in the considered time window. The following schemes exploit this approach.

Lightweight Temporal Compression (LTC) [61]: the LTC algorithm is a low complexity PLA technique. Specifically, let $x(n)$ be the points of a time series with $n = 1, 2, \ldots, N$. The LTC algorithm starts with $n = 1$ and fixes the first point of the approximating line segment to $x(1)$. The second point $x(2)$ is transformed into a vertical line segment that determines the set of all "acceptable" lines $\Omega_{1,2}$ with starting point $x(1)$. This vertical segment is centered at $x(2)$ and covers all values meeting a maximum tolerance $\varepsilon \geqslant 0$, i.e., lying within the interval $[x(2) - \varepsilon, x(2) + \varepsilon]$, see Fig. 4.2a. The set of acceptable lines for $n = 3$, $\Omega_{1,2,3}$, is obtained by the intersection of $\Omega_{1,2}$ and the set of lines with starting point $x(1)$ that are acceptable for $x(3)$, see Fig. 4.2b. If $x(3)$ falls within $\Omega_{1,2,3}$ the algorithm continues with the next point $x(4)$ and the new set of acceptable lines $\Omega_{1,2,3,4}$ is obtained as the intersection of $\Omega_{1,2,3}$ and the set of lines with starting point $x(1)$ that are acceptable for $x(4)$. The procedure is iterated adding one point at a time until, at a given step s, $x(s)$ is not contained in $\Omega_{1,2,\dots,s}$. Thus, the algorithm sets $x(1)$ and $x(s-1)$ as the starting and ending points of the approximating line segment for $n = 1, 2, \dots, s-1$ and starts over with $x(s-1)$ considering it as the first point of the next approximating line segment. In our example, $s = 4$, see Fig. 4.2b.

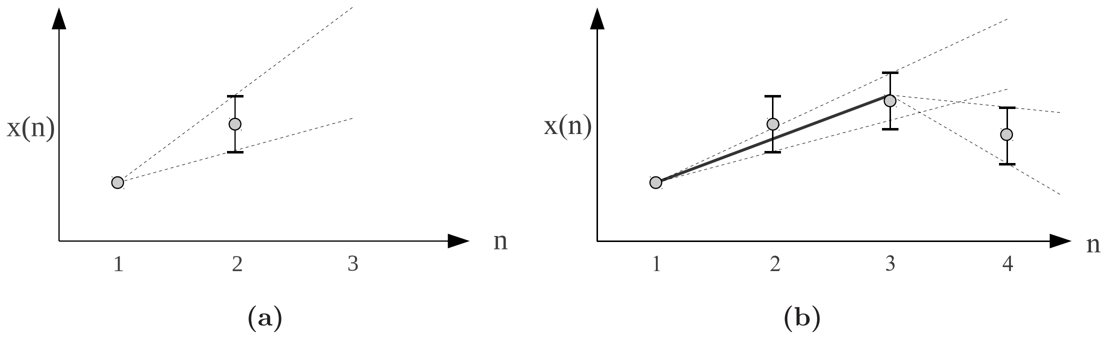


Figure 4.2. *Lightweight Temporal Compression example.*

When the inclusion of a new sample does not comply with the allowed maximum tolerance, the algorithm starts over looking for a new line segment. Thus, it self-adapts to the characteristics of $x(n)$ without having to fix beforehand the lapse of time between subsequent updates.

PLAMLiS [68]: as LTC, PLAMLiS represents the input data series $x(n)$ through a sequence of line segments. Here, the linear fitting problem is converted into a set-covering problem, trying to find the minimum number of segments that cover the entire set of values over a given time window. This problem is then solved through a greedy algorithm as explained in [68]. This algorithm is outperformed in terms of complexity by its enhanced version, that we discuss next.

Enhanced PLAMLiS [69]: is a top-down recursive segmentation algorithm with smaller computational cost with respect to PLAMLiS. Consider the input time series $x(n)$ and a time window $n = 1, 2, \ldots, N$. The algorithm starts by taking a first segment $(x(1), x(N))$, if the maximum allowed tolerance ε is met for all points along this segment the algorithm ends. Otherwise, the segment is split in two segments at the point $x(i)$, $1 < i < N$, where the error is maximum, obtaining the two segments $(x(1), x(i))$ and $(x(i), x(N))$. The same procedure is recursively applied on the resulting segments until the maximum error tolerance is met for all points.

4.2.2.2 Polynomial Regression (PR)

The above methods can be modified by relaxing the constraint that the endpoints of the segments $x(i)$ and $x(j)$ $(j > i)$ must be actual points of $x(n)$. In this case, polynomials of given order $p \geqslant 1$ are used as the approximating functions, whose coefficients are found through standard regression methods based on least squares fitting [75]. Specifically, we start with a window of p samples, since a p-order polynomial exactly interpolates p points, for which we obtain the best fitting polynomial coefficients. Thus, we keep increasing the window length of one sample at a time, computing the new coefficients, and stop when the target error tolerance is no longer met.

We remark that, tracing a line between two fixed points as done by LTC and PLAMLiS has a very low computational complexity, while least squares fitting can have a significant cost. Polynomial regression obtains better results in terms of approximation at the cost of higher computational complexity (which increases with the polynomial order).

4.2.2.3 Auto-Regressive (AR) Methods

Auto Regressive (AR) models in their multiple flavors (AR, ARMA, ARIMA, etc.) have been widely used for time series modeling and forecasting in fields like macro-economics or market analysis. The basic idea is to obtain a model based on the history of the sampled data, i.e., on its correlation structure. When used for signal compression, AR obtains a model from the input data and sends this model to the receiver in place of the actual time series. The reconstructed model is thus used at the data collection point (the sink) for data prediction until it is updated by the encoder device. Specifically, each node locally verifies the accuracy of the predicted data values with respect to the collected samples. If the accuracy is within a prescribed error tolerance, the node assumes that the current model will be sufficient for the sink to rebuild the data within the given error tolerance. Otherwise, the parameters from the current model are encoded and a new model is built as a replacement for the old one. As said above, the model parameters are sent to the sink at the end of each transmission window in place of the original data.

Adaptive Auto-Regressive Moving Average (A-ARMA) [67]: the basic idea of A-ARMA [67] is that of having each sensor node compute an ARMA model based on $N' < N$ consecutive samples. In order to reduce the complexity in the model estimation process, adaptive ARMA employs low-order models, whereby the validity of the model being used is checked through a moving window technique. Specifically, a sensor node builds an ARMA model $M^{(0)} = ARMA(p, q, N', 0)$ considering N' samples starting from the first sample (sample 0) of the current transmission window (p and q are the orders related to the auto-regressive and moving average components of the ARMA filter). Hence, this model is updated considering N' subsequent samples at a time until the prescribed error tolerance is met, at which point a new ARMA model is built and the update/check procedure is iterated for this one. At the end of the transmission window of N samples, the parameters of all the ARMA models that have been obtained to describe the input time series (within the prescribed error tolerance) are sent to the sink in place of the original data, as discussed above.

Modified Adaptive Auto-Regressive (MA-AR): according to A-ARMA the model is updated over fixed-size windows of N' samples. A drawback of this is that, especially for highly noisy environments, the estimation over fixed-size windows can lead to poor results when used for forecasting. MA-AR allows the estimation to be performed on time windows whose size is adapted according to the signal statistics. A more detailed discussion of ARMA methods can be found in [76].

4.2.3 Compression Methods Based on Entropy Coding (Stage-C)

As a representative technique for Stage-C we consider the algorithm in [71], proposed by

Marcelloni and Vecchio (MV). This algorithm works in three steps: (a) Differential Pulse-Modulation Coding (DPCM), (b) quantization and (c) huffman entropy encoding. After de-noising, step (a) employs a simple differential encoding model (DPCM), which operates on the differences between consecutive input samples. The rationale behind this differential scheme is that WSN signals are usually smooth and slow time-varying. Hence, the difference between samples is expected to be small, leading to a small amount of information to be encoded.

In the quantization block (b), the difference between subsequent samples is quantized. This is the most important step of the algorithm and probably where most of the compression performance is achieved. In fact, given the small expected value of the DPCM differences, a quantizer with only a small number of levels can be used without impacting too much on the signal representation accuracy.

In our performance evaluation, in order to carry out a fair comparison among the considered compression schemes, we bound the maximum error tolerance for each sample, setting it as a constant input parameter equal for all the algorithms. As we did for the other compression schemes, the MV algorithm has been as well adapted to consider this. Specifically, a first pass is performed to find the maximum difference at the output of the DPCM. Based on this, the number of levels of the quantizer is selected so that the quantization error remains smaller than a target error tolerance; this returns the quantizer for the given input signal. After this, a second pass is executed, using the selected quantizer, to obtain the final encoded symbols. Note that this is slightly different from [71], where optimal quantizers are calculated offline through a dedicated optimization stage following different optimization criteria. While the latter approach is also valuable, it does not allow for a precise control of the maximum error tolerance and a fair comparison with the other compression schemes that we consider in this chapter.

Finally, the entropy encoding step (c) exploits the fact that the quantization levels have different probabilities. Once again, environmental signals are quite smooth and therefore small differences are more likely. Hence, a Huffman encoder is designed to assign the shorter binary codewords to the most probable levels. The set of binary codewords is selected so that no member is a prefix of another member and, in turn, the corresponding code is uniquely decodable. This dictionary can be sent together with the compressed data frame, or can be statistically precomputed and shared between the communicating entities.

4.3 Performance Comparison for Interference-free Networks

This section focuses on single- and multi-hop WSNs where the interference due to channel access is negligible or absent. In this case, the energy expenditure at the MAC is only confined to transmission and reception energy, by also keeping into account the protocol overhead at the MAC in terms of packet headers. However, further energetic inefficiencies due to channel contentions and waiting times due to floor acquisition are neglected (their impact will be considered later on in Section 4.4). The objectives of this section are:

- to provide a thorough performance comparison of the compression methods of Section 4.2. The selected performance metrics are: 1) compression ratio, 2) computational and transmission energy and 3) reconstruction error at the receiver, which will be defined below.
- To quantify the impact on the compression performance of the statistical properties of the input signals.
- To investigate whether or not data compression leads to energy savings in single- and multi-hop interference-free WSN scenarios, and obtain quantitative measurements of possible benefits as a function of compression ratio and energy consumption of the wireless end-nodes hardware (micro-controller and radio).
- To obtain, through numerical fitting, close-form equations which model the considered performance metrics as a function of key parameters.

Toward the above objectives, we present simulation results obtained using synthetic signals with varying correlation length. These signals make it possible to give a fine grained description of the performance of the selected techniques, so as to look comprehensively at the entire range of variation of the temporal correlation statistics. Real datasets are then used to validate the proposed empirical fitting formulas.

4.3.1 Preliminary Definitions

Before delving into the description of the results, in the following we give some definitions.

Definition 4.3.1. Correlation length

Given a stationary discrete time series $x(n)$ with $n = 1, 2, \ldots, N$, we define **correlation length** of $x(n)$ as the smallest value $n^\star$ such that the autocorrelation function of $x(n)$ is smaller than a predetermined threshold ρ_{th}. The autocorrelation is:

$$\rho_x(n) = \frac{\mathrm{E}\left[(x(m) - \mu_x)(x(m+n) - \mu_x)\right]}{\sigma_x^2} \ ,$$

where μ_x and σ_x^2 are the mean and the variance of $x(n)$, respectively. Formally, $n^\star$ is defined as:

$$n^\star = \operatorname*{argmin}_{n>0} \{\rho_x(n) < \rho_{\text{th}}\} \ .$$

Below, we define the performance metrics that will be considered in the remainder of the chapter.

Definition 4.3.2. Compression ratio

Given a finite time series $x(n)$ and its compressed version $\hat{x}(n)$, we define **compression ratio** η the quantity:

$$\eta = \frac{N_b(\hat{x})}{N_b(x)} \ ,$$

where $N_b(\hat{x})$ and $N_b(x)$ are the number of bits used to represent the compressed time series $\hat{x}(n)$ and the original one $x(n)$, respectively.

Definition 4.3.3. Reconstruction error and error tolerance

Given a discrete time series $x(n)$ and its compressed version $\hat{x}(n)$, we define the reconstruction error at time $n \geqslant 1$ as $e(n) = |x(n) - \hat{x}(n)|$, where $| \cdot |$ is the Euclidean distance. The error tolerance ε is the maximum permitted error at the receiver, i.e., it must be $e(n) \leqslant \varepsilon$ for all n.

Definition 4.3.4. Energy consumption for compression

Is the energy drained from the battery to accomplish the compression task. For every compression method we have recorded the number of operations to process the original time series $x(n)$ accounting for the number of additions, multiplications, divisions and comparisons. Thus, depending on selected hardware architecture, we have mapped these figures into the corresponding number of clock cycles and we have subsequently mapped the latter into the corresponding energy expenditure.

Definition 4.3.5. Transmission Energy
Is the energy consumed for transmission, obtained accounting for the radio chip characteristics, channel attenuation effects and the protocol overhead due to physical (PHY) and medium access (MAC) layers.

Definition 4.3.6. Total Energy Consumption
Is the sum of the energy consumption for compression and transmission and is expressed in [Joule].

In the computation of the energy consumption for compression, we only accounted for the operations performed by the CPU, without considering the possible additional costs related to other peripherals of the micro-controller.

For the communication cost we have only taken into consideration the transmission energy, neglecting the cost of switching the radio transceiver on and off and the energy spent at the destination to receive the data. The former are fixed costs that would also be incurred without compression, while the latter can be ignored if the receiver is not a power constrained device. Moreover, we do not consider link-level retransmissions due to channel errors or multi-user interference.

4.3.2 Generation of Synthetic Stationary Signals

The synthetic stationary signals have been obtained through a known method to enforce the first and second moments to a white random process, see [77] [78]. Our objective is to obtain a random time series $x(n)$ with given mean μ_x, variance σ_x^2 and autocorrelation function $\rho_x(n)$. The procedure works as follow:

1. A random Gaussian series $G(k)$ with $k = 1, 2, \ldots, N$ is generated in the frequency domain, where N is the length of the time series $x(n)$ that we want to obtain. Every element of $G(k)$ is an independent Gaussian random variable with mean $\mu_G = 0$ and variance $\sigma_G^2 = 1$.

2. The Discrete Fourier Transform (DFT) of the autocorrelation function $\rho_x(n)$ is computed, $S_x(k) = \mathcal{F}[\rho_x(n)]$, where $\mathcal{F}[\cdot]$ is the DFT operator.

3. We compute the entry-wise product $X(k) = G(k) \circ S_x(k)^{\frac{1}{2}}$.

4. We finally obtain the correlated and Gaussian time series $x(n)$ as $\mathcal{F}^{-1}[X(k)]$.

This is equivalent to filter a white random process with a linear, time invariant filter, whose transfer function is $\mathcal{F}^{-1}[S_x(k)^{\frac{1}{2}}]$. The stability of this procedure is ensured by a suitable choice for the correlation function, which must be square integrable. For the simulations we have used a Gaussian correlation function [79], i.e., $\rho_x(n) = \exp\{-an^2\}$, where a is chosen in order to get the desired correlation length $n^\star$ as follows:

$$a = -\frac{\log(\rho_{\mathrm{th}})}{(n^\star)^2} \; .$$

Without loss of generality, we generate synthetic signals with $\mu_x = 0$ and $\sigma_x^2 = 1$. In fact, applying an offset to the generated signals and a scale factor does not change the resulting correlation. For an in depth characterization of the Gaussian correlation function see [79].

Also, to emulate the behavior of real WSN signals, we superimpose noise to the synthetic signals, so as to mimic random perturbations due to limited precision of the sensing hardware and random fluctuations of the observed physical phenomenon. The noise is modeled as a zero mean white Gaussian process with standard deviation σ_{noise}.

4.3.3 Hardware Architecture

We selected the TI MSP430 [80] micro-controller using the corresponding 16 bit floating point package for the calculations and for the data representation. In the active state, the MSP430 is powered by a current of 330 μA at 2.2 V and it has a clock rate of 1 MHz. The resulting energy consumption per CPU cycle is $E_0 = 0.726$ nJ. The number of clock cycles needed for the floating point operations are given in Table 5.8 of [80].

For radio, we selected the TI CC2420 RF transceiver [81], an IEEE 802.15.4 [82] compliant radio. For commercial radio transceivers, the current consumption associated with the transmission activity is typically selected from a finite set of values, that for the CC2420 are 8, varying from a minimum of 8.5 mA to a maximum of 17.4 mA, with a supply voltage of 3.3 V for an effective data rate of 250 kbps, see [81]. Thus, the energy cost associated with the transmission of a bit, $E'_{Tx}[\ell]$, given the current power level $\ell \in \{1, \ldots, 8\}$ ranges from 112 nJ to 230 nJ, which correspond to the energy spent by the micro-processor during 154 and 316 clock cycles, respectively. The current level, and consequently the output power of the radio transceiver, has to be chosen according to the considered scenario, which includes the transmission distance, the channel noise level, the type of environment (e.g., free space, indoor, presence of obstacles), etc.

We remark that the results that we obtain for this specific architecture can be promptly generalized to different CPUs and radios. As we show later in Section 4.3.8, this is possible by separating algorithm-dependent and hardware-dependent terms in the calculation of the overall energy consumption. In particular, the compression performance of all algorithm is evaluated using 16 bits arithmetics, the natural choice for the MSP430, a 16-bit word processor.

4.3.4 Theoretical Bound for Signal Compression

Given the discrete and Gaussian time series of Section 4.3.2, from the theory in [83] we can derive the theoretical lower bound on the transmission rate $R_{\min}$ (bits/sample):

$$R_{\min}(n^\star, N, \varepsilon) = \frac{1}{N}\sum_{i=1}^{N} \max\left\{0, \frac{1}{2}\log_2\left(\frac{\zeta_i^2}{\varepsilon}\right)\right\} ,$$

where ε is the maximum permitted distortion at the receiver, N is the number of input samples and ζ_i are the eigenvalues of the covariance matrix $\mathbf{\Sigma}(n^\star)$ of $x(n)$ for $n = 1, \ldots, N$. Hence, for given $(n^\star, N, \varepsilon)$ we can bound the compression ratio achievable by any practical scheme as:

$$\eta \geqslant \frac{R_{\min}(n^\star, N, \varepsilon)}{R_0} , \tag{4.1}$$

where $R_0 = 16$ is the rate expressed in bits/sample in the uncompressed case for our hardware.

4.3.5 Simulation Setup

For the results that we discuss in what follows, we used synthetic signals with correlation length $n^\star$ varying in $\{1, 10, 20, 50, \ldots, 500\}$ samples, where after 20, $n^\star$ varies in steps of 30 (we have picked $\rho_{\text{th}} = 0.05$ for all the results shown in this chapter). We consider time series of $N = 500$ samples (time slots) at a time, progressively taken from a longer realization of the signal, so as to avoid artifacts related to the generation technique. We recall that the signals are correlated Gaussian with zero-mean and unit variance. Moreover, a Gaussian noise with standard deviation $\sigma_{\text{noise}} = 0.04$ has been added to the signal, as per the signal generation method of Section 4.3.2. For the reconstruction accuracy, the absolute error tolerance has been set to $\varepsilon = \xi\sigma_{\text{noise}}$, with $\xi \geqslant 0$. In the following graphs, each point is obtained by averaging the outcomes of 10^4 simulation runs. For a fair comparison, the same realization of the input time series $x(n)$ has been used for all the compression methods considered, for each simulation run and value of $n^\star$. Moreover, all the compression algorithms have been configured with the same error tolerance, so that the energy compression and consumption figures that we obtain are for the same reconstruction fidelity at the receiver.

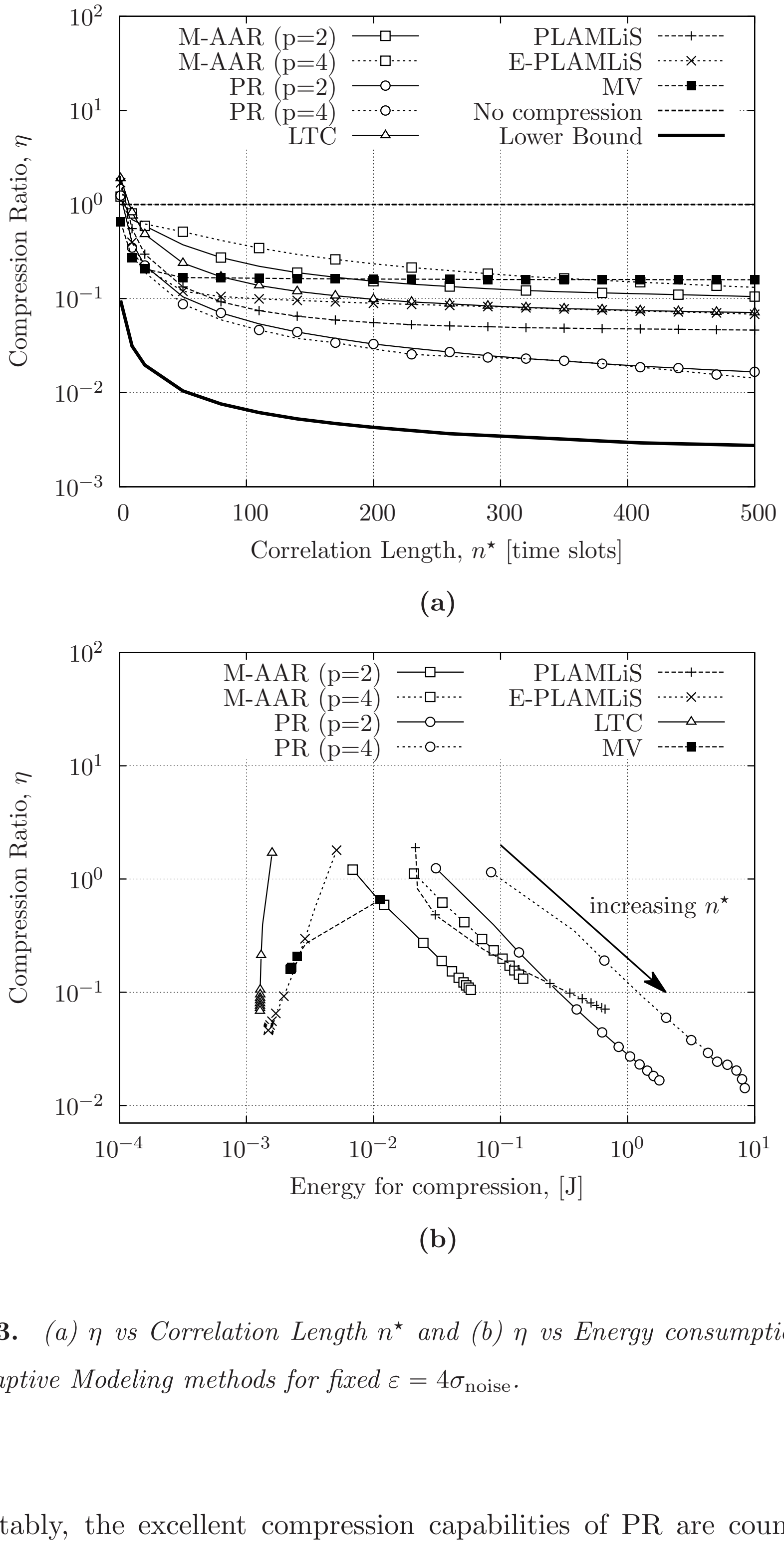
4.3.6 Compression Ratio vs Processing Energy

In the following, we analyze the performance in terms of compression effectiveness and computational complexity (energy) for the lossy compression methods of Section 4.2.

Adaptive Modeling Methods: in this first set of results we compare the performance of the following compression methods: 1) Modified Adaptive Autoregressive (M-AAR); 2) Polynomial Regression (PR); 3) Piecewise Linear Approximation (PLAMLiS); 4) Enhanced Piecewise Linear Approximation (E-PLAMLiS); 5) Lightweight Temporal Compression (LTC) and 6) Marcelloni and Vecchio's algorithm (MV) . For the M-AAR autoregressive filter and the polynomial regression (PR) we show results for the two orders, $p = \{2, 4\}$. The lower bound on the compression ratio η is also plotted for comparison, see (4.1).

Fig. 4.3a shows the Compression Ratio achieved by the six compression methods as a function of the correlation length $n^\star$. These results reveal that for small values of $n^\star$ the compression performance is poor for all compression schemes, whereas it improves for increasing correlation length, by reaching a floor value for sufficiently large $n^\star$. This confirms that $n^\star$ is a key parameter for the performance of all schemes. Also, the compression performance differs among the different methods, with PR giving the best results. This reflects the fact that, differently from all the other methods, PR approximates $x(n)$ without requiring its fitting curves to pass through the points of the given input signal. This entails some inherent filtering, that is embedded in this scheme and makes it more robust against small and random perturbations.

Fig. 4.3b shows the energy consumption for compression. For increasing values of $n^\star$ the compression ratio becomes smaller for all schemes, but their energy expenditure substantially



(a)

(b)

Figure 4.3. *(a) η vs Correlation Length $n^\star$ and (b) η vs Energy consumption for compression for the Adaptive Modeling methods for fixed $\varepsilon = 4\sigma_{\text{noise}}$.*

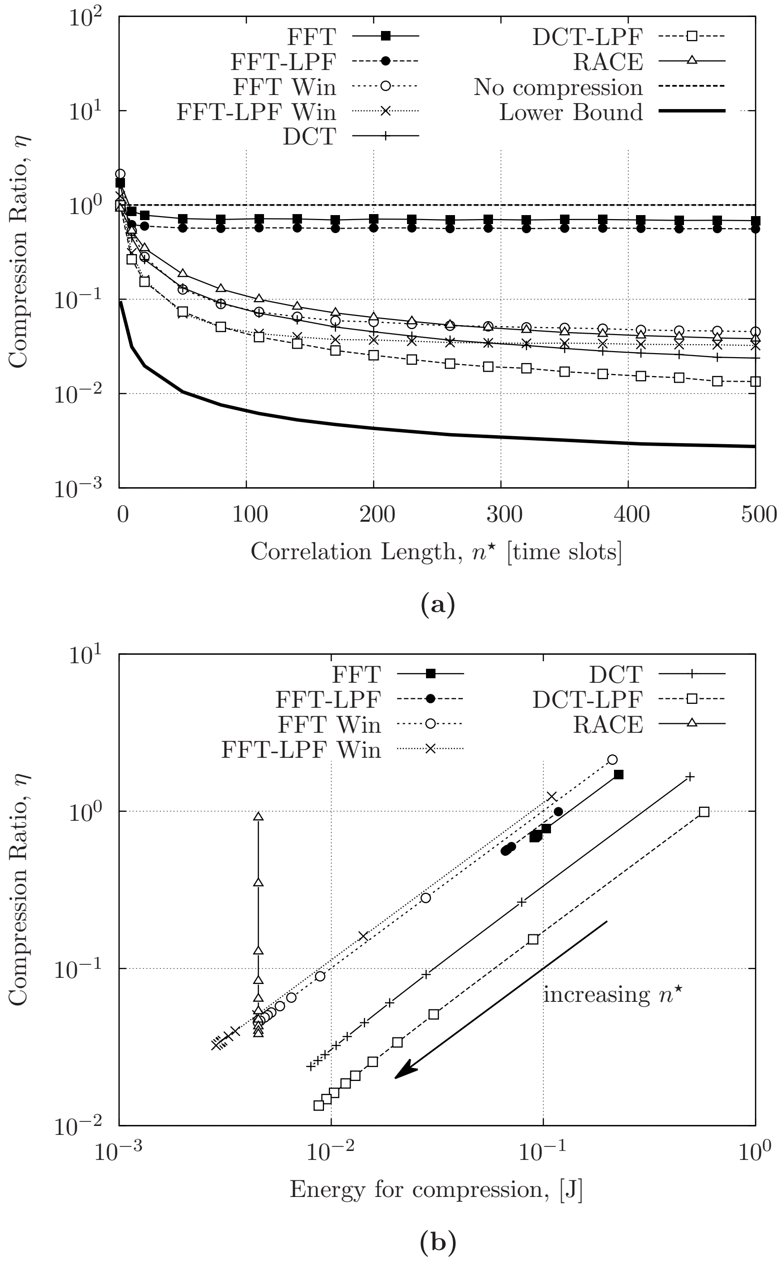
differs. Notably, the excellent compression capabilities of PR are counterbalanced by its demanding requirements in terms of energy. M-AAR and PLAMLiS also require a quite large amount of processing energy, although this is almost one order of magnitude smaller than that of PR. LTC, E-PLAMLiS and MV have the smallest energy consumption among all schemes.

We now discuss the dependence of the computational complexity (which is strictly related to the energy spent for compression) on $n^\star$. LTC encodes the input signal $x(n)$ incrementally, starting from the first sample and adding one sample at a time. Thus, the number of operations that it performs only weakly depends on the correlation length and, in turn, the energy that it spends for compression is almost constant with varying $n^\star$. E-PLAMLiS takes advantage of the increasing correlation length: as the temporal correlation increases, this method has to perform fewer "divide and reiterate" steps, so the number of operations required gets smaller and, consequently, also the energy spent for compression is reduced. MV performs almost the same number of operations for different correlation lengths, except for very small values of $n^\star$. This occurs because, in order to meet the error constraint for uncorrelated signals ($n^\star \approx 1$), the quantization step has to use a high number of levels (DPCM signals have wider ranges), and with an increasing number of levels the entropy encoder assigns an exponentially increasing number of bits to some symbols. As a consequence, also the number of operations related to the assignment of these codewords increases. We recall that, in order to fairly compare MV with the other methods that we analyze, we adapted it as explained in Section 4.2.3. Specifically, in the evaluation of the energy consumption associated with the compression operation at the transmitter, we also consider the operations performed for the online selection of the quantizer, so as to meet a target error tolerance. In our case, the results differ from those in [71] where optimal quantizers are computed offline, and only the final encoding stage is performed on the nodes, which entails a lower energy consumption. For the remaining methods the complexity grows with $n^\star$. For PLAMLiS, this is due to the first step of the algorithm, where for each point the longest segment that meets the given error tolerance has to be found, see Section 4.2. When $x(n)$ is highly correlated, these segments become longer and PLAMLiS has to check a large number of times the tolerance constraint for each of the N samples of $x(n)$. For M-AAR and PR every time a new sample is added to a model (autoregressive for the former and polynomial for the latter), this model must be updated and the error tolerance constraint has to be checked. These tasks have a complexity that grows with the square of the length of the current model. Increasing the correlation length of the input time series also increases the length of the models, leading to smaller compression ratios and, in turn, a higher energy consumption.

Fourier- and Wavelet-based Methods: we now analyze the performance of the Fourier- and Wavelet-based compression schemes of Section 4.2. We consider the same simulation setup as above. Fig. 4.4a shows that the compression performance of Fourier-based methods still improves with increasing $n^\star$. The methods that perform best are FFT Windowed, FFT-LPF Windowed and DCT-LPF, which achieve very small compression ratios, e.g., η is around 10^{-2} for $n^\star \geqslant 300$. Conversely, FFT and FFT-LPF, due to their edge discontinuity problem (see Section 4.2), need to encode more coefficients to meet the prescribed error tolerance constraint and thus their compression ratio is higher, i.e., around 1. RACE is outperformed by other DCT-based solutions in terms of compression performance, at all correlation lengths. As will be discussed shortly, this scheme may be interesting for its lightweight character in terms of energy consumption requirements. The energy cost for compression is reported in Fig. 4.4b, where $n^\star$ is varied as an independent parameter. The compression cost for all the FFT/DCT schemes is given by a first contribution, which represents the energy needed to evaluate the FFT/DCT of the input signal $x(n)$. Thus, there is a second contribution which depends on the number of transformation coefficients that are picked. Specifically, a decreasing $n^\star$ means that the signal is less correlated and, in this case, more coefficients are to be considered to meet a given error tolerance. Further, for each of them, an inverse transform has to be evaluated to check whether an additional coefficient is required. This leads to an increasing computational cost for decreasing $n^\star$.

RACE instead, as described in Section 4.2.1.5, only performs an initial Wavelet decomposition and subsequently checks the reconstruction error thanks to the coefficient selection phase along the constructed tree, without having to compute an inverse transform at each step. Hence, its energy consumption remains nearly constant while varying the correlation length $n^\star$ and is lower than that of FFT and DCT schemes. Finally, we note that FFT-based methods achieve the best performance in terms of compression ratio among all schemes of Figs. 4.3b and 4.4b (DCT-LPF is the best performing algorithm), whereas PLA schemes give the best performance in terms of energy consumption for compression (LTC is the best among them).

Applicability to real-world signals: in Table 4.1, we show the typical sampling rate and



(a)

(b)

Figure 4.4. *(a) η vs Correlation Length $n^{\star}$ and (b) η vs Energy consumption for compression for the Fourier-based methods for fixed $\varepsilon = 4\sigma_{\text{noise}}$.*

the correlation length for selected real-world signals. Luminosity and temperature data are taken from the database used in [84], readings from load sensors are taken from a structural monitoring WSN installed by WorldSensing in the Palau Sant Jordi of Barcelona (ES), whereas seismic data is obtained from the measurements in [85]. The high quality (HQ)

Signal type	Sampling rate [Hz]	Typical $n^\star$ [samples]
Indoor temperature	1/60	563
Humidity	1/600	355
Load sensors	1/5	402
Outdoor temperature	10	135
Luminosity	1/300	100
Music (HQ)	44.1 k	33
Music (LQ)	8192	4
Speech	8192	8
Seismic	150	3

Table 4.1. *Typical correlation length $n^\star$ for selected real-world signals.*

musical sample and the speech data are respectively from an excerpt of classical music by Mozart and from a sample of speech from an adult female, these datasets are available at [86]. The low quality (LQ) musical sample is from the Händel Messiah's Hallelujah Chorus.

We focus on compression schemes for WSNs that are signal-agnostic and, as such, try to approximate the signals on the fly through some modeling technique. These, are however effective for slowly varying signals, say, with correlation length larger than 50 samples. Typically, these kinds of signals are monitored by WSNs gathering climatic/environmental data or structural health. Audio signals, such as music and voice, seismic signals, or signals related to online traffic monitoring show abrupt variations, are highly non-stationarity and are characterized by very short correlation lengths (usually smaller than 10 samples). While the techniques presented here can be used for the compression of these signals, dedicated algorithms, which are outside the scope of this chapter, are expected to lead to better results.

4.3.7 Application Scenario

In this section, we evaluate the selected compression methods considering the energy consumed for transmission of typical radios in Wireless Sensor Networks (WSN) for a single- and a multi-hop network, where there is no multiple-user interference at the channel access.

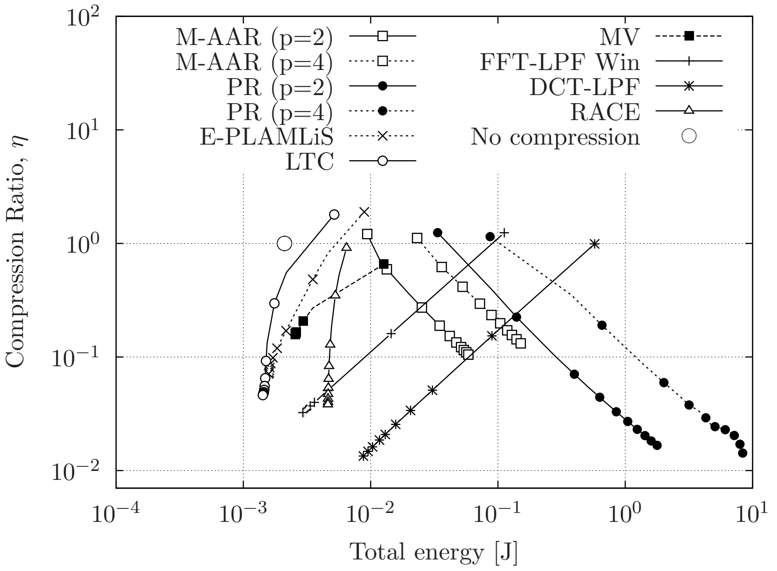


Figure 4.5. *Compression Ratio η vs Total Energy Consumption: comparison among lossy compression schemes.*

These results will be extended in Section 4.4 to Multi-hop networks with interference.

Single-hop Performance: Fig. 4.5 shows the performance in terms of Compression Ratio η *vs* Total Energy Consumption for a set of compression methods when applied to an interference-free single-hop WSN scenario. PLAMLiS was not considered as its performance is always dominated by E-PLAMLiS and we only show the performance of the best Fourier-based schemes. In both graphs the large white dot represent the case where no compression is applied to the signal, which is entirely sent to the gathering node. Note that energy savings can only be obtained for those cases where the total energy lies to the left of the no compression case. For the following results, we have set the transmission power of the radio transceiver to the maximum level, in order to show the best achievable performance when data compression is applied.

Notably, in spite of the adoption of the maximum power level, the computational energy is comparable to that spent for transmission, thus, only LTC and Enhanced PLAMLiS can achieve some energy savings (see Fig. 4.5). All the other compression methods entail a high number of operations and, in turn, perform worse than the no compression case in terms of overall energy expenditure. This remarkable result is a consequence of that, as mentioned

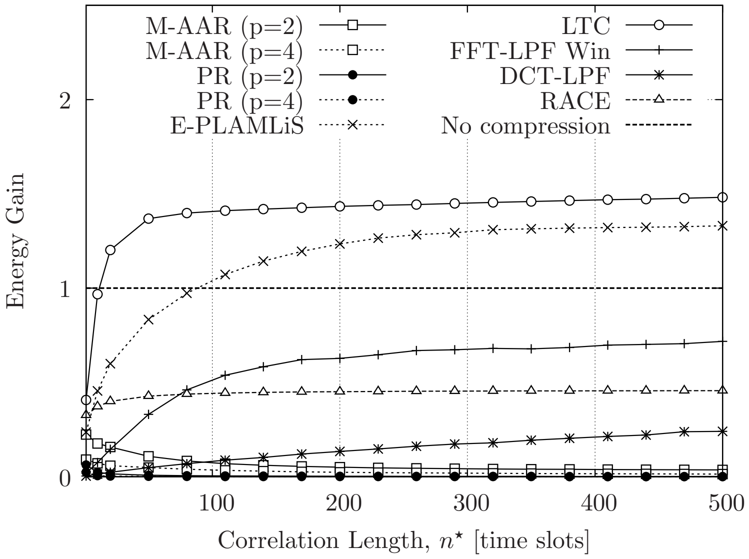


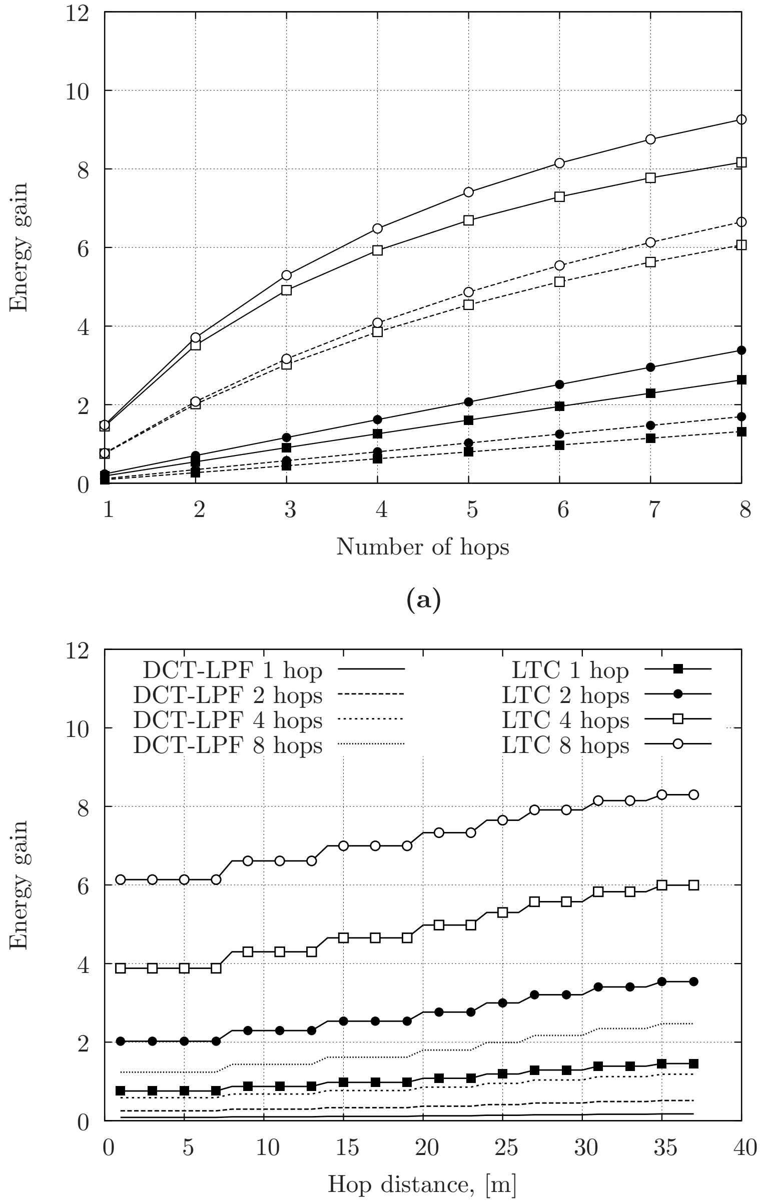
Figure 4.6. *Maximum Energy Gain vs Correlation Length $n^\star$ for a single-hop scenario.*

in Section 4.3.3, using current technologies, only a few hundred CPU instructions can be executed to compress a single bit of information and be energy efficient.

The total energy gain, defined as the ratio between the energy spent for transmission in the case with no compression and the total energy spent for compression and transmission using the selected compression techniques, is shown in Fig. 4.6. The method that offers the highest energy gain is LTC, although other methods such as DCT-LPF can achieve better compression performance (see Fig. 4.5). Note that in this scenario the total energy is highly influenced by the computational cost. Thus, the most lightweight methods, such as LTC and enhanced PLAMLiS, perform best.

Multi-hop Performance: in Fig. 4.7 we focus on multi-hop networks, and evaluate whether further gains are possible when the compressed information has to travel multiple hops to reach the data gathering point. In this case, both transmission and reception energy are accounted for at each intermediate relay node. In the following, only LTC and DCT-LPF are shown, as these are the two methods that respectively perform best in terms of complexity and compression efficiency.

In Fig. 4.7a, we set the error tolerance $\varepsilon = 4\sigma_{noise}$, the correlation length of the input signal $n^\star \in \{300, 500\}$ and we evaluate the possible gains for the maximum and the minimum



(a)

(b)

Figure 4.7. *(a) Energy gain vs number of hops for* $\varepsilon = 4\sigma_{\text{noise}}$*. Solid lines are used to indicate maximum transmission power, dashed lines to indicate minimum transmission power. Results for DCT-LPF are shown with black filled markers, whereas white filled markers are used for LTC. The type of marker indicates the correlation length of the input signal, specifically:* (□, ■) *for* $n^\star = 300$*,* (○, •) *for* $n^\star = 500$*. (b) Energy gain vs hop distance for LTC and DCT-LPF with* $\varepsilon = 4\sigma_{\text{noise}}$ *and* $n^\star = 300$*.*

transmission power levels, so as to respectively obtain the upper and lower bounds on the achievable performance. As shown in this figure, the energy gain increases with the number of hops. This is because, although the energy spent for the compression at the source node is comparable to that spent for the transmission, the compression cost (compression energy) is only incurred at the source node; while each additional relay node only has to send compressed data. We also note that DCT-LPF is not energy efficient in single-hop scenarios, but it can actually provide some energy gains when the number of hops is large enough (e.g., larger than 2), and the transmission power is set to the maximum level. For the minimum transmission power, DCT-LPF starts being energy efficient only after $5 - 6$ hops, see Fig. 4.7a.

In Fig. 4.7b we show the maximum achievable energy gain versus the distance between hops. Given the distance, the transmission power is selected according to the Friis path loss formula (with path loss exponent $\alpha = 3.5$, which is typical for WSNs [87]), considering the transmission power levels and the receiver sensitivity $P_{th} = -95$ dBm of the CC2420 transceiver [81]. For each value of the distance, we evaluated the energy gain using the minimum transmission power level that leads to a received power above P_{th}. As shown in Fig. 4.7b, the energy gain increases with the distance, as the transmission power becomes progressively higher of that needed for compression. This effect becomes more pronounced when the number of hops is increased, as the relay nodes only have to forward the data (no processing), thus benefiting from the smaller number of bits to be received and transmitted.

4.3.8 Numerical Fittings

In this section, we provide close-formulas to accurately relate the achievable compression ratio η to the relative error tolerance ξ and the computational complexity, N_c, which is expressed in terms of number of clock cycles per bit to compress the input signal $x(n)$. These fittings have been computed for the best compression methods, namely, LTC and DCT-LPF.

Note that, until now, we have been thinking of η as a performance measure which depends on the chosen error tolerance $\varepsilon = \xi\sigma_{noise}$. This amounts to considering ξ as an input parameter for the compression algorithm. In the following, we approximate the mathematical relationship between η and ξ, by conversely thinking of ξ as a function of η, which is now our input parameter. N_c can as well be expressed as a function of η.

We found these relationships through numerical fitting, running extensive simulations with synthetic signals. The relative error tolerance ξ can be related to the compression ratio η through the following formulas:

$$\xi(n^\star, \eta) = \begin{cases} \dfrac{p_1\eta^2 + p_2\eta + p_3}{\eta + q_1} & \text{LTC} \\ \dfrac{p_1\eta^4 + p_2\eta^3 + p_3\eta^2 + p_4\eta + p_5}{\eta + q_1} & \text{DCT-LPF}\,, \end{cases} \tag{4.2}$$

where the fitting parameters p_1, p_2, p_3, p_4, p_5, and q_1 depend on the correlation length $n^\star$ and are given in Table 4.2 for LTC and DCT-LPF. These fitting formulas have been validated against real world signals measured from the environmental monitoring WSN testbed deployed on the ground floor of the Department of Information Engineering (DEI), University of Padova, Italy [88]. This dataset consists of measures of temperature and humidity, sensed with a sampling interval of 1 minute (temperature) and 10 minutes (humidity) for 6 days. Correlation lengths are $n_T^\star = 563$ and $n_H^\star = 355$ for temperature and humidity signals, respectively. The empirical relationships of Eq. (4.2) are shown in Fig. 4.8a and 4.8b through solid and dashed lines, whereas the markers indicate the performance obtained applying LTC and DCT-LPF to the considered real datasets. As can be noted from these plots, although the numerical fitting was obtained for synthetic signals, Eq. (4.2) closely represents the actual tradeoffs. Also, with decreasing $n^\star$ the curves relating ξ to η remain nearly unchanged in terms of functional shape but are shifted toward the right. Finally, we note that the dependence on $n^\star$ is particularly pronounced at small values of $n^\star$, whereas the curves tend to converge for increasing correlation length (larger than 110 in the figure).[2]

For the computational complexity, we found that N_c scales linearly with η for both LTC and DCT-LPF. Hence, N_c can be expressed through a polynomial as follows:

$$N_c(n^\star, \eta) = \alpha\eta + \gamma n^\star + \beta\ . \tag{4.3}$$

N_c exhibits a linear dependence on both $n^\star$ and η; the fitting coefficients are shown in Table 4.3. We remark that the computational complexity as given by (4.3) is that achievable by a temporal compressor configured with a compression ratio η, that keeps the reconstruction error bounded according to the error tolerance $\varepsilon = \xi(n^\star, \eta)\sigma_{\text{noise}}$ (see (4.2)). Note that, differently from Fig. 4.4, this reasoning entails the compression of our data without fixing beforehand the error tolerance ε, which instead directly follows from η and $n^\star$.

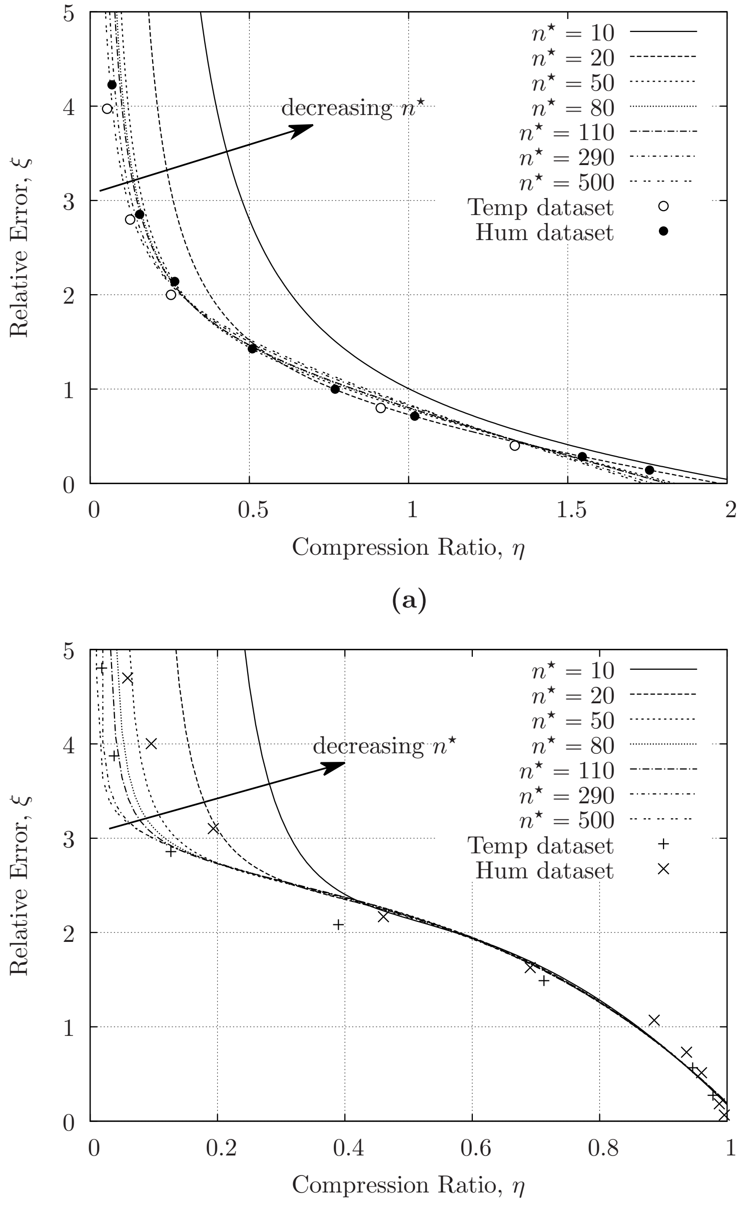
[2] Note also that the there is a lower bound on the achievable reconstruction accuracy as the signal correlation $n^\star$ increases. This is due to the noise that is superimposed to the useful signal.

Compression	$n^\star$	Fitting coefficients					
Method		p_1	p_2	p_3	p_4	p_5	q_1
LTC	10	-0.35034	0.27640	0.92834	–	–	-0.15003
	20	-0.51980	0.86851	0.31368	–	–	-0.09245
	50	-0.80775	1.38842	0.17465	–	–	-0.03705
	80	-0.85691	1.45560	0.18208	–	–	-0.02366
	110	-0.86972	1.46892	0.19112	–	–	-0.01736
	290	-0.97242	1.61970	0.17280	–	–	-0.00747
	500	-1.03702	1.70305	0.17466	–	–	0.00267
DCT-LPF	10	2.05351	-12.70381	14.49624	-4.52198	0.82292	-0.16165
	20	-0.92752	-3.07506	3.07560	1.06902	0.02898	-0.09025
	50	-1.90344	-0.17491	-0.13500	2.43821	-0.03826	-0.03929
	80	-2.59629	1.41404	-1.40970	2.81971	-0.04122	-0.02667
	110	-2.57150	1.43655	-1.51646	2.87138	-0.02747	-0.01913
	290	-3.43806	3.17964	-2.67444	3.13226	-0.01531	-0.00848
	500	-3.99007	4.17811	-3.22636	3.22590	-0.01102	-0.00560

Table 4.2. *Fitting coefficients for* $\xi(n^\star, \eta)$.

Compression	Fitting coefficients		
Method	α	β	γ
LTC	16.1	105.4	$3.1 \cdot 10^{-16}$
DCT-LPF	$48.1 \cdot 10^{3}$	82.3	$-2 \cdot 10^{-13}$

Table 4.3. *Fitting coefficients for* $N_c(n^\star, \eta)$.



(a)

(b)

Figure 4.8. *Fitting functions $\xi(n^\star, \eta)$ vs experimental results: (a) LTC, (b) DCT-LPF. The correlation $n^\star$ of the considered datasets for temperature and humidity is* 563 *and* 355 *samples, respectively.*

Further, in (4.3) the dependence on $n^\star$ is much weaker than that on η and for practical purposes can be neglected without loss of accuracy. For this reason, in the remainder of this

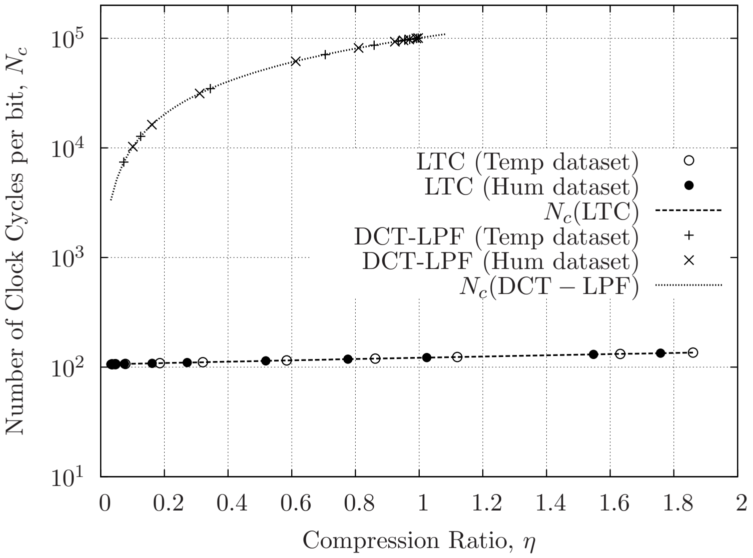


Figure 4.9. *Fitting functions $N_c(\eta)$ vs experimental results.*

section we consider the simplified relationship:

$$N_c(\eta) = \alpha\eta + \beta \; . \tag{4.4}$$

The accuracy of Eq. (4.4) is verified in Fig. 4.9, where we plot our empirical approximations against the results obtained for the real world signals described above. The overall energy consumption is obtained as $N_b(x)N_c(\eta)E_0$.

Tradeoffs: in the following, we use the above empirical formulas to generalize our results to any processing and transmission technology, by separating out technology dependent and algorithm-dependent terms. Specifically, a compression method is energy efficient when the overall cost for compression ($E_c(x)$) and transmission of the compressed data ($E_{Tx}(\hat{x})$) is strictly smaller than the cost associated with transmitting $x(n)$ uncompressed ($E_{Tx}(x)$). Mathematically, $E_c(x) + E_{Tx}(\hat{x}) < E_{Tx}(x)$. Dividing both sides of this inequality by $E_{Tx}(x)$ and rearranging the terms leads to:

$$\frac{E_{Tx}(x)}{E_c(x)} = \frac{E'_{Tx}[\ell]N_b(x)}{E_0 N_c N_b(x)} > \frac{1}{1-\eta} \; ,$$

where the energy for transmission $E_{Tx}(x)$ is expressed as the product of the energy expenditure for the transmission of a bit $E'_{Tx}[\ell]$ (for the selected output power level $\ell \in \{1, \ldots, 8\}$) and the number of bits of $x(n)$, $N_b(x)$. The energy for compression is decomposed in the product of three terms: 1) the energy spent by the micro-controller in a clock cycle E_0, 2) the number of clock cycles performed by the compression algorithm per (uncompressed) bit of $x(n)$, N_c and 3) the number of bits composing the input signal $x(n)$, $N_b(x)$. With these energy costs and the above fitting Eq. (4.4) for N_c we can rewrite the above inequality so that the quantities that depend on the selected hardware architecture appear on the left hand side, leaving those that depend on algorithmic aspects on the right hand side. The result is:

$$\frac{E'_{Tx}[\ell]}{E_0} > \frac{N_c(\eta)}{1-\eta} = \frac{\alpha\eta + \beta}{1-\eta} \,, \tag{4.5}$$

where α and β are the algorithmic dependent fitting parameters indicated in Table 4.3. Eq. (4.5) can be used to assess whether a compression scheme is suitable for a specific device architecture. As an example, for the considered WSN architecture we have that $E'_{Tx}[8] =$ 230 nJ for the selected CC2420 radio for its highest transmission power , whereas for the TI MSP430 we have $E_0 = 0.726$ nJ and their ratio is $E'_{Tx}[8]/E_0 \simeq 316$. The numerical evaluation of the RHS of (4.5) for DCT-LPF reveals that this compression scheme is inefficient for any value of η, i.e., the overall energy expenditure due to transmission plus compression is higher than the energy spent in the case where compression is not applied. Instead, LTC provides energy savings for $\eta \leqslant 0.6$, that using the function $\xi(n^\star, \eta)$ for LTC can be translated into the corresponding (expected) error performance. Note that the knowledge of $n^\star$ is needed for this last evaluation. These results can be generalized to any other device technology, by comparing the RHS of (4.5) against the corresponding ratio $E'_{Tx}[\ell]/E_0$ and checking whether the inequality in (4.5) holds.

4.4 Performance Comparison for Interference-limited Multi-Hop Networks

In this section we generalize our findings to multi-hop WSN where data is routed along a tree and eventually collected by a sink node. In doing so, we model the channel access dynamics in terms of transmission schedules, idle times and collisions, accounting for the corresponding energy and delay terms. For the sake of analytical tractability, we account for static routing paths. The main question that we try to answer here is whether additional benefits arise when further protocol inefficiencies are accounted for. Especially, we are concerned with the benefits that may be achieved for DCT schemes, which provide the best

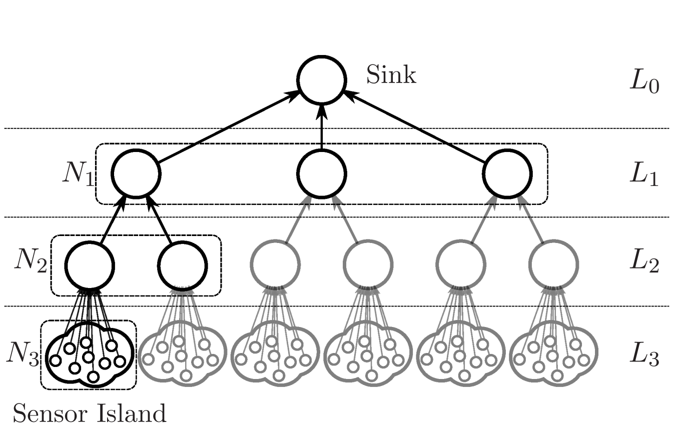


Figure 4.10. *Multi-hop WSN scenario.*

compression performance but that, as we have seen above, may be inefficient for single-hop networks when the channel access is idealized.

4.4.1 Analysis for Interference Limited Networks

Scenario: we consider the multi-hop WSN of Fig. 4.10 where the field readings are gathered by the sensors placed in a number of WSN islands (at level L_{K-1}) and then routed to the data collector node (the WSN sink, at level L_0), through a data collection tree. This tree is organized according to a hierarchical structure with K levels, $L_0, L_1, \ldots, L_{K-1}$, whereby N_k is the number of children nodes for a root located at level $k-1$ with $k = 1, \ldots, K-1$ and $N_K = 0$. Thus, $N(k) = \prod_{i=1}^{k} N_i$, $k \geqslant 1$ is the total number of nodes at level k, $N(0) = 1$, and $\sum_{k=0}^{K-1} N(k)$ is the total number of sensor nodes, including the sink.

To reduce the interference among the data forwarding processes taking place in the different levels of the tree we adopt a pipelining scheduling technique as done in, e.g., [89]. Starting with the lowest level L_{K-1}, the data collection protocol works in rounds of T seconds that are further subdivided into 3 sub-rounds, S_1, S_2 and S_3 of $T/3$ seconds each. During S_1, the $N(K-1)$ nodes at level L_{K-1} compete for the channel to send their data to their respective roots located in L_{K-2}, which act as receivers. During the next sub-round S_2, the nodes in L_{K-2} contend for the channel to forward their data to the nodes in the next level L_{K-3}, the nodes in L_{K-1} sleep and those in L_{K-3} act as receivers. In the final sub-round S_3, the nodes in L_{K-3} forward their data toward their upper level $K-4$ and those in levels $K-1$ and $K-2$ sleep. In the next sub-round (again of type S_1), the nodes at level L_{K-4} forward their data and those in L_{K-1} can concurrently transmit, being outside their interference range. This procedure is iteratively applied to each level so that the nodes that are three levels apart share the same schedule and concurrently transmit in the same sub-round. The nodes in each level $k = 1, 2, \dots, K-2$ will receive in one sub-round, transmit in the next one, and stay silent in the last. The nodes in the last level L_{K-1} transmit in a sub-round and sleep during the following two. We further assume that the nodes belonging to the same level but to a different sub-tree do not interfere with each other.

The nodes in the sensor islands (L_{K-1}) are the only ones that generate endogenous traffic, each according to a Poisson process with rate λ_{K-1} pkt/s, which depends on their sampling rate for the underlying physical process. The rates for the nodes at the upper levels depend on the aggregated traffic, which is recursively computed accounting to MAC contentions and routing. The channel contentions during the transmission sub-rounds are governed through a protocol that is similar to S-MAC [3], extending the modeling approach of [90]. The authors of the latter paper present a model for single-hop networks, where all nodes are allowed to talk to any other node. Below, we extend their analysis for the scenario of Fig. 4.10, where all the nodes in a certain level can only communicate with their respective root (located in the next level toward the sink). Hence, a mathematical analysis is developed to characterize the performance within each level, aggregating the contributions from L_{K-1} up to L_0 so as to obtain the overall network performance, i.e., from the WSN islands to the sink. Our extension is reported in what follows.

Analysis: we assume that each node has a finite FIFO queue that can store up to Q packets. During a transmission sub-round, only the nodes with non empty queue wake up and participate in the channel contention. In order to maintain synchronization across nodes, in S-MAC a fixed interval at the beginning of each active period is reserved for the exchange of SYNC packets. After this phase S-MAC exploits an RTS/CTS/DATA/ACK handshake to guarantee the successful transmission of the data. Specifically, each node access the channel after a backoff time picked uniformly at random from a fixed length contention window of $W\delta$ seconds, where W is the contention window size and δ is the time duration of an access slot. The first node that accesses the channel (the winner of the contention) sends an RTS packet and remains active until the completion of the RTS/CTS/DATA/ACK handshake, in which case its packet is successfully transmitted. All the remaining nodes go to sleep as soon as they overhear an RTS packet and wake up at the next transmission sub-round. If multiple nodes attempt to access the channel in the same access slot, their RTS packets will collide and the data packets of these nodes are dropped, causing a loss event. We also assume that the nodes remains active for the entire duration of their reception sub-rounds.

The analysis that follows is applied to each level of the discussed multi-hop WSN, starting from the sensor islands and recursively moving towards the sink. The MAC queue of each node in level $k = 0, 1, \dots, K-1$ is modeled through a Markov chain, with transition probabilities depending on the corresponding arrival rate λ_k and on the probability $p \triangleq p_s + p_f$ of removing one packet from the queue (either due to a successful transmission, w.p. p_s, or due to a collided one, w.p. p_f).[3] For the levels above L_{K-1} we calculate λ_k aggregating the traffic of the N_{k+1} underlying nodes and accounting for the respective packet losses, as follows:

$$\lambda_k = \text{PDR}_{k+1} \cdot N_{k+1} \cdot \lambda_{k+1} \qquad \forall\, k = 0, 1, \dots, K-2\,, \tag{4.6}$$

where PDR_{k+1} is the average packet delivery ratio for the level immediately below. The packet generation rate for the nodes in the WSN islands (level L_{K-1}) is application dependent as is denoted by γ pkt/s. Accounting for the reduction in the number of packets sent due to the utilization of temporal compression algorithms, these nodes generate endogenous data traffic according to rate λ_{K-1}, where:

$$\lambda_{K-1} = \begin{cases} \gamma & \text{no} - \text{compression} \\ \gamma\eta & \text{compression}\,. \end{cases} \tag{4.7}$$

Following [90], given the Markov model for the MAC queues, for each node the stationary probability of empty queue π_o can be evaluated as a function of p, i.e., $\pi_o = f(p)$.[4] Also, given the number of nodes in the same group (i.e., sharing the same root node), the probability of winning a contention p can be expressed as a further function of π_o. i.e., $p = p_s + p_f = g(\pi_o)$. In fact, note that p_s and p_f depend on the number of nodes at level k that have at least one packet in their buffer and thus transmit toward the same root located at level $k-1$. This number is a r.v. having a binomial mdf with parameters N_k and π_0, i.e., $B(N_k, \pi_o)$. For each

[3] We remark that p, p_s and p_f are to be calculated for each level k, according to the aggregated input traffic coming from the previous level $k-1$. Here, we omit their explicit dependence on k for the sake of readability.

[4] In what follows, in addition to p_s and p_f, the index k is also omitted from p and π_o for the sake of a more concise and readable notation.

level $k = 1, \ldots, K-1$, the pair (p, π_o) is evaluated numerically by finding the only point where the two curves $f(p)$ and $g(\pi_o)$ intersect. This returns the (steady-state) operational point for the MAC system, which is successively used to evaluate the performance of interest, in particular: 1) the Packet Delivery Ratio, 2) the Delay and 3) the Mean Energy Consumption.

1) Packet Delivery Ratio (PDR): the PDR for a given level k, (PDR_k), is defined as the ratio of successfully delivered packets over the total number of generated data packets and derived as follows:

$$\mathrm{PDR}_k = \frac{(1-\pi_0)p_s}{\lambda_k T} .$$

For our multi-hop network, the Total Packet Delivery Ratio from level L_{K-1} up to the sink is obtained as $\prod_{k=1}^{K-1} \mathrm{PDR}_k$.

2) Delay: the delay for a given level k, D_k, is defined as the number of time slots needed to deliver a data packet to the next hop. The average delay for each level can be obtained through the analysis in [90]; the total average delay is obtained through the aggregation of the delays experienced in each level from the WSN islands all the way to the sink.

3) Mean Energy Consumption: for the energy consumption analysis, we recall the features of our version of S-MAC.

1. The time slot is divided in three sub-rounds, all the nodes in the same level share the same schedule and are synchronized.

2. In the transmission sub-round only the nodes with non empty queue wake up and contend for the channel.

3. The nodes that wake up perform a SYNC phase, which is followed by a channel contention. Only the winner of this contention transmits a data packet, whereas all other nodes go to sleep as soon as they overhear an RTS packet or detect a collision.

4. The winner of the contention goes to sleep as soon as the data packet is acknowledged (in the case of a successful transmission), or after an RTS collision (when two or more node access the channel in the same access slot).

5. The nodes remain in listening(sleep) mode for the whole duration of their reception(sleep) sub-round.

With these assumptions, we define the Mean Energy per time slot E as the sum of the mean energy spent in the three sub-rounds:

$$E = E_{\mathrm{S}_1} + E_{\mathrm{S}_2} + E_{\mathrm{S}_3} \; . \tag{4.8}$$

Without loss of generality we hereby consider S_1 as the transmission sub-round, S_2 as the reception sub-round and S_3 as the sleep sub-round.

The average energy consumption in the transmission sub-round can be further factorized as:

$$E_{\mathrm{S}_1} = E_{\mathrm{sync}} + E_{\mathrm{data}} + E_{\mathrm{sleep}} \; , \tag{4.9}$$

where E_{sync}, E_{data} and E_{sleep} respectively account for the average energy consumption in the SYNC, the data transmission and the sleep phase occurring within a sub-round of type S_1. Assuming that a the duration of a SYNC phase is T_{sync}, that the transmission of a SYNC packet takes t_{SYNC} and that a node transmits a SYNC packet every N_{sync} time slots, E_{sync} is obtained as:

$$E_{\mathrm{sync}} = \frac{t_{\mathrm{SYNC}} P_{\mathrm{Tx}} + (T_{\mathrm{sync}} - t_{\mathrm{SYNC}}) P_{\mathrm{Rx}} + T_{\mathrm{sync}} P_{\mathrm{Rx}} (N_{\mathrm{sync}} - 1)}{N_{\mathrm{sync}}} \; , \tag{4.10}$$

where P_{Tx} and P_{Rx} are the radio power consumption for transmission and reception, respectively.

In order to evaluate E_{data} and E_{sleep}, we consider a tagged node having a packet to send and look at the duration of its backoff window in the following three cases: C1) the node successfully transmits a data packet, C2) the RTS sent by this node collides, and C3) the node goes back to sleep as it detects channel activity before its backoff timer expires. The durations of these events is referred to as W_s, W_c and W_t for cases C1, C2 and C3, respectively (expressed in number of access slots). For a given number of nodes that take part in the contention, N_o, we obtain:

$$W_s(N_o) = \sum_{i=0}^{W-1} \frac{i \cdot \frac{N_o}{W} \cdot \left(\frac{W-i-1}{W}\right)^{N_o-1}}{\sum_{j=0}^{W-1} \frac{N_o}{W} \cdot \left(\frac{W-j-1}{W}\right)^{N_o-1}} \; , \tag{4.11}$$

$$W_c(N_o) = \sum_{i=0}^{W-1} \frac{i \cdot \left[\left(\frac{W-i}{W}\right)^{N_o} - \left(\frac{W-i-1}{W}\right)^{N_o} - \frac{N_o}{W} \cdot \left(\frac{W-i-1}{W}\right)^{N_o-1}\right]}{\sum_{j=0}^{W-1} \left(\frac{W-j}{W}\right)^{N_o} - \left(\frac{W-j-1}{W}\right)^{N_o} - \frac{N_o}{W} \cdot \left(\frac{W-j-1}{W}\right)^{N_o-1}} \; , \tag{4.12}$$

$$W_t(N_o) = \sum_{i=0}^{W-1} \frac{i \cdot \left[\left(\frac{W-i}{W}\right)^{N_o} - \left(\frac{W-i-1}{W}\right)^{N_o} \right]}{\sum_{j=0}^{W-1} \left(\frac{W-j}{W}\right)^{N_o} - \left(\frac{W-j-1}{W}\right)^{N_o}} \ . \tag{4.13}$$

Inside the sum of (4.11) we have the conditional probability that the tagged node wins the contention, accessing the channel in slot i, whereas all other nodes that participate in the contention pick an access slot greater than i, given that the contention is successful. The remaining equations (4.12) and (4.13) are obtained similarly, accounting for C2 and C3. The expected values for these durations are obtained averaging W_s, W_s and W_s over the possible values of N_o, as:

$$\mathrm{E}\left[W_{\{s,t,c\}}\right] = \sum_{n=1}^{N} \binom{N}{n} \cdot \pi_o^{N-n} \cdot (1-\pi_o)^n \cdot W_{\{s,t,c\}}(n) \ . \tag{4.14}$$

The mean durations in seconds is evaluated as $T_{\{s,c,t\}} = \mathrm{E}\left[W_{\{s,t,c\}}\right] \cdot \delta$.

Finally E_{data} and E_{sleep} can be computed accounting for cases C1 (w.p. $(1-\pi_o)\cdot p_s$), C2 (w.p. $(1-\pi_o)\cdot p_f$), C3 (w.p. $(1-\pi_o)\cdot(1-p_s-p_f)$) and a further case C4) where the tagged node does not participate in the contention as its queue is empty (w.p. π_o):

$$\begin{aligned} E_{\text{data}} = {} & (1-\pi_o) \cdot p_s \cdot [(t_{\text{RTS}} + t_{\text{DATA}}) \cdot P_{\text{Tx}} \\ & + (t_{\text{CTS}} + t_{\text{ACK}} + T_s) \cdot P_{\text{Rx}} + E_c] \\ & + (1-\pi_o) \cdot p_f \cdot [t_{\text{RTS}} \cdot P_{\text{Tx}} + (t_{\text{CTS}} + T_c) \cdot P_{\text{Rx}} + E_c] \\ & + (1-\pi_o) \cdot (1-p_s-p_f) \cdot [(t_{\text{RTS}} + T_t) \cdot P_{\text{Rx}}] \\ & + \pi_o \cdot 0 \ , \end{aligned} \tag{4.15}$$

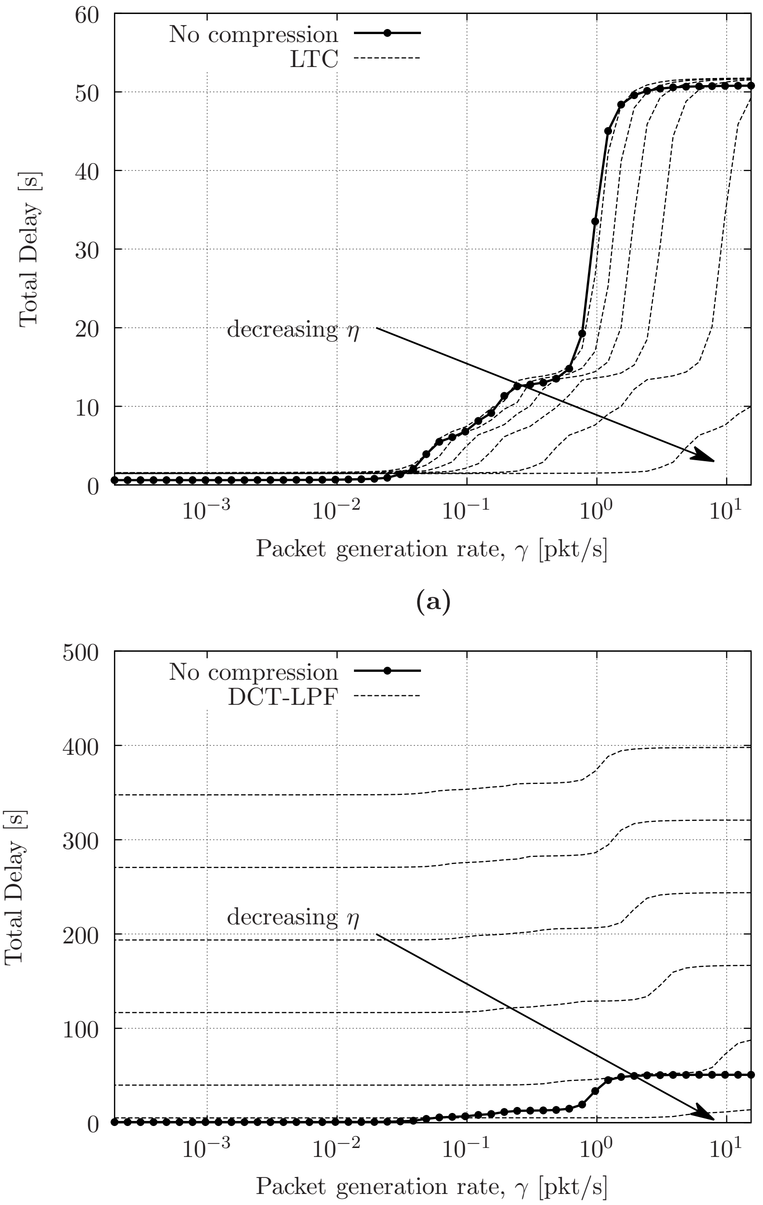
$$\begin{aligned} E_{\text{sleep}} = {} & (1-\pi_o) \cdot p_s \cdot (T/3 - T_{\text{sync}} - t_{\text{RTS}} - t_{\text{CTS}} \\ & - t_{\text{DATA}} - t_{\text{ACK}} - T_s) \cdot P_{\text{Sl}} \\ & + (1-\pi_o) \cdot p_f \cdot (T/3 - T_{\text{sync}} - t_{\text{RTS}} - t_{\text{CTS}} - T_c) \cdot P_{\text{Sl}} \\ & + (1-\pi_o) \cdot (1-p_s-p_f) \cdot (T/3 - T_{\text{sync}} - t_{\text{RTS}} \\ & - T_t) \cdot P_{\text{Sl}} + \pi_o \cdot (T/3) \cdot P_{\text{Sl}} \ . \end{aligned} \tag{4.16}$$

Here, t_{RTS}, t_{DATA}, t_{CTS}, t_{ACK} represent the time for transmitting an RTS, DATA, CTS and an ACK, respectively. P_{Sl} is the power consumption of the radio transceiver in the sleep mode. E_c in (4.15) is the energy consumption for the compression of a data packet, evaluated according to the packet length, the required number of CPU cycles (see (4.4)) and the energy consumption per CPU cycle E_0 ($E_c = 0$ if data is sent uncompressed). The probabilities p_s, p_f and π_o are computed as in [90].

4.4.2 Results

In this section we provide some results for the scenario of Fig. 4.10, where we consider a network with $K = 4$ levels, and with $N_1 = 3$, $N_2 = 2$ and $N_3 = 30$, leading to a total number of 190 nodes. The packet generation rate at the lowest level λ_3 is obtained considering an average inter-sampling time for the underlying physical phenomenon $I_s \in [10^{-3}, 10^2]$ seconds. Moreover, considering a packet payload size of 127 bytes and 16 bits per sample, the resulting packet generation rate for the endogenous traffic is $\gamma = 1/(63.5 I_s)$ packets/s. For the compression methods, we focus on LTC and DCT-LPF as from Section 4.3.8 we know that these respectively perform best in terms of computational complexity and compression efficiency. Thus, we compare their performance at different compression ratios, $\eta \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9\}$, and without compression. The queue length of each node is set to $Q = 10$ pkts, its contention window to $W = 50$ access slots and the duration of an access slot is set to $\delta = 0.2$ ms. For the computation of the energy terms (compression and radio activity) we consider the power consumptions of the CC2420 radio transceiver and of the MSP430 micro-controller. In the results that follows, different performance metrics are shown as a function of the compression ratio η. As per our working methodology in this chapter, each value of η (and, in turn, each curve in the following graphs) is characterized by a corresponding error tolerance $\xi(n^\star, \eta)$, attainable from the correlation length $n^\star$ and (4.2). A decreasing η corresponds to an increasing error as shown in Fig. 4.8.

In Figs. 4.11a 4.11b we plot the the total delivery delay as a function of the packet generation rate for each node in the WSN islands, γ. We consider the total delay incurred in compressing the data and transmitting them through multiple hops, as per the scenario of Fig. 4.10. For LTC, the delay is slightly longer than no compression at low data traffic (roughly $\gamma \leqslant 0.02$ packets/second), where the latter outperforms LTC by about 1.5 seconds (due to the additional execution time for compression). As γ gets larger, the delay performance when sending the data uncompressed is substantially impacted and this is due to the increased level of congestion in the network (i.e., number of collisions) and, in turn, to the longer waiting time that the packets experience in the network queues. The same fact



(a)

(b)

Figure 4.11. *Total Delay [s] vs packet generation rate γ at the lowest level for: (a) LTC; (b) DCT-LPF.*

occurs for LTC, where however the level of congestion is milder due to the lower amount of data that is injected into the network when compression is applied. Notably, LTC remains quite lightweight for all values of η and, thus, for all the corresponding values of ξ. This is because LTC requires a small number of operations (in fact, with LTC a single pass on the

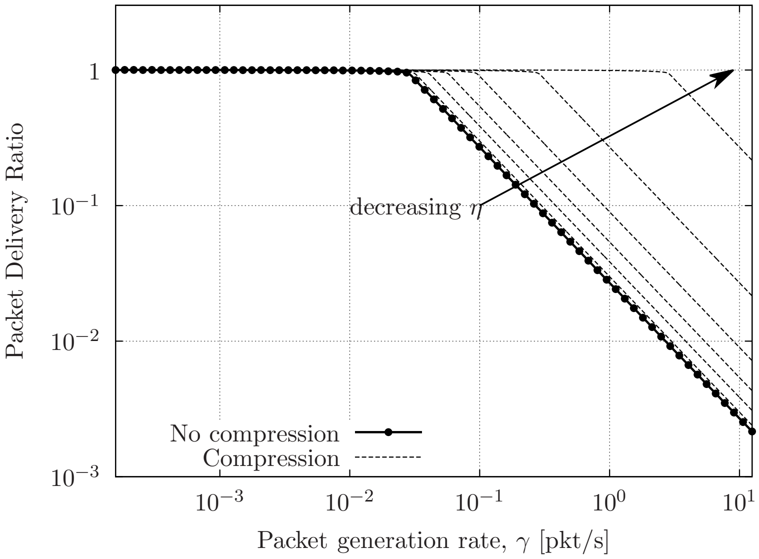


Figure 4.12. *Packet Delivery Ratio vs packet generation rate γ at the lowest level.*

input signal is required to provide a compressed sequence meeting the required reconstruction fidelity ξ). DCT-LPF shows a totally different behavior due to our implementation of this scheme, see Section 4.2.1.2, whereby we iteratively check the reconstruction quality at the source. In general, one may use a different approach, by performing a single DCT transform for each data frame and retaining a certain number of coefficients, so as to obtain to the wanted compression performance, irrespective of the resulting signal representation accuracy. In this case, the delay for DCT would only be slightly longer than that of LTC in Fig. 4.11a. However, no control on the reconstruction quality can be assured in this case.

The Packet Delivery Ratio is shown in Fig. 4.12. When the arrival rate is small, all the packets are successfully delivered; conversely, as the arrival rate becomes larger, the queue lengths increase and the nodes start dropping packets. With temporal compression, the number of packets in the network is reduced and this leads to a higher PDR.

In Fig. 4.13, we investigate the energy balance arising from the tradeoff,compression versus transmission, by showing the overall average energy consumption over the entire network . Notably, for LTC, from Fig. 4.13a we see that the additional cost incurred in compressing the data is well counterbalanced by the higher efficiency of the channel access procedure due to the reduction in the data traffic, i.e., fewer packet collisions and more sleeping opportunities for the sensor nodes. However, this results does not hold when the compression method is DCT-LPF as in this case the cost associated with the compression of the data is much higher

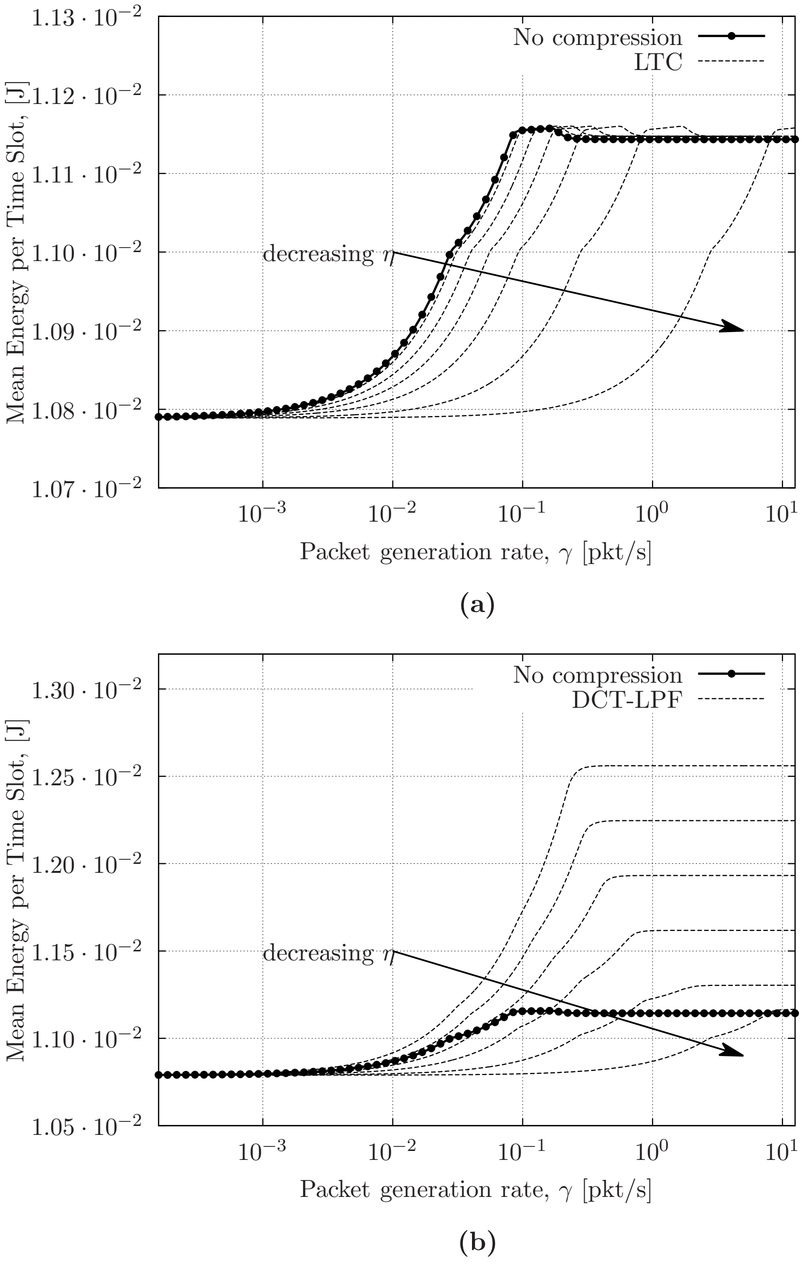


Figure 4.13. *Mean energy per time slot [J] vs packet generation rate γ at the lowest level: (a) LTC; (b) DCT-LPF.*

and some energy gain can only be achieved if the compression ratio is significant. Given these results, we advocate the use of linear compression methods such as LTC, as these are likely to lead to energy savings. The use of other methods should be carefully evaluated, as they are energy-inefficient in most practical cases and can lead to unacceptable delays .

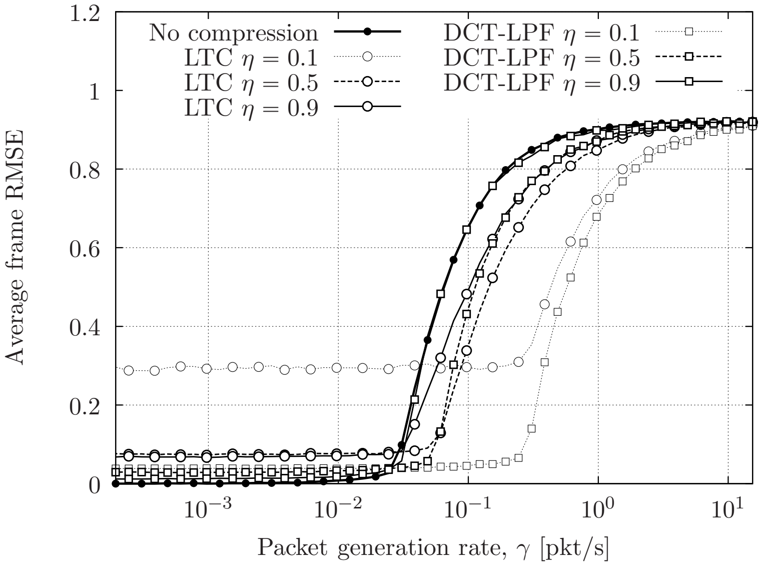


Figure 4.14. *Average frame RMSE vs packet generation rate γ for an input signal with $n^\star = 290$.*

Finally, in Fig. 4.14 we focus on the reconstruction performance for the network scenario of Fig. 4.10 in the presence of packet losses due to collisions and without accounting for any packet retransmission mechanism. To this end, we define a new reconstruction error metric, termed *frame Root Mean Square Error (RMSE)*, which is evaluated at the receiver as the RMSE for every N data samples. Since in both the compressed and uncompressed case the encoded data (corresponding to a window of N input samples) can be fragmented into multiple packets, we devised a simple loss tolerant decoding scheme for each method. In particular, 1) in the uncompressed case, whenever a packet is lost, the decoder uses the last valid received sample as its replacement. 2) For LTC, we assume that fragmentation only occurs between subsequent segment extremes $(n_i, x(n_i))$ (see Fig. 4.2) but that, for each of them, no splitting of n_i and $x(n_i)$ occurs across different packets. In this way, the receiver can always reconstruct the segments starting from a valid set of points (although in the case of a packet loss a longer segment will be used to represent the lost data points). 3) For DCT, the lost coefficients are considered equal to zero when the inverse transform is applied at the receiver.

As shown in Fig. 4.14, when the packet generation rate γ is small ($\gamma < 10^{-2}$ in the figure), the frame RMSE is constant and only depends on η and on the selected compression scheme. In this case, LTC shows a higher average RMSE than DCT-LPF for the same compression ratio η, as seen in Section 4.3.8. As γ increases beyond a certain threshold γ_{th}, the transmission channel gets saturated and, in turn, packets start being lost due to collisions. Note that this occurs earlier for the uncompressed case and for those cases where the compression does not effectively reduce the amount of traffic in the network (e.g., $\eta = 0.9$). This leads to an abrupt increase of the frame RMSE as γ increases beyond the channel saturation point γ_{th}.

On the other hand, when the compression schemes are configured so as to effectively reduce the amount of data that is transmitted over the network ($\eta > 0.5$ in Fig. 4.14), we have that γ_{th} moves to the right and this produces a beneficial effect in terms of frame RMSE, which remains constant and acceptable for a wider range of transmission rates. As an example, when the compression ratio is $\eta = 0.1$, the signal reconstruction error at the receiver can be kept small up to a packet generation rate that is roughly one order of magnitude larger than γ_{th} in the uncompressed case.

4.5 Conclusions

In this chapter we have systematically compared lossy compression algorithms for constrained sensor networking, by investigating whether energy savings are possible depending on signal statistics, compression performance and hardware characteristics. Our results reveal that, for wireless transmission scenarios, the energy required by compression algorithms has the same order of magnitude of that spent for transmission at the physical layer. In this case, the only class of algorithms that provides some energy savings is that based on piecewise linear approximations, as these algorithms have the smallest computational cost. We have additionally obtained fitting formulas for the best compression methods to relate their computational complexity, approximation accuracy and compression ratio performance. These have been validated against real datasets and can be used to assess the effectiveness of the selected compression schemes for further hardware architectures. In the last part of the chapter we investigate how these compression schemes perform in terms of energy efficiency, reduced network delay and increased reliability for multi-hop networks in the presence of realistic channel access procedures. Interestingly, we find that linear compression (e.g., LTC) is beneficial in all cases but more energy-hungry DCT methods often perform worse than no compression in terms of energy expenditure. The use of the latter is thus discouraged and should be carefully evaluated depending on the specific scenario at hand. Also, when packet losses affect the data delivery, as due to, e.g., packet collisions, correctly configured compression schemes also outperform the uncompressed transmission case in terms of reconstruction fidelity at the receiver. Essentially, this is due to the corresponding reduction in the transmitted data traffic and, in turn, in the packet collision probability.

5

Optimal Compression Policies for Energy Harvesting Wireless Sensor Networks

Energy constrained system such as WSNs can increase their lifetime by extracting energy from the environment. This is done by means of energy harvesting devices, e.g., solar panels, wind turbines or scavenging antennas. This new scenario where sensor nodes are powered by renewable sources, known as Energy Harvesting Wireless Sensor Networks (EHWSNs), opened a new line of research in the scientific community. In this chapter we consider the problem of optimizing the compression policy of an energy harvesting node equipped with a photovoltaic panel.

5.1 Introduction

In recent years, Energy management in Wireless Sensor Networks (WSNs) has attracted a large attention in the networking research community. Previous work has covered efficient designs in terms of channel access [91], data compression [92] and compressive data gathering and aggregation [84, 93]. However, the common objective that has been considered so far has been that of maximizing the network lifetime through an efficient use of the available energy, considering non-replenishable batteries. Notably, recent advances in energy harvesting systems have opened up new opportunities in terms of energetically self-sustainable (or "perpetual") designs. Note that this calls for new objectives, as optimal policies no longer have to be energy frugal through the entire lifetime of the nodes, but should intelligently assess when energy is available, delivering high performance when the energy income is abundant and putting the nodes into some energy saving mode when it is scarce [30].

Lately, this new line of research has attracted the attention of many researchers. Most approaches dealt with the design of energy neutral transmission policies [31–34,94]. The common assumptions that are made in these papers concern some knowledge (either deterministic or in terms of statistics) of the energy and data arrival processes. This is then exploited to devise optimal transmission schedules (and power) so as to maximize the throughput or minimize the transmission completion time. In [31] a single sensor node transmitting over a fading channel affected by additive Gaussian noise is considered. Energy arrivals and channel states are modeled as Poisson processes and are assumed to be known causally. Optimal transmission policies are evaluated offline for static as well as fading channels. Online policies are also addressed through a dynamic programming formulation. [94] and [32] extend and generalize the results of [31] by relaxing the assumptions on packet arrivals and infinite size of the battery respectively. The authors of [34] present structural results on the optimal transmission policies in the presence of a Markovian energy input process, also assessing the performance gap with respect to selected heuristics. Other approaches dealt with the optimization of the transmission policies in multi user scenarios [33], proposing an online adaptive policy that maximizes the long-term transmission rate (while satisfying energy and power constraints).

Very little work has been done on the optimization of compression techniques in Energy Harvesting Wireless Sensor Networks (EHWSNs). The gain that data compression can offer in energy constrained WSNs scenarios has been studied in [92], where the tradeoffs between compression and transmission energy have been investigated for several lossy compression methods. There, the authors have shown that depending on the specific hardware architecture, compression may lead to substantial gains in terms of energy efficiency. However, in an EHWSN the dynamic of the energy buffer provides additional challenges for the optimization of compression policies.

To the best of our knowledge, the only papers that consider the allocation of energy for compression are [95, 96]. In [95], the authors investigate a scenario where the energy cost of running the source acquisition system (sensing, sampling and compression) is comparable with that associated with the radio transmission activity. They design a system where, based on the statistics of the energy harvesting process, the measured data quality, the channel SNR, and the data queue state, an energy management unit performs energy allocation between source acquisition and data transmission. The objective is that of optimally balancing competing requirements such as signal reconstruction fidelity, queue stability and delay. For a single node scenario, they propose a class of distortion optimal and energy neutral resource allocation policies that keep the data queue stable by meeting an average distortion constraint. The authors of [96] extend [95] to a multi-hop WSN scenario. Specifically, they use the Lyapunov-optimization framework with penalty functions to concoct distributed algorithms that meet quality of service requirements such as queue stability and distortion minimization.

In this work, we address an optimization problem similar to that of [95]. Our approach here is however much more practical as instead of using information theoretic bounds (as also done in [96]) for the energy consumption and the rate-distortion region, we consider realistic rate-distortion curves and energy consumption figures for *lossy* compression methods. Our aim is to obtain the compression/transmission policy that jointly maximizes throughput and reconstruction fidelity at the data collection point, while meeting some predefined energy constraints, i.e., the battery charge level should never go below a predefined guard threshold. In addition, we study the impact of perfect vs delayed Channel State Information (CSI) at the transmitter and of power control over a multipath channel. The contributions of our work are summarized next:

- we model, through a Constrained Markov Decision Problem (CMDP), the transmission and energy dynamics of a sensor node implementing practical lossy compression methods;

- we present an algorithm to numerically evaluate optimal compression/transmission policies, using a Lagrangian relaxation approach combined with a dichotomic search for the optimal Lagrangian;

- we derive theoretical results on the structure of the optimal policy, demonstrating that under suitable but realistic assumptions it is non decreasing in each of the system state components;

- we present a thorough numerical evaluation of optimal as well as heuristic policies for different scenarios, gauging the impact of perfect vs delayed CSI and power control.

The reminder of this chapter is organized as follows. In Section 5.2 we give a high-level view of the problem under analysis, identifying the various system blocks. In Section 5.3 we

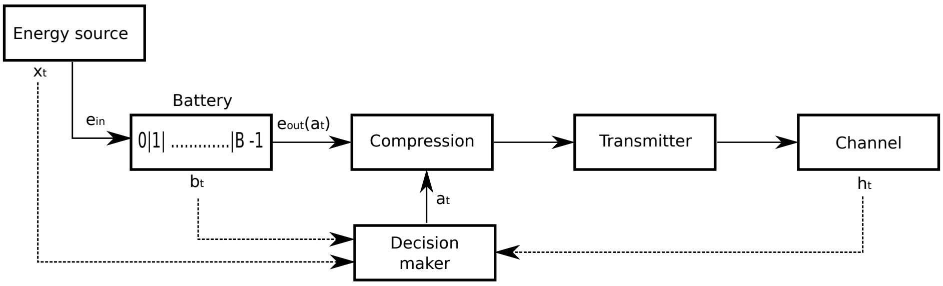


Figure 5.1. *Sensor Node diagram.*

detail our CMDP formulation, present the algorithm to numerically find the optimal policies and discuss the theorems unveiling their structure. In Section 5.5 we discuss a model for a power controlled Rayleigh faded channel. In Section 5.6 we validate the main theorems, and analyze various performance tradeoffs, the role of CSI and power control. In Section 5.7 we report our concluding remarks.

5.2 System Model

We consider a power controlled transmission scenario where a sensor node communicates wirelessly with a data collector (the sink). The sender is powered by a rechargeable battery (energy buffer) that stores the energy harvested from a renewable source, such as a solar panel. The task of the sensor is to sense some physical signal every $T_{\rm sens}$ seconds and report its measurements to the sink. We assume that this task is delay tolerant, i.e., the sensor can store a certain number of measurements $N_{\rm m}$ in its memory buffer and send one or more packets of data after a time $T_{\rm rep} = N_{\rm m} T_{\rm sens}$. Since most physical signals exhibit strong temporal correlation, the data in the buffer can be compressed by means of a *lossy* compression method so as to reduce the amount of data to be sent. Note that lossy compression permits to trade some accuracy in the data reconstruction at the sink for energy savings at the transmitter. Power control is instead exploited at the sender to keep the probability of successful reception around a predetermined target value, as detailed in Section 5.5. For analytical tractability, we assume that the sensor operates in slotted time $t = 0, 1, 2, \dots$.

For the following analysis we consider the diagram of Fig. 5.1. Specifically:

- **Energy Source**: this block models the energy source. The output of this block is $e_{\rm in}$, i.e., the number of energy quanta that are harvested in a time slot. The source dynamics are tracked through a two-state Markov model, where x_t is the source state in the current slot t. In detail, $x_t = 0$ represents a low energy state (e.g., the "night") and $x_t = 1$ represents a high energy state (day). More details on the source model are given in Section 5.3.

- **Battery**: the energy harvested by the Energy Source block is stored in the Battery, i.e., an energy buffer. The state of the buffer is indicated as b_t. e_{out} energy quanta are drawn from the battery in each time slot. Note that e_{out} depends on the compression ratio adopted by the sensor in the current time slot and on the channel state (due to power control).

- **Compression**: this block describes the compression mechanism, linking the energy consumption, the size of the output packet and the representation accuracy of the received data to the action taken by the sensor.

- **Transmitter**: this block accounts for the energy consumption associated with the transmission, which depends on the size of the packet to be transmitted (and hence on the compression ratio) and on the radio transceiver transmission power. The latter, in turn, depends on the channel state as detailed in Section 5.5.

- **Channel**: the channel block is used to model the channel variations in time. As described in Section 5.5, the channel evolution is tracked through a suitable Markov chain. The channel state is indicated with h_t.

- **Decision Maker**: this block takes as input the three components of the system state, namely b_t, x_t and h_t and makes a decision about the current compression ratio a_t, according to a certain quality criterion, see Section 5.3.

5.3 CMDP Representation

In this section we formulate our constrained joint data transmission and compression problem as a Constrained Markov Decision Process (CMDP). A Lagrangian relaxation approach is utilized to deal with the constraint on the energy buffer, which is taken into account to assure that the node remains energetically self-sufficient, while delivering the best performance in terms of reconstruction fidelity at the data collector.

5.3.1 Constrained Markov Decision Process Formulation

We assume that the transmission system operates in slotted time, where $t = 0, 1, 2, \ldots$ is the time index. $\mathcal{S}$ is the state space and the system evolves slot by slot according to the behavior of the energy source, that of the channel and the action taken by the node, i.e., the adopted transmission/compression policy. Our problem is modeled as a CMDP, defined through the tuple $(\mathcal{S}, \mathcal{A}, P, r(\cdot), c(\cdot))$, where $\mathcal{S}$ represents the system state, $\mathcal{A}$ is the action state, P are the transition probabilities (representing the system dynamics), $r(\cdot)$ and $c(\cdot)$ respectively represent a reward and a cost function (which depend on the system state and on the action taken in the current time slot). Specifically, let $s \in \mathcal{S}$ be the system state at time t, with $s_t = s$. We refer to $a_t \in \mathcal{A}_s$ as the action taken by the decision maker at time t, where $\mathcal{A}_s$ is a finite set. The system evolution is Markovian with transition probabilities:

$$p(j|i, a) = P[s_{t+1} = j | s_t = i, a_t = a] \ , \ i, j \in \mathcal{S} \ . \tag{5.1}$$

A decision rule is a function $d_t : \mathcal{S} \to A$ that specifies the action to select when the system is in state s at decision epoch t. In our formulation, actions correspond to the degree of compression, as we further detail below. A policy $\pi = (d_1, d_2, \ldots)$ is a sequence of decision rules. Let Φ_D denote the set of all pure stationary policies $\pi = (d, d, \ldots) = d^\infty$ where d is a deterministic function of the current state s and does not depend on time t. The (finite) reward $r(s_t, a_t) \geqslant 0$ is the instantaneous payoff of taking action a_t in state s_t. As we detail shortly, the notion of reward is used to quantify the reconstruction fidelity of the compressed signal. For any admissible policy $\pi \in \Phi_D$, the expected average reward is:

$$v^\pi(s) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_s^\pi \left[r(s_t, a_t) \right] \ . \tag{5.2}$$

Our goal is to compute the optimal policy $\pi^\star$ that maximize the expected average reward (5.2)

$$v_s^{\pi^\star} = \max_{\pi \in \Phi_D} v_s^\pi \tag{5.3}$$

subject to the global constraint

$$C_s^{\pi^\star} = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_s^{\pi^\star} \left[c(s_t, a_t) \right] \leqslant C_{\max} \tag{5.4}$$

Here, $c(s_t, a_t) \geqslant 0$ is the (known) finite cost incurred in slot t, which is related to the energy buffer state, whereas $C_{\max} \geqslant 0$ is a user defined parameter. Any policy $\pi^\star$ that maximizes

v_s^π will be termed optimal. The constraint $C_{\max}$ is termed *feasible* when there exists at least one policy $\pi \in \Phi_D$ that meets (5.4). The reward of the policy $\pi^\star$ that is optimal subject to constraint (5.4) will be denoted by $v^\star(C_{\max})$. Next, we specify each element of the tuple $(\mathcal{S}, \mathcal{A}, P, r(\cdot), c(\cdot))$ for the problem under analysis.

5.3.1.1 States

Each state $s \in \mathcal{S}$ is composed of three components $s = [b, x, h]$, where $\mathcal{S} = \mathcal{B} \times \mathcal{X} \times \mathcal{H}$. $\mathcal{B}$ denotes the energy buffer (battery) state space, $\mathcal{X}$ denotes the state space of the energy income process, and $\mathcal{H}$ denotes the channel state space. $b \in \mathcal{B}$ is the energy buffer state, $\mathcal{B} = \{0, 1, ..., B\}$ contains all possible energy buffer levels, where the available energy has been discretized into $B + 1$ energy quanta. $x \in \mathcal{X} = \{0, 1\}$ is the energy source state (assumed observable). In detail, a certain amount of energy can be harvested in each time slot t. When $x = 0$, the energy source is in a "bad" state, where the amount of energy quanta harvested from the source in slot t is $e_{\text{in}} = 0$ w.p. 1. On the other hand, when $x = 1$, a number $e_{\text{in}} \in \mathcal{E}^{(1)} = \{1, \ldots, E\}$ of energy quanta is harvested in the current slot t, where e_{in} is distributed according to some mass distribution function (mdf). Hence, the evolution of the energy source state is represented through a two state Markov chain, where E represents the maximum number of energy quanta that can enter the energy buffer in a time slot.

The channel state $h \in \mathcal{H} = \{1, \ldots, H\}$ affects the reception probability at the receiver and, since we consider a power controlled transmission system, it also affects the power consumption at the transmitter. We assume h to be independent of x and b and we model it through a further Markov chain. The procedure used to derive the channel state transition matrix for a Rayleigh channel model is detailed in Section 5.5. Optimal policies will be computed in the case where the transmitter has perfect Channel State Information (CSI), in the case where the CSI at the transmitter is obtained with a certain delay (delayed CSI) and when the transmitter has no CSI at all.

5.3.1.2 Actions

At each decision epoch t the transmitter (decision maker) observes the system state $s = s_t$ and chooses an action $a = a_t$ from the action set $\mathcal{A}_s$. In our model, a_t corresponds to the compression ratio η that has to be used by the transmitter in the current slot t. Here,

η is defined as the ratio of the size of the compressed signal with respect to that of its uncompressed version, see [92]. For the action set, we have $\mathcal{A}_s \subseteq \{0, 1, \dots, n\}$. In particular, when $a_t = 0$ the transmitter will be silent in the current transmission slot t. For $0 < a_t < n$, the transmitter will compress the data stored in the memory buffer with a compression ratio $\eta = a_t/n$ and it will thus send the compressed data to the collector. When $a_t = n$, the transmitter will send the data without performing any compression ($\eta = 1$).

5.3.1.3 Transition Probabilities

Let t and $s_t = [b_t, x_t, h_t]$ respectively be the current time index and the system state in slot t. With a_t we indicate the action taken in the current slot t and we refer to $e_{\text{out}}(s_t, a_t)$ as the number of energy quanta that are taken by the buffer (i.e., the energy consumption) given that action a_t is chosen. Hence, the energy buffer evolves as:

$$b_{t+1} = \max\{0, \min\{b_t + e_{\text{in}} - e_{\text{out}}(s_t, a_t), B\}\} = [b_t + e_{\text{in}} - e_{\text{out}}(s_t, a_t)]^{\dagger} \ . \tag{5.5}$$

If s_t is the system state, action a_t is admissible only if $e_{\text{out}}(s_t, a_t) \leqslant b_t$. Further, $e_{\text{out}}(s_t, a_t)$ is given by the sum of two components: the energy consumption associated with the compression task $e_{\text{c}}(a_t)$ and that associated with the transmission task $e_{\text{tx}}(a_t, h_t)$. In the present analysis, as a compression technique we select the Lightweight Temporal Compression (LTC) algorithm, since it strikes a good balance in terms of compression vs energy consumption. As shown in [92], for LTC the processing energy $e_{\text{c}}(a_t)$ is related to the compression ratio $\eta = a_t/n$ through the linear relationship:

$$e_{\text{c}}(a_t) = \begin{cases} 0 & \eta \in \{0, 1\} \\ \left(\alpha \dfrac{a_t}{n} + \beta\right) N_b E_0 & 0 < \eta < 1 \ . \end{cases} \tag{5.6}$$

Note that when $\eta \in \{0, 1\}$ the node does not perform any compression, thus we have $e_{\text{c}}(0) = 0$. In (5.6), N_b is the number of bits that are to be compressed, E_0 is the energy consumption of the micro controller in one clock cycle, α and β are two fitting coefficients, see [92]. We remark that the results that we discuss in this chapter are rather general and can be readily extended to other compression approaches, i.e., by just replacing (5.6) with the appropriate function.

The energy consumption $e_{\text{tx}}(a_t, h_t)$ depends on the number of bits N_b that the node has to transmit, on the action a_t and on the channel state h_t:

$$e_{\rm tx}(a_t, h_t) = \frac{a_t}{n} N_b E_{\rm tx}(h_t) \ , \tag{5.7}$$

where $E_{\rm tx}(h_t)$ is the energy consumption associated with the transmission of one bit and depends on the specific radio technology and on the channel state h_t (due to the power control mechanism).

The transition probability from state $s_t = [b_t, x_t, h_t]$ to state $s_{t+1} = [b_{t+1}, x_{t+1}, h_{t+1}]$ given that action a_t is selected is:

$$p(s_{t+1}|s_t, a_t) = \delta\left(b_{t+1} - [b_t + e_{\rm in} - e_{\rm out}(s_t, a_t)]^\dagger\right) \cdot p_{e_{\rm in}}(e_{\rm in}|x_t) \cdot p_x(x_{t+1}|x_t) \cdot p_h(h_{t+1}|h_t) \ , \tag{5.8}$$

where $\delta(\cdot)$ is the indicator function (equal to 1 if the argument is zero and null otherwise), $p_{e_{\rm in}}(e_{\rm in}|x_t)$ is the mdf of the input energy in state x_t, while $p_x(x_{t+1}|x_t)$ and $p_h(h_{t+1}|h_t)$ are respectively obtained from the transition probability matrices of the energy source and the channel.

5.3.1.4 Reward Function

In order to maximize the reconstruction fidelity at the data collector, the reward function $r(a_t)$ is chosen to be a strictly increasing function of the selected compression ratio $\eta = a_t/n$. The reward is zero if the selected action is $a_t = 0$ and reaches one when $a_t = n$. We define $r(a_t)$ as:

$$r(a_t) = \begin{cases} 0 & a_t = 0 \\ 1 - \left(\dfrac{p_1(a_t/n)^2 + p_2 a_t/n + p_3}{a_t/n + q_1}\right)\sigma^2_{\rm noise} & a_t \in \{1, \dots, n\} \end{cases} \tag{5.9}$$

where the polynomial function in (5.9) is the LTC rate distortion curve obtained in [92], which relates the error in the reconstruction process to the compression ratio η. p_1, p_2, p_3 and q_1 are suitable constants, whereas $\sigma^2_{\rm noise}$ is the variance of the white noise that is superimposed to the sensed signal. Again, a different compression scheme requires to know the associated rate distortion function, which has to be plugged into (5.9).

5.3.1.5 Cost Function

Since we want to prevent the energy buffer from depletion, we impose a positive cost for those states where the battery level is below a pre-determined threshold $b_{\rm th}$. Also, the cost increases linearly when b decreases below $b_{\rm th}$. Hence, at time t, the cost is defined as:

$$c(b_t) = \begin{cases} 0 & b_t \geqslant b_{\text{th}} \\ \dfrac{b_{\text{th}} - b}{b_{\text{th}}} & b_t < b_{\text{th}} \ . \end{cases} \tag{5.10}$$

Note that in our case $r(s_t, a_t)$ only depends on the action a_t and the cost $c(s_t, a_t)$ only depends on the state component b_t, see Fig. 5.2. These facts will be used in Section 5.4.2 to characterize the structure of the optimal policies.

5.4 Optimal Policies

The unconstrained problem (5.3) is solved by first writing the Bellman's optimality equation:

$$v(s) = \max_{a \in \mathcal{A}_s} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p\left(s'|s,a\right) v(s') \right\} \ , \tag{5.11}$$

where the corresponding optimal policy is given by

$$a^\star(s) = \operatorname*{argmax}_{a \in \mathcal{A}_s} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p\left(s'|s,a\right) v(s') \right\} \ . \tag{5.12}$$

The optimal average reward $v(s)$ can be obtained through, e.g., the Value Iteration (VI) algorithm [97], whereas $a^\star(s)$ indicates the mapping from s to the optimal action, i.e., $d : \mathcal{S} \rightarrow A$. When the constraint (5.4) is added to the problem, a Lagrangian relaxation approach can be used to convert the constrained formulation (see (5.3) and (5.4)) into an equivalent unconstrained MDP. This is achieved by defining a new reward function, $r(s, a; \beta)$, through the Lagrangian multiplier $\beta > 0$:

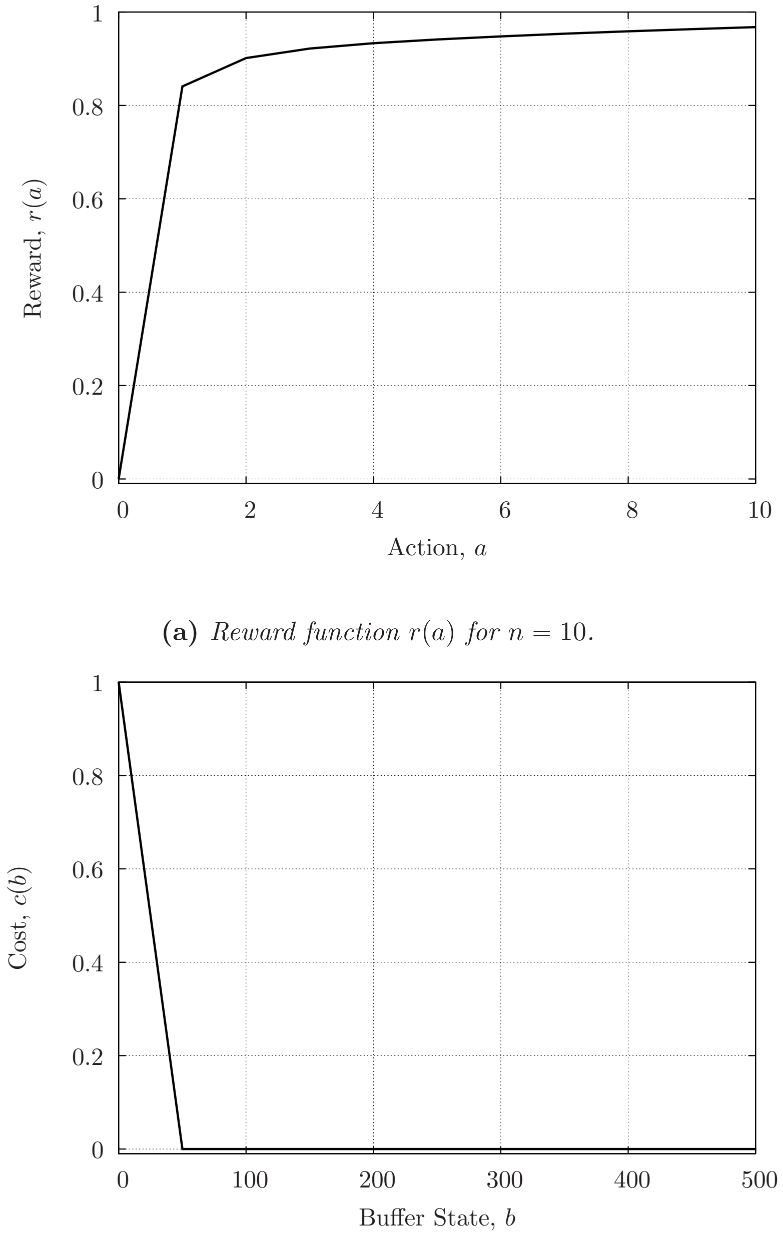
$$r(s,a;\beta) = r(s,a) - \beta c(s,a) \ . \tag{5.13}$$

The new optimality equation is thus:

$$v_\beta(s) = \max_{a \in \mathcal{A}_s} \left\{ r(s,a;\beta) + \sum_{s' \in \mathcal{S}} p\left(s'|s,a\right) v_\beta(s') \right\} \ , \tag{5.14}$$

that for any fixed value of β can be solved via VI. Theorem 12.7 of [98] proves the existence of an optimal policy and also the fact that the Lagrangian formulation (5.14) can be solved through a search on the parameter β as we now explain.

In detail, we need to find the optimal Lagrangian multiplier $\beta^\star$ for which the constraint (5.4) is satisfied with equality. To this end, we adapt the technique of [99], [100], where a



(a) *Reward function $r(a)$ for $n = 10$.*

(b) *Cost function $c(b)$ for $b_{\mathrm{th}} = 50$ and $B = 500$.*

Figure 5.2. *Example of reward and cost functions.*

Q-learning algorithm is applied in order to determine the proper Lagrangian multiplier for a feasible $C_{\max}$. This algorithm finds the optimal policy π^* which satisfies (5.4) with equality. Specifically, π^* is a mixture of two pure and stationary policies [98], referred to here as π^+ and π^-. In order to obtain these policies we solve the constrained problem for two values of β, namely $\beta^+ = \beta^\star + \delta\beta$ and $\beta^- = \beta^\star - \delta\beta$, for policies π^+ and π^-, respectively. We denote the costs of these two policies with $C^+ \geqslant C_{\max}$ and $C^- \leqslant C_{\max}$, where we have:

$$C_{\max} = qC^+ + (1-q)C^- \; . \tag{5.15}$$

Solving this equation, we obtain q, i.e., the weight parameter of the mixture. Hence, the optimal policy $\pi^\star$ in each state $s \in \mathcal{S}$ is given by:

$$\pi^\star(s) = \begin{cases} \pi^+(s) & \text{w.p.} \quad q \\ \pi^-(s) & \text{w.p.} \quad 1-q \; . \end{cases} \tag{5.16}$$

Note that when $C_{\max} = 0$ the probability that the buffer is emptied below b_{th} is zero (buffer outage), whereas an increasing $C_{\max}$ corresponds to higher buffer outage probabilities.

5.4.1 Numerical Solver

Next, we outline the algorithms that we have used to solve the CMDP. Alg. 1 is a standard VI that is used to solve the Lagrangian (unconstrained) problem, for a given value of β. Here, we use (5.14) as an update rule to iteratively compute the optimal average reward $v_\beta(s)$. The converge $v_\beta^n(s) \to v_\beta(s)$ is assured by the standard results in, e.g., [97], whereas $\text{sp}\,(\cdot)$ is the span seminorm operator [101]. Alg. 2 implements a dichotomic search over the β parameter, as an outer loop. Note that this search strategy is effective because, as proven, e.g., in Lemmas 3.1 and 3.2 of [102], the optimal Lagrangian reward $v_\beta(s)$ is a uniformly absolutely continuous, monotone and non-increasing function of β. Previous algorithms implemented similar searches by starting with a small (large) initial value of β for which the cost constraint is not (is) satisfied and increase (decrease) it using suitable step sizes (that usually get smaller as the algorithm gets closer to the optimal solution, see, e.g., Eq. (3.9) of [99]). For each β, $v_\beta(s)$ is found through the VI of Alg. 1 (as we do here), and the associated optimal policy π_β is obtained. The algorithm stops when the average cost gets sufficiently close to the chosen constraint $C_{\max}$, see also Alg. 2 of [100]. The design of the (decreasing) step size is however critical. Our algorithm is instead more robust as it does not require any heuristic for the

```
Value Iteration Algorithm
select $v_\beta^0 \in V$, $\epsilon > 0$ and set $n = 0$ ;
repeat
    $v_\beta^{n+1}(s) = \max_{a \in \mathcal{A}_s} \left\{ r(s, a; \beta) + \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\beta^n(j) \right\}$ ;
    $n = n + 1$ ;
until sp $\left( v_\beta^{n+1} - v_\beta^n \right) < \epsilon$;
$a_\beta^\star(s) = \mathrm{argmax}_{a \in \mathcal{A}_s} \left\{ r(s, a; \beta) + \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\beta^n(j) \right\}$ ;
```

Algorithm 1: Value Iteration for a fixed value of β

adaptation of the step size, which is automatically updated by the dichotomic adaptation rule $\beta^{n+1} \leftarrow f(\beta^n)$, where $f(\cdot)$ is a suitable function (see Alg. 2). Also, provided that $\beta_{\max}$ is selected so that the average cost of the optimal policy $\pi^\star_{\beta_{\max}}$ is larger than $C_{\max}$, Alg. 2 is guaranteed to converge and this descends from the monotonic behavior of $v_\beta(s)$. Thus, it will always find a pair (β^+, β^-) that verifies the above properties.

5.4.2 Structure of Optimal Policies

In this section, we present some results on the structure of the optimal policies arising from the optimization technique of Section 5.4. For improved clarity, the proof of the theorems is given in the Appendix. Moreover, the properties that we discuss here are further elaborated in Section 5.6 where we present some numerical results. Before delving into the description of the main theorems, in the following we introduce some useful definitions.

Definition 5.4.1 (Supermodularity)**.** A function $f : X \times A \times B \to \mathbb{R}$ is supermodular in (x, a) for a fixed parameter β, if for all $x' \geqslant x$ and $a' \geqslant a$

$$f(x', a'; \beta) - f(x', a; \beta) \geqslant f(x, a'; \beta) - f(x, a; \beta) \ . \tag{5.17}$$

Definition 5.4.2 (First-order Stochastic Dominance)**.** Let X_1 and X_2 be random variable with the same support $\mathcal{X}$. X_1 first-order dominates X_2, or $X_1 \succeq X_2$ if

$$F_{X_1}(x) \leqslant F_{X_2}(x) \tag{5.18}$$

for all $x \in \mathcal{X}$, where F_{X_i} is the cumulative distribution function (cdf) of X_i.

```
Lagrangian multiplier update
set $n = 0$, $\beta_{\min} = 0$, $\beta_{\max}$ , $\beta^0 = \beta_{\min}$ ;
repeat
    compute $\pi^\star_\beta(s)$ via Algorithm 1 ;
    compute the stationary distribution $\rho(s)$ induced by $\pi^\star_\beta(s)$ ;
    if $\sum_{s\in\mathcal{S}} \rho(s)c(s, a^\star_\beta(s)) > 0$ then
        $\beta^{n+1} = \frac{\beta^n + \beta_{\max}}{2}$;
        $\beta_{\min} = \beta^n$ ;
    else
        $\beta^{n+1} = \frac{\beta^n + \beta_{\min}}{2}$;
        $\beta_{\max} = \beta^n$ ;
    end
    $n = n + 1$ ;
until $|\beta^{n+1} - \beta^n| < \epsilon$;
```

Algorithm 2: Dichotomic Algorithm for the Lagrangian Multiplier Update

Definition 5.4.3 (Stochastically Increasing Family)**.** Let $\{X_\theta\}_{\theta\in\mathbb{R}}$ be a family of random variables on the same support $\mathcal{X}$. $\{X_\theta\}$ is stochastically increasing if

$$X_{\theta'} \succeq X_\theta \tag{5.19}$$

whenever $\theta' \geqslant \theta$.

We now present some results on the structure of the optimal policy. Specifically, we prove that provided some properties for the reward function and for the transition probabilities of the channel and the source state, the optimal policy has a threshold structure and is monotonically increasing in each system state component. This means that optimal policies can be efficiently stored in the node memory in the form of a lookup table having a few entries. In each decision epoch then, the node just need to evaluate its state and select the proper action through a simple query.

Theorem 5.4.1. *Let the instantaneous Lagrangian reward function $r(s, a; \beta)$ be supermodular in the pair (b, a), concave and non-decreasing in b, then the optimal policy $\pi^\star_\beta$ is a monotone non-decreasing function of the buffer state b.*

Theorem 5.4.2. *Let the assumptions of Theorem 5.4.1 hold, let $r(s, a; \beta)$ be supermodular in the pair (h, a), and let $p_h(h'|h)$ be stochastically increasing in h, then the optimal policy $\pi^\star_\beta$ is a monotone non-decreasing function of the channel state h.*

Theorem 5.4.3. *Let the assumptions of Theorem 5.4.1 hold, let $r(s, a; \beta)$ be supermodular in the pair (x, a), and let $p_x(x'|x)$ be stochastically increasing in x, then the optimal policy $\pi^\star_\beta$ is a monotone non-decreasing function of the energy source state x.*

5.5 Channel Model

Next, we derive an H-state Markovian model for a power controlled transmission link in the presence of Rayleigh fading. We consider that the transmitter can adapt its transmission power according to H (radio specific) transmission levels $P_{\mathrm{tx}}[i]$, with $i \in \mathcal{H} = \{1, \dots, H\}$. These are assigned to each channel state to assure that the packet error probability P_{pkt} remains smaller than or equal to a given target ζ, i.e., $P_{\mathrm{pkt}} \leqslant \zeta$ for states $i \leqslant H - 1$. This amounts to identifying $H + 1$ thresholds for the fading power and using them to characterize the H channel states, so that power levels $[P_{\mathrm{tx}}[1], P_{\mathrm{tx}}[2], \dots, P_{\mathrm{tx}}[H]]$ with $P_{\mathrm{tx}}[1] < P_{\mathrm{tx}}[2] < \dots < P_{\mathrm{tx}}[H]$ are respectively assigned to states $[1, 2, \dots, H]$. For a certain Signal to Noise Ratio (SNR) γ and compression level a, the packet error rate is evaluated as:

$$P_{\mathrm{pkt}}(\gamma, a) = 1 - (1 - P_{\mathrm{bit}}(\gamma))^{L(a)} \ , \tag{5.20}$$

where $L(a)$ is the packet size expressed in bits and $P_{\mathrm{bit}}(\gamma)$ is the bit error rate for the selected modulation scheme. Note that the packet size depends on the level of compression $a \in \{0, 1, \dots, n\}$ in the current time slot and is obtained as: $L(a) = aL_{\max}/n$, where $L_{\max}$ is the maximum packet size expressed in bits. Assuming a $\pi/4$-DQPSK modulation, the bit error rate is approximated as:

$$P_{\mathrm{bit}}(\gamma) = \frac{4}{3} \mathrm{erfc}\left(\sqrt{\gamma}\right) \ . \tag{5.21}$$

From (5.20) and (5.21), assuming $L(a) = L_{\max}$ (i.e., $a = n$), the SNR threshold γ_{th} corresponding to the target packet error probability ζ is:

$$\gamma_{\mathrm{th}} = \left(\mathrm{erfc}^{-1}\left(\frac{3}{4}\left(1 - (1 - \zeta)^{1/L_{\max}}\right)\right)\right)^2 \ . \tag{5.22}$$

Thus we have that $P_{\mathrm{pkt}}(\gamma, a) \leqslant \zeta$, with $a \in \{1, \dots, n\}$ is equivalent to requiring that $\gamma \geqslant \gamma_{\mathrm{th}}$. For state $i \in \mathcal{H}$, the SNR γ at the receiver is given by $\gamma(t) = \gamma_0[i]\alpha(t)$, where $\gamma_0[i] =$

$P_{\rm rx}^0[i]/(N_0 B)$ is the average SNR, which depends on the path loss and on the noise power $N_0 B$ (N_0 is the power spectral density and B is the transmission bandwidth), whereas $\alpha(t)$ is the fading power. The average received power $P_{\rm rx}^0[i]$ for transmission level i is obtained as:

$$\left[P_{\rm rx}^0[i]\right]_{\rm dB} = [P_{\rm tx}[i]]_{\rm dB} + [G]_{\rm dB} - [A]_{\rm dB} - [\mathcal{PL}]_{\rm dB} \ , \tag{5.23}$$

where $[G]_{\rm dB}$ and $[A]_{\rm dB}$ respectively represent the total antenna gains (transmitter and receiver) and the total attenuation losses (transmitter and receiver), whereas $[\mathcal{PL}]_{\rm dB}$ is the path loss attenuation, expressed in dB, that among other parameters depends on the distance d between the transmitting sensor and the sink see, e.g., Chapter 2 of [103]. The fading thresholds $\alpha[0] \geqslant \alpha[1] \geqslant \cdots \geqslant \alpha[H]$ are evaluated as follows. In state $i \in [1, \dots, H-1]$, the transmission power is $P_{\rm tx}[i]$ and the fading is distributed in $[\alpha[i], \alpha[i-1]]$. From the condition $\gamma \geqslant \gamma_{\rm th}$, for state i we must have that:

$$[\alpha[i]]_{\rm dB} \geqslant [\gamma_{\rm th}]_{\rm dB} - [\gamma_0[i]]_{\rm dB} \ , \tag{5.24}$$

using equality in the previous equation returns the lower edge of the interval $[\alpha[i], \alpha[i-1]]$. Moreover, we have $\alpha[0] = +\infty$ and $\alpha[H] = 0$. Note that H is the only channel state for which the QoS requirement $P_{\rm pkt} \leqslant \zeta$ cannot be met. Once the thresholds have been computed, the transition probabilities are derived as in [104, 105] according to:

$$p_x(j|i) = \frac{\int_{\sqrt{\alpha[i]}}^{\sqrt{\alpha[i-1]}} \int_{\sqrt{\alpha[j]}}^{\sqrt{\alpha[j-1]}} f_{R_1R_2}(r_1, r_2, \rho) dr_1 dr_2}{\theta_i} \tag{5.25}$$

$$f_{R_1R_2}(r_1, r_2, \rho) = \frac{4 r_1 r_2}{\lambda} e^{-(r_1^2 + r_2^2)/\lambda} I_0(2\rho r_1 r_2/\lambda) \tag{5.26}$$

where $f_{R_1R_2}(r_1, r_2, \rho)$ is the bivariate Rayleigh joint pdf of the fading envelope [104], $\lambda = 1 - \rho^2$, $\rho = J_0(2\pi f_d T_p)$ is the correlation of two samples of the corresponding Rayleigh envelope that are spaced by T_p seconds, f_d is the Doppler frequency, T_p is the transmission duration, $J_0(\cdot)$ and $I_0(\cdot)$ are the Bessel function and the modified Bessel function of the fist kind and order zero. θ_i are the steady state probabilities for each channel state that are readily computed from the fading thresholds, using the Rayleigh SNR pdf, see, e.g., [105].

5.6 Results

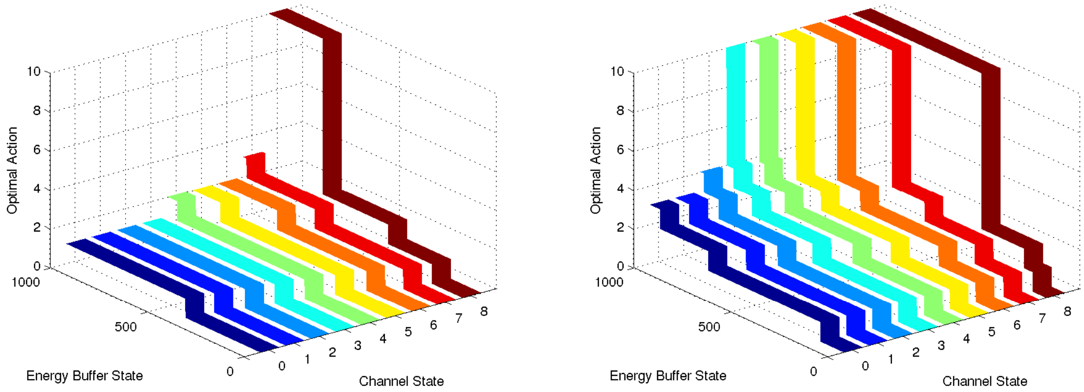
In this section, we discuss the performance of the optimal policies obtained as detailed in Section 5.3. We first look at their structure, validating what predicted by Theorems 5.4.1, 5.4.2 and 5.4.3. After that, we present a thorough performance evaluation for different energy budget scenarios, highlighting the relations among the system parameters and the average reconstruction fidelity at the sink. Moreover, we show how the performance scales when the assumption of perfect CSI is removed, i.e., when the channel state information is retrieved with a certain delay (delayed CSI). Finally, we describe two heuristic policies, comparing them against the optimal solution and investigating the impact of power control. In the results that we discuss in this chapter, we use $C_{\max} = 0$ so that $\pi^\star$ always avoids actions that lead to buffer outage events. Since the cost in our model is non negative and $C_{\max} = 0$, the weight parameter q is always equal to 0 and we always take π^- as the optimal policy.

5.6.1 Optimal Policies: Structural Results

With Theorem 5.4.1 we have shown that the optimal policy is non-decreasing in the energy buffer state b. For this result to hold, we need that the Lagrangian reward $r(s, a; \beta)$ is A1) supermodular in the pair (b, a), A2) concave and non decreasing in b. From (5.13), we have that $r(s, a)$ only depends on a, whereas from (5.10) it descends that the cost function $c(s, a)$ only depends on b. From these facts and $\beta > 0$ it follows that $r(s, a; \beta)$ satisfies A1 and A2.

Theorem 5.4.2 states that the optimal policy is also non-decreasing in the channel state component of the system state, provided that A3) the family of transition probabilities $p_h(h'|h)$ is stochastically increasing in h. This condition means that the probability of going to a state $h' \in \mathcal{H}$ is higher from a state $h_1 \in \mathcal{H}$ that is closer to h' than any other state $h_2 \in \mathcal{H}$ such that $|h_1 - h'| < |h_2 - h'|$, for every choice of h'. With our channel modeling technique of Section 5.5 assumption A3 is verified. Besides that, we also need to verify that A4) $r(s, a; \beta)$ is supermodular in the pair (h, a), but since the reward only depends on s through its buffer component b, condition A4 is met.

In order for Theorem 5.4.3 to hold, A5) $p_x(x'|x)$ has to be stochastically increasing in x, A6) the energy income in states x_i has to take values in disjoint ordered sets and A7) $r(s, a; \beta)$ must be supermodular in the pair (x, a). Here, we model the energy source through a Markov chain with transition probabilities that verify A5. Moreover, we deterministically set the energy income in state $x = 0$ to 0, while the energy income in state $x = 1$ is distributed according to a truncated Gaussian r.v. that takes values in $[1, E]$: hence, also the



(a) *Night* **(b)** *Day*

Figure 5.3. *Structure of the optimal policy.*

assumption A6 is verified. Finally, A7 is also met due to similar arguments as those discussed for Theorem 5.4.1 above.

In Fig. 5.3, we show an example policy for $B = 1000$ energy quanta, where the action space is quantized according to $\mathcal{A}_s = \{0, 1, \dots, n\}$ with $n = 10$ and there are nine channel states. From this plot we see that the results of the theorems hold. In detail, the optimal policy is monotonically non decreasing in the buffer state (i.e., along each stripe in the figures), in the channel state (across different stripes in the same figure) and in the source state (see any two stripes, for the same channel state, in Figs. 5.3a and 5.3b).

5.6.2 Optimal Policies: Performance Evaluation

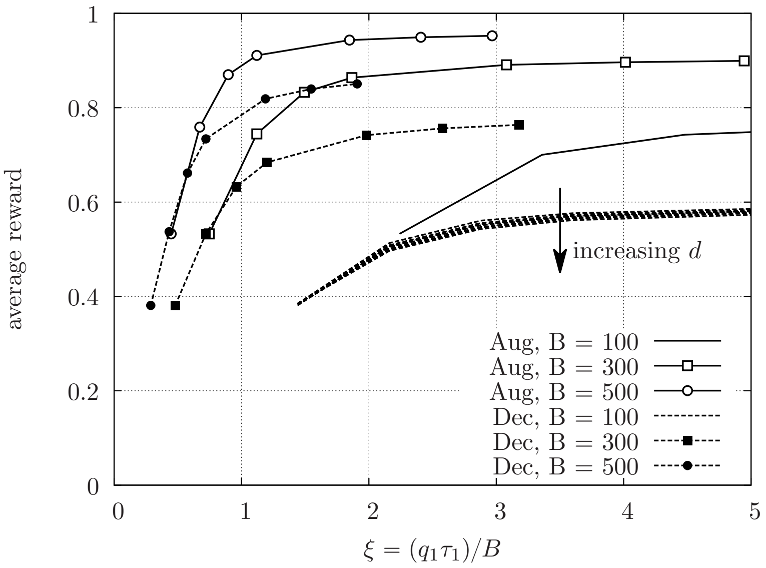
In this section, we discuss the performance of the optimal policies that have been numerically obtained utilizing the algorithms of Section 5.4.1. Different scenarios have been considered, varying the energy buffer size, the average energy income in state $x = 1$, the transition probabilities for the energy source and the distance d between the sensor node and the sink. These settings have been summarized introducing the new variable ξ, which describes the average energy income for each set of parameters. Specifically, ξ is defined as $\xi = q_1 \tau_1 / B$, where q_1 is the average energy income in the "good" state $x = 1$, τ_1 represents the average time the energy source model stays in $x = 1$ before moving to $x = 0$ and B is the energy buffer size. We start by investigating the average reward (reconstruction fidelity) that is earned at the sink by the optimal policy, which is evaluated as:

$$R \overset{\text{def}}{=} \sum_{s \in \mathcal{S}} \rho(s) r(s, a^\star(s)) P_{\text{suc}}(s, a^\star(s)) \ , \tag{5.27}$$

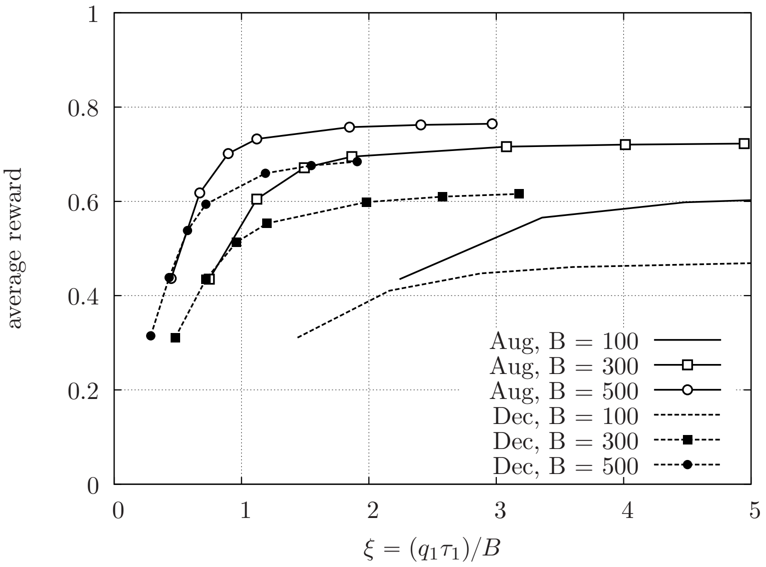
where $\rho(s)$ is the steady state probability distribution induced by the optimal policy $\pi^\star$, $r(s, a^\star(s))$ is the immediate reward accrued by $\pi^\star$ in state $s = [b, x, h]$, see (5.9), and $P_{\text{suc}}(s, a^\star(s)) = E[(1 - P_{\text{pkt}}(\gamma, a^\star(s)))]$ (see (5.20)) is the average probability that a packet compressed according to the optimal action $a^\star(s)$ is successfully received at the sink, when the channel is in state $h \in \mathcal{H}$ (the expectation is taken for the SNR γ in the interval $[\gamma_0[h]\alpha[h], \gamma_0[h]\alpha[h-1]]$, which corresponds to the SNR range associated with state h, see Section 5.5). Note that compressing the packet gives us a further gain in terms of $P_{\text{suc}}(s, a^\star(s))$, as smaller packet sizes result in smaller packet error probabilities. Hence, the average reconstruction fidelity at the sink, evaluated through (5.27), is benefited from a higher degree of compression when the SNR is low. On the other hand, for high SNR values, a higher fidelity is obtained for $a = 1$ as in this case the dominating term in (5.27) is the immediate reward $r(s, a^\star(s))$.

Fig. 5.4 shows the results for the average reward R for different values of ξ. In particular, we considered two different scenarios for the average duration τ_1 of the "good" energy state $x = 1$, picking $\tau_1 = 14$ hours and $\tau_1 = 10$ hours. These parameters are taken from [106], where a Markovian model for the statistical description of the energy harvested by outdoor micro-solar panels for WSN applications is presented. According to the results in this paper, $\tau_1 = 14$ and $\tau_1 = 10$ respectively correspond to the average duration of sunlight in a day for the city of Los Angeles in the months of August and December. For the battery size, we considered three values, i.e., $B \in \{100, 300, 500\}$ energy quanta. Finally, for each pair (τ_1, B), the average energy income in a time slot q_1 is varied in the set $\{8, 12, 16, 20, 33, 43, 53\}$ energy quanta. Also, in Figs. 5.4a and 5.4b we respectively plot R for the two cases of perfect and delayed-CSI at the transmitter.

From Fig. 5.4, we see that for any given pair (τ_1, B), the average reward R increases with ξ. This is due to the higher average energy income q_1 that allows the system to reach the end of a "good" period $x = 1$ (i.e., day) with a higher residual level of energy in the battery which, in turn, permits to additionally transmit a certain amount of data during the "bad" state (night). R is also monotonically increasing on B and τ_1. This is again because, having



(a) *Perfect CSI.*



(b) *Delayed CSI (T = 3 time slots).*

Figure 5.4. *Average reconstruction fidelity at the sink vs ξ (see (5.27) and (5.28)). $\tau_1 \in \{10, 14\}$ hours, $B \in \{100, 300, 500\}$ energy quanta.*

a bigger battery (higher B) or increasing the duration of the "good" state (higher τ_1) both result in a higher energy availability.

In Fig. 5.4a we also plot results for different values of d, i.e., the distance between the transmitting node and the sink. We only show these results for one scenario (December,

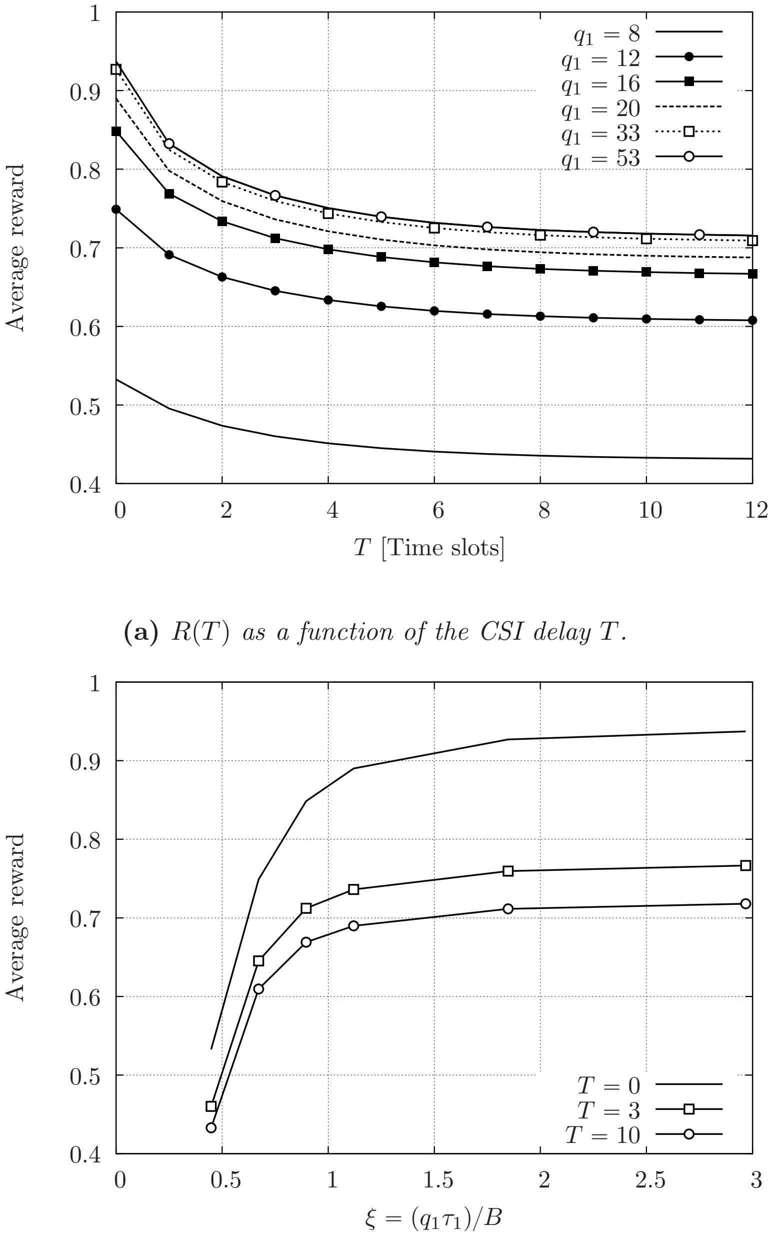
$B = 100$ energy quanta), the others having a similar behavior. d only affects the channel transition probabilities $p_h(h'|h)$, with higher values of d leading to worse performance, since on average the channel is more likely to be in a SNR region where the required energy consumption (for the given quality parameter ζ) is higher.

In Fig. 5.4b we show the average reconstruction fidelity at the sink, $R(T)$, when the CSI at the transmitter is known subject to a certain delay $T \geqslant 1$ (expressed in time slots):

$$R(T) \overset{\text{def}}{=} \sum_{s \in \mathcal{S}} \rho(s) r(s, a^\star) \left[\sum_{h' \in \mathcal{H}} p_h^T(h'|h) P_{\text{suc}}(s', a^\star(s)) \right] , \tag{5.28}$$

where $s = [b, x, h]$, $s' = [b, x, h']$ and $p_h^T(h'|h)$ corresponds to the probability that the channel is in state h' in slot $t + T$, given that the channel state is h in slot t, for any given $t \geqslant 0$. If $\mathbf{H}$ is the channel transition probability matrix, see Section 5.5, $p_h^T(h'|h)$ is the the entry in position (h, h') of the T-step channel transition probability matrix $\mathbf{H}^T$. This probability is used to track a delayed representation of the channel behavior, where T is the delay in the acquisition of the CSI. Note that for $T = 0$, $\mathbf{H}^0$ is the identity matrix and (5.28) reduces to (5.27). As expected, from Figs. 5.4a and 5.4b we see that adding some uncertainty on the channel state at the transmitter results in an overall reduction of the system performance. This is especially detrimental when the transmitter thinks that the channel in the current slot is good (high SNR) and, in turn, sends its packet uncompressed and using a small power level. In fact, if the actual channel state is instead rather bad (low SNR), this behavior is exactly the opposite of what the transmitter should do. As a result, the packet is lost with high probability and this corresponds to a waste of energy and also to a loss of reward, as no signal is recovered at the sink for this time slot.

In Fig. 5.5a we show the average reward $R(T)$ as a function of T in the August scenario, with $B = 500$ and varying q_1. We observe that as the value of T increases, the performance decreases until reaching a minimum value (around $T \simeq 10$ time slots) and this occurs for all the considered values of q_1. This is because $\mathbf{H}^T$ converges towards the stationary distribution of the channel for an increasing T. Fig. 5.5b shows the average reward $R(T)$ in the same scenario for different values of T. Interestingly, the performance gap $R(0) - R(T)$, with $T \geqslant 1$, is an increasing function of ξ. This is because, as discussed above, an increasing T corresponds to a higher transmission error probability. Moreover, for higher value of ξ, the optimal action in most of the system states corresponds to transmitting the data packets uncompressed and transmission errors when ξ is higher have a higher impact on the performance. In this case, in fact, the lost packet would have implied a higher reward, as it is uncompressed and thus its contribution to the reconstruction fidelity is higher and also the energy wasted due to its erroneous transmission is higher than that of sending it compressed.



(a) *$R(T)$ as a function of the CSI delay T.*

(b) *$R(T)$ as a function of ξ for $T \in \{0, 3, 10\}$.*

Figure 5.5. *Average reconstruction fidelity at the sink $R(T)$: impact of delayed CSI.*

5.6.3 Optimal vs Heuristics Policies

In this section we compare the performance of the optimal policy against that of two heuristic policies:

- Constant Compression Policy (CC): in any given time slot, CC transmits a compressed data packet according to a constant compression level $a_{\mathrm{CC}} \in \{1, \ldots, n\}$ for each state s where $b > b_{\mathrm{th}}$, whereas no transmission is performed otherwise. With this policy the reward and the energy consumption per transmitted packet are constant, and can be tuned according to the selected compression level. For the results that follow we have set $a_{\mathrm{CC}} = 1$, corresponding to the smallest compression ratio $\eta = a_{\mathrm{CC}}/n = 0.1$.

- No Compression Policy (NC): in any given time slot, NC transmits a data packet uncompressed ($a_{\mathrm{NC}} = 0$) if $b > b_{\mathrm{th}}$, whereas no transmission is performed otherwise. Note that an uncompressed packet implies the highest reward, but also the energy consumption associated with the transmission of a full packet is the highest.

In Fig. 5.6, we show the temporal evolution for the first 500 time slots for NN, NC and the optimal policy, along with the corresponding evolution of channel and source states. The first three graphs respectively show the energy buffer state evolution (solid line) and the action taken in each time slot (dots) for the optimal policy, CC and NC. The last graph shows the channel state (solid line) and the energy source state (dashed line). It is interesting to note how the optimal policy manages to keep the battery in a state that is neither fully charged nor below the threshold $b_{\mathrm{th}} = 50$, modulating the action that the system takes in each state. In this way, the buffer constraint is met, the average fidelity is maximized and no input energy is wasted. This is not the case for the other two policies. In particular, CC loses some efficiency as it wastes some harvested energy when the energy buffer is full, while NC drains out the energy buffer too fast and, in turn, forces the node to remain idle more often.

Fig. 5.7 shows the percentage of used transmission slots and the average reward per slot R. We consider the same scenario of Fig. 5.5a, with perfect CSI (i.e., $T = 0$). For all the policies the percentage of used slots increases with ξ, CC achieving the best performance since its energy consumption is always that of $a_{cc} = 1$, which is the least energy demanding action. NC, on the other hand, always uses the most expensive action and thus it more often drains out the energy buffer. Thus, the percentage of used slots for this NC is smaller. The optimal

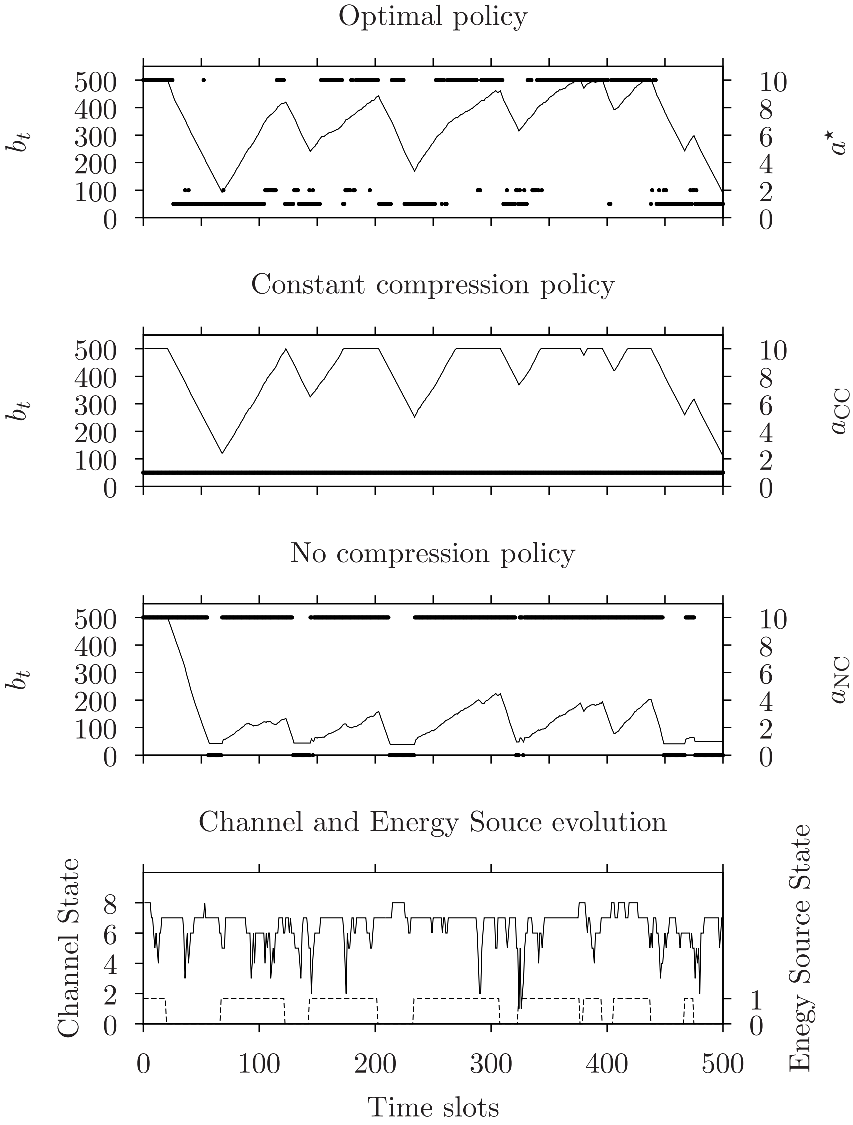
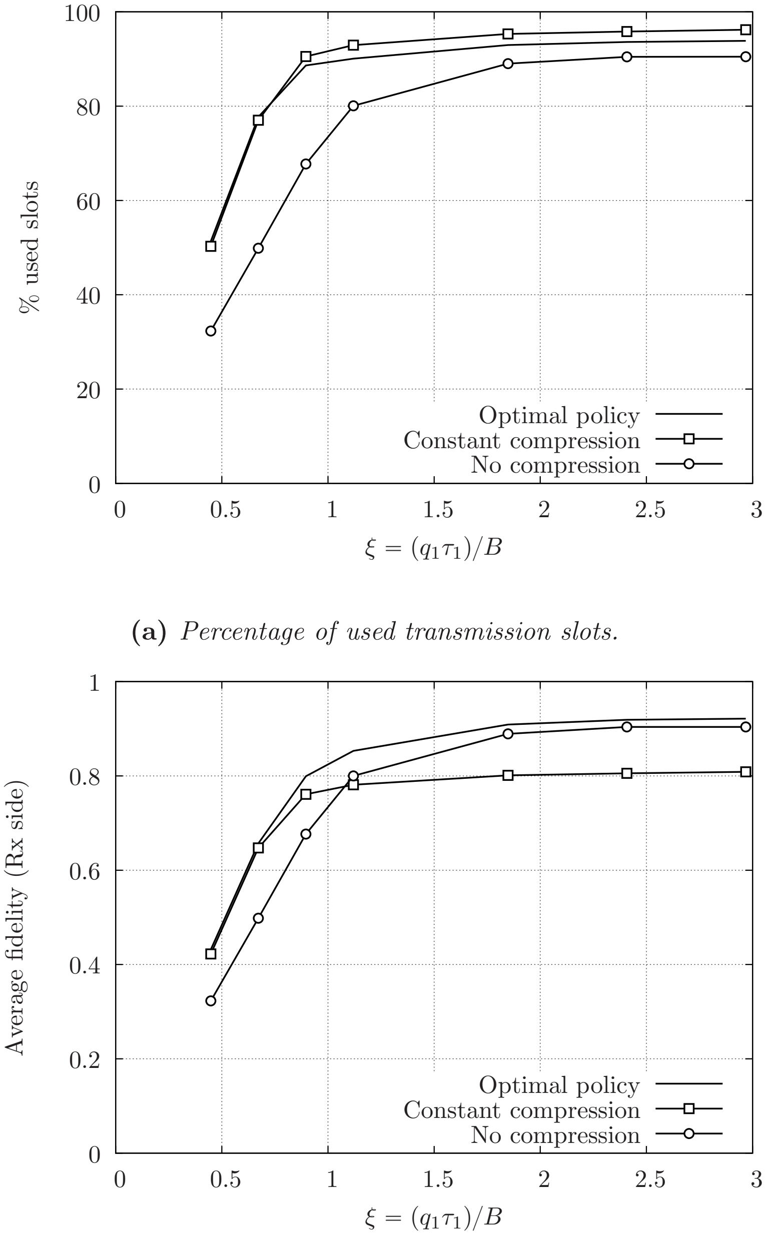


Figure 5.6. *Performance comparison between optimal and heuristic policies: temporal evolution example.*

policy modulates the behavior of the node depending on the system state, the percentage of used slots is in between that of CC and NC, but as Fig. 5.7b shows, its average reward is always higher.

5.6.4 Impact of Power Control

In this section we evaluate the impact of power control at the transmitter. This means that the system state is only composed by the energy buffer state b and the energy source



(a) *Percentage of used transmission slots.*

(b) *Average reconstruction fidelity at the sink R.*

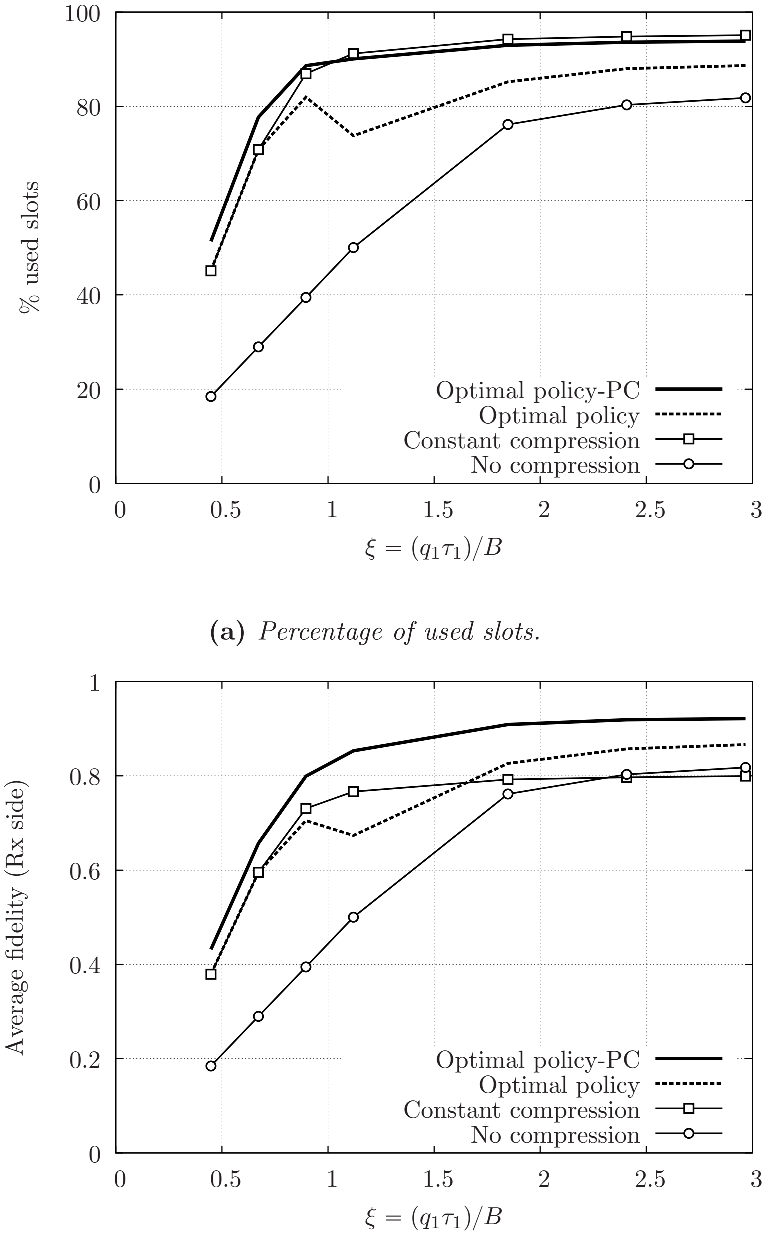
Figure 5.7. *Performance comparison between optimal and heuristic policies.*

state x. Thus, we obtain the optimal policy for this new system state and we investigate its performance in the presence of a Rayleigh faded channel, see Section 5.5, and a fixed transmission power. The probability of a successful transmission is still a function of the channel state, but with a fixed transmission power we can no longer assure that the error probability will be bounded. In fact, setting a low transmission power will lead to a small energy consumption, but the only successful transmissions will occur when the channel state is good (i.e., high SNR). On the other hand, with a high transmission power almost all the transmitted packets will be correctly received, but the energy consumption will be very high and the battery will be drained quickly, forcing the transmitter to stop and wait for incoming energy in order to meet the constraint on the energy buffer level.

Fig. 5.8 shows the performance of the optimal policy ("Optimal policy") along with the performance of CC and NC when power control is not applied. We set the transmission power to the maximum level for this graph. For comparison, we also show the performance of the optimal policy when power control is applied ("Optimal policy-PC"). The performance without power control is lower for all policies, both in terms of transmission activity (% of used slots) and fidelity R. The optimal policy without power control shows a transition around $\xi = 1$. This happens because, in the considered example, when $\xi < 1$ the average amount of energy harvested during a "good" period is no longer sufficient to fully recharge the battery. Thus, the behavior of the optimal policy tends to mimic that of CC, favoring the transmission of many compressed packets with low average reward over the transmission of a smaller number of high rewarding (uncompressed) data packets. On the other hand, when $\xi > 1$, the input energy is more abundant, and the behavior of the optimal policy is similar to NC.

5.7 Conclusions

In this chapter, we have considered the problem of finding optimal transmission/compression policies for an Energy Harvesting sensor node. We have modeled the system through a Constrained Markov Decision Problem, by taking into account the dynamics of the energy harvesting process, as well as arising from channel and energy buffer behaviors. Our model exploits the results obtained in Chapter 4, which characterize the lossy compression methods in terms of rate distortion curve and associated energy consumption. We have investigated the policies arising from the optimization procedure both from a theoretical point of view, verifying that their structure has some monotonicity properties, and through simulation results in different energy budget scenarios. Moreover, the impact of Channel State Information and power control at the transmitter have been investigated. Finally, we have compared the performance of the optimal policy against that of heuristic policies, showing that a policy that



(a) *Percentage of used slots.*

(b) *Average reconstruction fidelity at the sink R.*

Figure 5.8. *Impact of power control.*

modulates the node behavior depending on the system state leads to a more efficient use of the available energy and also exhibits better performance in terms of average representation accuracy.

6 Conclusion

In this thesis we have presented a framework for the performance evaluation of lossy compression methods and the analysis and optimization of compression policies in Energy Harvesting Wireless Sensor Networks (EHWSNs).

In the first two chapters, we have developed two mathematical models, one for space–time correlated signals and one for the representation of renewable energy sources. In particular, the first model allowed us to analyze the statistical properties of real world signals, i.e., their correlation structure, and accurately reproduce them. We have validated this model against real datasets, showing that it is possible to use this tool to finely tune the correlation of synthetic signals, which is particularly useful for simulation purposes.

The model we have developed in Chapter 3 for a renewable energy source, specifically a photovoltaic panel, finds application in energy self–sustainable systems, such as, for instance, in the simulation of energy harvesting communication networks and is as well useful to extend current theoretical work through more realistic energy models. Our approach has been validated against real energy traces, showing good accuracy in their statistical description in terms of first and second order statistics.

In Chapter 4 we have investigated the performance of lossy compression schemes for energy constrained scenarios. This performance comparison has been carried out considering the compression capabilities, the complexity and the energy consumption as performance metrics. In particular, we have investigated whether energy savings are possible depending on signal statistics, compression performance and hardware characteristics. Our study reveals that, depending on the hardware architecture considered, the energy spent for computation may be comparable with the energy spent for the data transmission. In such a scenario, the only methods that can actually lead to energy savings are linear approximation methods, that have the smallest computational cost. In this chapter, we have also derived closed form expression for the best compression methods relating their computational complexity, approximation accuracy and compression ratio. These formulas can be used to investigate whether a compression method leads to energy savings when used with different hardware architectures. Finally, we have extended our analysis to multi-hop scenarios, where we have considered a realistic channel access procedure. We have shown that linear compression is beneficial in all cases and that more energy-hungry methods often perform worse than no compression in terms of energy expenditure and transmission delay.

In the last part of this thesis we have considered the problem of optimizing the compression strategies for an Energy Harvesting sensor node. We modeled through a Constrained Markov Decision Process the dynamics of the transmission and the energy buffer of a sensor node implementing a lossy compression algorithm. We have solved the optimization problem through Lagrangian relaxation and have discussed the structure of the optimal policy arising from such model. In addition, we have shown that the optimal policy can effectively maximize the throughput of the system and the representation accuracy, while fulfilling a predefined energy constraint. Moreover, we have investigated the effect that the delay on the Channel State Information acquisition and the power control have on the performance.

6.1 Future research directions

The work presented in this thesis deals with the optimization of lossy compression policies for EHWSNs; even though we have presented a rather complete framework, including signal and energy source models, the networking scenario can be further generalized. Other energy sources can be investigated (e.g., wind, vibrational or electromagnetic energy), as the statistical behavior of different renewable sources can lead to different results. Moreover, a data buffer model can be included in the optimization problem. In our model, in fact, we have assumed that the sensor node has always some data to transmit, while in a real scenario also the data has a stochastic behavior that can be included in the optimization framework. Finally, the natural extension of the work we presented in the last part of the thesis is to consider the optimization problem from a network perspective, accounting for multiple (possibly cooperating) sensor nodes. This can be done defining a reward function that takes into account the network performance and, similarly to what we did in Chapter 4, considering a realistic channel access procedure. In this way, it would be possible to relate the performance of the optimal policy also to network parameters, such as the number of nodes, the topology or the selected MAC protocol.

A Appendix

In this Section we give the proof of Theorem 5.4.1, 5.4.2 and 5.4.3. In what follows, we omit the dependence on the time index t. For improved readability and with a slight abuse of notation, with a_i, b_i, h_i and x_i we respectively indicate elements of action, battery, channel and energy source sets.

A.1 Proof of theorem 5.4.1

Proof. Since

$$\pi^\star_\beta([b,x,h]) = \underset{a}{\operatorname{argmax}} \left\{ Q_\beta([b,x,h],a) \right\} \ , \tag{A.1}$$

in order to prove that the optimal policy is non-decreasing in the buffer state b, we have to prove that $Q_\beta([b,x,h],a)$ is supermodular in the pair (b,a). Now $Q_\beta([b,x,h],a)$ is defined as:

$$Q_\beta([b,x,h],a) = r(s,a;\beta) + \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\beta(s') \tag{A.2}$$

where the first term in the right hand side of the equation, i.e., $r(s,a;\beta)$, is supermodular by assumption. Let us rewrite the second term as:

$$\begin{aligned} Q_1([b,x,h],a) &= \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\beta(s') && \text{(A.3)} \\ &= \sum_{h' \in \mathcal{H}} \sum_{x' \in \mathcal{X}} \sum_{e_{\text{in}}=0}^{E} p_h(h'|h) p_x(x'|x) p_{e_{\text{in}}}(e_{\text{in}}|x) v_\beta([b + e_{\text{in}} - e_{\text{out}}(h,a), x', h']) \ , && \text{(A.4)} \end{aligned}$$

where we used the independence of the channel evolution and of the source state evolution. We need to prove that the sum in (A.4) is supermodular in (b,a) for any $h \in \mathcal{H}$ and $x \in \mathcal{X}$.

Using Lemma A.1.1 we have that $v_\beta([b,x,h])$ is concave in b, thus it can be shown that $v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a),x',h'])$ is supermodular in (b,a) for any $h'\in\mathcal{H}$ and $x'\in\mathcal{X}$. In fact, since $v_\beta([b,x,h])$ is concave in b, it holds that:

$$v_\beta([b_1,x,h])+v_\beta([b_2,x,h])\leqslant v_\beta([\alpha b_1+(1-\alpha)b_2,x,h])+v_\beta([(1-\alpha)b_1+\alpha b_2,x,h]) \quad \text{(A.5)}$$

for $0\leqslant\alpha\leqslant 1$. Substituting $b_1=b+e_{\text{in}}-e_{\text{out}}(h,a')$, $b_2=b'+e_{\text{in}}-e_{\text{out}}(h,a)$ and $\alpha=(e_{\text{out}}(h,a')-e_{\text{out}}(h,a))/(e_{\text{out}}(h,a')-e_{\text{out}}(h,a)+b'-b)$, and rearranging the terms we get:

$$\begin{aligned}&v_\beta([b'+e_{\text{in}}-e_{\text{out}}(h,a'),x,h])-v_\beta([b'+e_{\text{in}}-e_{\text{out}}(h,a),x,h])\\&\qquad\geqslant v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a'),x,h])-v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a),x,h])\ ,\end{aligned} \quad \text{(A.6)}$$

that for $a'\geqslant a$ and $b'\geqslant b$ proves the supermodularity of v_β in (b,a) for any channel state h and source state x. Furthermore, positive weighted sum of supermodular function is also supermodular, hence $Q_\beta([b,x,h],a)$ is supermodular in (b,a) and the monotonic structure of the optimal policy in the energy state b is proven. □

Lemma A.1.1. *Under the assumptions of Theorem 5.4.1, the expected average reward function $v_\beta(\mathbf{s})$ is concave and monotonically non-decreasing in the energy buffer state b, for any channel state h, and energy source state x.*

Proof. Proceeding by induction, as the V.I. algorithm converges for any $v_\beta^0([b,x,h])$, let us choose $v_\beta^0([b,x,h])$ concave non-decreasing function of the buffer state b. Now assume that $v_\beta^m([b,x,h])$ is concave non-decreasing in b, we have to prove that $v_\beta^{m+1}([b,x,h])$ is also concave non-decreasing in b. We recall that $v_\beta^{m+1}([b,x,h])$ is defined as:

$$v_\beta^{m+1}([b,x,h])=\max_a Q_\beta^m([b,x,h],a) \quad \text{(A.7)}$$

If v_β^m is concave in b it can be shown (see proof of Theorem 5.4.1) that $Q_\beta^m([b,x,h],a)$ is supermodular in the pair (b,a). Therefore,

$$Q_\beta^m([b',x,h],a')-Q_\beta^m([b,x,h],a')\geqslant Q_\beta^m([b',x,h],a)-Q_\beta^m([b,x,h],a) \quad \text{(A.8)}$$

for some $a'\geqslant a$ and $b'\geqslant b$. Using the concavity of $Q_\beta^m([b,x,h],a)$ in b, we can write:

$$Q_\beta^m([b',x,h],a')-Q_\beta^m([b,x,h],a')\geqslant Q_\beta^m([b'+p,x,h],a)-Q_\beta^m([b+p,x,h],a) \quad \text{(A.9)}$$

for some $p \geqslant 0$. Now substituting $b' = \bar{b}$ and $b = \bar{b} - p$ we obtain

$$Q^m_\beta([\bar{b}, x, h], a') - Q^m_\beta([\bar{b} - p, x, h], a') \geqslant Q^m_\beta([\bar{b} + p, x, h], a) - Q^m_\beta([\bar{b}, x, h], a) \ , \tag{A.10}$$

and rearranging the terms:

$$Q^m_\beta([\bar{b} + p, x, h], a) - Q^m_\beta([\bar{b}, x, h], a') \leqslant Q^m_\beta([\bar{b}, x, h], a) - Q^m_\beta([\bar{b} - p, x, h], a') \ . \tag{A.11}$$

Taking $a' = a = \text{argmax}_a \, Q^m_\beta([\bar{b}, x, h], a)$ and using (A.7) we get:

$$v^{m+1}_\beta([\bar{b} + p, x, h]) - v^{m+1}_\beta([\bar{b}, x, h]) \leqslant v^{m+1}_\beta([\bar{b}, x, h]) - v^{m+1}_\beta([\bar{b} - p, x, h]) \ , \tag{A.12}$$

that proves concavity of v^{m+1}_β in b. □

A.2 Proof of Theorem 5.4.2

Proof. Since

$$\pi^\star_\beta([b, x, h]) = \underset{a}{\text{argmax}} \left\{ Q_\beta([b, x, h], a) \right\} \ , \tag{A.13}$$

in order to prove that the optimal policy is non-decreasing in the channel state h, we have to prove that $Q_\beta([b, x, h], a)$ is supermodular in the pair (h, a). Now $Q_\beta([b, x, h], a)$ is defined as:

$$Q_\beta([b, x, h], a) = r(s, a; \beta) + \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\beta(s') \tag{A.14}$$

where the first term in the right hand side of the equation, i.e., $r(s, a; \beta)$, is supermodular in (h, a) since it does not depend on h and it is monotonically non-decreasing in a. Let us rewrite the second term as:

$$\begin{aligned} Q_1([b, x, h], a) &= \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\beta(s') & \text{(A.15)} \\ &= \sum_{h' \in \mathcal{H}} \sum_{x' \in \mathcal{X}} \sum_{e_\text{in}=0}^{E} p_h(h'|h) p_x(x'|x) p_{e_\text{in}}(e_\text{in}|x) v_\beta([b + e_\text{in} - e_\text{out}(h, a), x', h']) \ , & \text{(A.16)} \end{aligned}$$

where we used the independence of the channel evolution and of the source state evolution. We need to prove that the sum in (A.4) is supermodular in (h, a) for any $b \in \mathcal{B}$ and $x \in \mathcal{X}$

that is:

$$\begin{aligned}
&\sum_{x'\in\mathcal{X}} p_x(x'|x) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x) \sum_{h'\in\mathcal{H}} p_h(h'|h_1) v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_1),x',h']) \\
&\qquad - \sum_{x'\in\mathcal{X}} p_x(x'|x) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x) \sum_{h'\in\mathcal{H}} p_h(h'|h_1) v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_2),x',h']) \\
\geqslant\ &\sum_{x'\in\mathcal{X}} p_x(x'|x) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x) \sum_{h'\in\mathcal{H}} p_h(h'|h_2) v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_1),x',h']) \\
&\qquad - \sum_{x'\in\mathcal{X}} p_x(x'|x) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x) \sum_{h'\in\mathcal{H}} p_h(h'|h_2) v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])
\end{aligned} \tag{A.17}$$

for $h_1 \geqslant h_2$, $a_1 \geqslant a_2$, for any $x \in \mathcal{X}$ and any $b \in \mathcal{B}$. Getting rid of the constant terms (A.17) can be rewritten as:

$$\begin{aligned}
&\sum_{h'\in\mathcal{H}} p_h(h'|h_1) \left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_2),x',h'])\right) \\
&\geqslant \sum_{h'\in\mathcal{H}} p_h(h'|h_2) \left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])\right) .
\end{aligned} \tag{A.18}$$

Since $p_h(h'|h)$ is stochastically increasing in h, a sufficient condition for (A.18) to hold (see Lemma 4.7.2 in [101]) is that

$$\begin{aligned}
&v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_1,a_2),x',h']) \\
&\geqslant v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])
\end{aligned} \tag{A.19}$$

i.e., v_β is supermodular in (h,a). Since b, x', h', e_{in} are fixed, and $e_{\text{out}}(h,a)$ is monotonically increasing in h, (A.19) can be rewritten (with some abuse of notation) as:

$$v_\beta([(b',a'),x',h']) - v_\beta([(b',a),x',h']) \geqslant v_\beta([(b,a'),x',h']) - v_\beta([(b,a),x',h']) \tag{A.20}$$

where $b' \geqslant b$ and $a' \geqslant a$. But this is the condition for the supermodularity of v_β in (b,a) that holds for Theorem 5.4.1. □

A.3 Proof of Theorem 5.4.3

Proof. Since

$$\pi^\star_\beta([b,x,h]) = \underset{a}{\operatorname{argmax}} \left\{Q_\beta([b,x,h],a)\right\} , \tag{A.21}$$

in order to prove that the optimal policy is non-decreasing in the source state x, we have to prove that $Q_\beta([b,x,h],a)$ is supermodular in the pair (x,a). Now $Q_\beta([b,x,h],a)$ is defined as:

$$Q_\beta([b,x,h],a) = r(s,a;\beta) + \sum_{s'\in\mathcal{S}} p(s'|s,a)v_\beta(s') \tag{A.22}$$

where the first term in the right hand side of the equation, i.e., $r(s,a;\beta)$, is supermodular in (x,a) since it does not depend on x and it is monotonically non-decreasing in a. Let us rewrite the second term as:

$$\begin{aligned} Q_1([b,x,h],a) &= \sum_{s'\in\mathcal{S}} p(s'|s,a)v_\beta(s') && \text{(A.23)} \\ &= \sum_{h'\in\mathcal{H}} \sum_{x'\in\mathcal{X}} \sum_{e_{\text{in}}=0}^{E} p_h(h'|h)p_x(x'|x)p_{e_{\text{in}}}(e_{\text{in}}|x)v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a),x',h'])\ , && \text{(A.24)} \end{aligned}$$

where we used the independence of the channel evolution and of the source state evolution. We need to prove that the sum in (A.4) is supermodular in (x,a) for any $b \in \mathcal{B}$ and $h \in \mathcal{H}$ that is:

$$\begin{aligned} &\sum_{h'\in\mathcal{H}} p_h(h'|h) \sum_{x'\in\mathcal{X}} p_x(x'|x_1) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_1)v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h']) \\ &\quad - \sum_{h'\in\mathcal{H}} p_h(h'|h) \sum_{x'\in\mathcal{X}} p_x(x'|x_1) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_1)v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_2),x',h']) \\ \geqslant\ &\sum_{h'\in\mathcal{H}} p_h(h'|h) \sum_{x'\in\mathcal{X}} p_x(x'|x_2) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_2)v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h']) \\ &\quad - \sum_{h'\in\mathcal{H}} p_h(h'|h) \sum_{x'\in\mathcal{X}} p_x(x'|x_2) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_2)v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_2),x',h']) \end{aligned} \tag{A.25}$$

for $x_1 \geqslant x_2$, $a_1 \geqslant a_2$, for any $h \in \mathcal{H}$ and any $b \in \mathcal{B}$. Getting rid of the constant terms (A.25) can be rewritten as:

$$\begin{aligned} &\sum_{x'\in\mathcal{X}} p_x(x'|x_1) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_1) \left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_2),x',h'])\right) \\ &\geqslant \sum_{x'\in\mathcal{X}} p_x(x'|x_2) \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_2) \left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h']) - v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])\right)\ . \end{aligned} \tag{A.26}$$

Since $p_x(x'|x)$ is stochastically increasing in x, a sufficient condition for (A.26) to hold (see Lemma 4.7.2 in [101]) is that

$$\begin{aligned}&\sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_1)\left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_2),x',h'])\right)\\ &\quad\geqslant \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_2)\left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])\right)\ .\end{aligned} \tag{A.27}$$

for each $x_1 \geqslant x_2$. If the energy income from states x_i takes values in disjoint ordered sets, i.e., $e_{\text{in}}^{x_i} \in \mathcal{E}^{x_i} = \{E_{\text{min}}^{x_i}, E_{\text{min}}^{x_i}+1, \dots, E_{\text{max}}^{x_i}\}$, $\mathcal{E}^{x_i} \cap \mathcal{E}^{x_j} = \varnothing$ for $i \neq j$, $\cup_i \mathcal{E}^{x_i} = \{0,1,\dots,E\}$, $E_{\text{max}}^{x_i} < E_{\text{min}}^{x_{i+1}}$, and since the sums in (A.27) are convex combinations we can write:

$$\begin{aligned}&\sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_1)\left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_2),x',h'])\right)\\ &\quad\geqslant v_\beta([b+E_{\text{min}}^{x_1}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+E_{\text{min}}^{x_1}-e_{\text{out}}(h,a_2),x',h'])\\ &\quad\geqslant v_\beta([b+E_{\text{max}}^{x_2}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+E_{\text{max}}^{x_2}-e_{\text{out}}(h,a_2),x',h'])\\ &\quad\geqslant \sum_{e_{\text{in}}=0}^{E} p_{e_{\text{in}}}(e_{\text{in}}|x_2)\left(v_\beta([b+e_{\text{in}}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+e_{\text{in}}-e_{\text{out}}(h_2,a_2),x',h'])\right)\ .\end{aligned} \tag{A.28}$$

Thus (A.27) holds if the innermost inequality in (A.28) holds. Rearranging the term we can write:

$$\begin{aligned}&v_\beta([b+E_{\text{min}}^{x_1}-e_{\text{out}}(h,a_1),x',h'])-v_\beta([b+E_{\text{max}}^{x_2}-e_{\text{out}}(h,a_1),x',h'])\\ &\quad\geqslant v_\beta([b+E_{\text{min}}^{x_1}-e_{\text{out}}(h,a_2),x',h'])-v_\beta([b+E_{\text{max}}^{x_2}-e_{\text{out}}(h,a_2),x',h'])\ .\end{aligned} \tag{A.29}$$

that holds since e_{out} is monotonically increasing in a and v_β is concave and increasing in b for Lemma A.1.1. □

List of Publications

The work presented in this thesis has appeared in the articles reported below.

Journal papers

[J1] **D. Zordan**, B. Martinez, I. Vilajosana and M. Rossi, “On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking”, *ACM Transactions on Sensor Networks*, accepted for publication.

[J2] **D. Zordan**, T. Melodia and M. Rossi, “Optimal Temporal Compression for Energy Harvesting Sensor Networks”, submitted to *IEEE Transactions on Wireless Communications.*

Conference papers

[C1] **D. Zordan**, G. Quer , M. Zorzi and M. Rossi, “Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields”, in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, December 2010, Houston, TX, US.

[C2] M. Miozzo, **D. Zordan**, P. Dini and M. Rossi, “SolarStat: Modeling Photovoltaic Sources through Stochastic Markov Processes”, submitted to *IEEE International Energy Conference (EnergyCon)*, May 2014, Dubrovnik, Croatia

Bibliography

[1] G. Quer, D. Zordan, R. Masiero, M. Zorzi, and M. Rossi, "WSN-Control: Signal Reconstruction through Compressive Sensing in Wireless Sensor Networks," in *IEEE SenseApp Workshop*, Denver, CO, US, Dec. 2010, pp. 937–944.

[2] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communication Magazine*, pp. 70–87, Apr. 2007.

[3] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for Wireless Sensor Networks," in *IEEE INFOCOM*, vol. 3, 2002, pp. 1567–1576.

[4] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-Efficient Data Representation and Routing for Wireless Sensor Networks Based on a Distributed Wavelet Compression Algorithm," in *Information Processing in Sensor Networks (IPSN)*, Nashville, TN, US, Apr. 2006.

[5] B. Wang, K. C. Chua, V. Srinivasan, and W. Wan, "Information Coverage in Randomly Deployed Wireless Sensor Network," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2994–3003, Aug. 2007.

[6] M. C. Vuran and I. F. Akyildiz, "Spatial Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 316–329, Apr. 2006.

[7] S. Bandyopadhyay, Q. Tian, and E. J. Coyle, “Spatio-temporal sampling rates and energy efficiency in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1339–1352, Dec. 2005.

[8] Y. Yu, B. Krishnamachari, and V. K. Prasanna, “Data Gathering with Tunable Compression in Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 276–287, Feb. 2008.

[9] D. Ganesan, D. Estrin, and J. Heidemann, “DIMENSIONS: Why do we need a new data handling architecture for sensor networks?” in *ACM Workshop on Hot Topics in Networks*, Princeton, NJ, US, Oct. 2002, pp. 143–148.

[10] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 4036–4048, Apr. 2006.

[11] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[12] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, “Data Acquisition through joint Compressive Sensing and Principal Component Analysis,” in *IEEE Globecom 2009*, Honolulu, HI, US, Nov.-Dec. 2009.

[13] R. Crepaldi, S. Friso, A. F. Harris III, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi, “The Design, Deployment, and Analysis of SignetLab: A Sensor Network Testbed and Interactive Management Tool,” in *IEEE Tridentcom*, Orlando, FL, US, May 2007.

[14] C. J. Willmott and K. Matsuura, “Global Climate Resource Pages,” Aug. 2009. [Online]. Available: http://climate.geog.udel.edu/~climate/

[15] D. Catrein and R. Mathar, “Gaussian Random Fields as a Model for Spatially Correlated Log-Normal Fading,” in *ATNAC*, Adelaide, South Australia, Dec. 2008.

[16] A. Jindal and K. Psounis, “Modeling Spatially Correlated Data in Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 466–499, Nov. 2006.

[17] N. Cressie, *Statistics for Spatial Data.* John Wiley, 1993.

[18] C. K. Wikle and N. Cressie, "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, vol. 86, no. 4, pp. 815–829, 1999.

[19] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 24–36, Feb. 2011.

[20] D. Zordan, G. Quer, and M. Rossi, "MATLAB Implementation of the Space-Time correlated signals generator," May 2010. [Online]. Available: http://www.dei.unipd.it/~rossi/software.html

[21] P. Abrahamsen, "A review of Gaussian random fields and correlation functions," Norwegian Computing Center, Oslo, Tech. Rep., 1997.

[22] P. Whittle, "On Stationary processes in the plane," *Biometrika*, vol. 41, no. 3-4, pp. 434–449, 1954.

[23] B. Matérn, "Spatial variation. Stochastic models and their application to some problems in forest surveys and other sampling investigations," *Meddelanden fran Statens Skogsforskningsinstitut*, vol. 49, no. 5, 1960.

[24] A. M. Yaglom, *An Introduction to the Theory of Stationary Random Functions*. Dover Phoenix Editions, 1962.

[25] N. Benvenuto and G. Cherubini, *Algorithms for Communications Systems and their Applications*. Wiley, 2002.

[26] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[27] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. Biswas, and K. Moessner, "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Communications Magazine*, vol. 51, no. 6, 2013.

[28] T. Chen, Y. Yang, H. Zhang, H. Kim, and K. Horneman, "Network energy saving technologies for green wireless access networks," *Wireless Communications, IEEE*, vol. 18, no. 5, pp. 30–38, 2011.

[29] G. Piro, M. Miozzo, G. Forte, N. Baldo, L. A. Grieco, G. Boggia, and P. Dini, "HetNets Powered by Renewable Energy Sources," *IEEE Internet Computing*, vol. 17, no. 1, pp. 32–39, 2013.

[30] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, 2013, Accepted for Publication.

[31] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with Energy Harvesting Nodes in Fading Wireless Channels: Optimal Policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, 2011.

[32] M. Gregori and M. Payaró, "Energy-Efficient Transmission for Wireless Energy Harvesting Nodes," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1244–1254, 2013.

[33] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Control of wireless networks with rechargeable batteries," *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 581–593, 2010.

[34] N. Michelusi, K. Stamatiou, and M. Zorzi, "Transmission Policies for Energy Harvesting Sensors with Time-Correlated Energy Supply," *IEEE Transactions on Communications, to appear*, vol. PP, no. 99, pp. 1–14, 2013.

[35] J. Jeong and D. Culler, "A Practical Theory of Micro-Solar Power Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 1, Apr. 2012.

[36] J. Dave, P. Halpern, and H. Myers, "Computation of incident solar energy," *IBM Journal of Research and Development*, vol. 19, no. 6, pp. 539–549, Nov. 1975.

[37] A. F. Zobaa and R. C. Bansal, *Handbook of Renewable Energy Technology.* World Scientific Publishing Co., 2011, Edited Book.

[38] NREL, National Renewable Energy Laboratory, "Renewable Resource Data Center," http://www.nrel.gov/rredc/.

[39] F. Lindholm, J. Fossum, and E. Burgess, "Application of the superposition principle to solar-cell analysis," *IEEE Transactions on Electron Devices*, vol. 26, no. 3, pp. 165–171, 1979.

[40] A. Luque and S. Hegedus, *Handbook of Photovoltaic Science and Engineering.* Wiley, 2003.

[41] F. Ongaro, S. Saggini, S. Giro, and P. Mattavelli, "Two-dimensional MPPT for photovoltaic energy harvesting systems," in *IEEE Workshop on Control and Modeling for Power Electronics (COMPEL)*, Boulder, CO, US, Jun. 2010.

[42] D. P. Hohm and M. E. Ropp, "Comparative study of maximum power point tracking algorithms," *Wiley Progress in Photovoltaics: Research and Applications*, vol. 11, no. 1, pp. 47–62, Jan. 2003.

[43] D. Brunelli, L. Benini, C. Moser, and L. Thiele, "An Efficient Solar Energy Harvester for Wireless Sensor Nodes," in *IEEE Design, Automation and Test in Europe (DATE)*, Munich, Germany, Mar. 2008, pp. 104–109.

[44] J. S. Simonoff, *Smoothing Methods in Statistics.* Springer-Verlag, 1996.

[45] Solarbotics Ltd., "SCC-3733 Monocrystalline solar cells," http://solarbotics.com/.

[46] Zolertia, "Z1 Platform Datasheet," http://www.zolertia.com/.

[47] *Solar-Stat: an Open Source Framework to Model Photovoltaic Sources through stochastic Markov Processes*, 2013. [Online]. Available: http://www.dei.unipd.it/~rossi/software.html

[48] L. Atzoria, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[49] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys Tutorials, to appear*, vol. PP, no. 99, pp. 1–18, 2012.

[50] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *ACM Communications*, vol. 47, no. 6, pp. 34–40, jun 2004.

[51] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, march-april 2006.

[52] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," in *ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, Baltimore, MD, US, Nov. 2004.

[53] C. Kappler and G. Riegel, "A Real-World, Simple Wireless Sensor Network for Monitoring Electrical Energy Consumption," in *European Workshop on Wireless Sensor Networks (EWSN)*, Berlin, Germany, 2004.

[54] F. Benzi, N. Anglani, E. Bassi, and L. Frosini, "Electricity Smart Meters Interfacing the Households," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4487–4494, 2011.

[55] S. Dodson, "The Internet of Things," *The Guardian*, oct 2003.

[56] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communication Magazine*, vol. 14, no. 2, pp. 70–87, Apr. 2007.

[57] S. Pattem and B. Krishnamachari, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, US, 2004.

[58] M. Sharaf, J. Beaver, A. Labrinidis, R. Labrinidis, and P. Chrysanthis, "TiNA: A Scheme for Temporal Coherency-Aware in-Network Aggregation," in *ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, San Diego, CA, US, sep 2003.

[59] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, "Energy Harvesting for Autonomous Wireless Sensor Networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, 2010.

[60] F. Marcelloni and M. Vecchio, “An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks,” *The Computer Journal*, vol. 52, no. 8, pp. 969–987, Nov. 2009.

[61] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, “Lightweight temporal compression of microclimate datasets,” in *IEEE International Conference on Local Computer Networks (LCN)*, Tampa, FL, US, Nov. 2004.

[62] A. van der Byl, R. Neilson, and R. Wilkinson, “An evaluation of compression techniques for Wireless Sensor Networks,” in *AFRICON*, Nairobi, Kenya, sep 2009.

[63] Y. Liang, “Efficient temporal compression in wireless sensor networks,” in *International Conference on Local Computer Networks (LCN)*, Bonn, Germany, oct. 2011.

[64] C. M. Sadler and M. Martonosi, “Data compression algorithms for energy-constrained devices in delay tolerant networks,” in *International Conference on Embedded Networked Sensor Systems (SenSys)*, Boulder, CO, US, nov 2006.

[65] K. C. Barr and K. Asanović, “Energy-aware lossless data compression,” *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, Aug. 2006.

[66] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, “Practical data compression in wireless sensor networks: A survey,” *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37–59, 2012.

[67] J.-L. Lu, F. Valois, and M. Dohler, “Optimized Data Aggregation in WSNs Using Adaptive ARMA,” in *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Venice, Italy, jul 2010.

[68] C. Liu, K. Wu, and J. Pei, “An energy-efficient data collection framework for wireless sensor networks by exploiting spatio-temporal correlation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010–1023, jul 2007.

[69] N. D. Pham, T. D. Le, and H. Choo, “Enhance exploring temporal correlation for data collection in WSNs,” in *IEEE International Conference on Research, Innovation and Vision for the Future (RIVF)*, Ho Chi Minh City, Vietnam, Jul. 2008.

[70] H. Chen, J. Li, and P. Mohapatra, “RACE: time series compression with rate adaptivity and error bound for sensor networks,” in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Fort Lauderdale, FL, US, Oct. 2004.

[71] F. Marcelloni and M. Vecchio, “Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization,” *Elsevier Information Sciences*, vol. 180, no. 10, pp. 1924–1941, May 2010.

[72] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii –xxxiv, Feb. 1992.

[73] D. Donoho, M. Vetterli, R. DeVore, and I. Daubechies, “Data compression and harmonic analysis,” *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2435–2476, 1998.

[74] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA, US: Academic Press Professional, Inc., 1990.

[75] G. M. Phillips, *Interpolation and Approximation by Polynomials*. Springer, 2003.

[76] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, “To Compress or Not To Compress: Processing vs Transmission Tradeoffs for Energy Constrained Sensor Networking,” Department of Information Engineering, University of Padova, Padova, Italy, Tech. Rep., 2012.

[77] R. B. Davies and D. S. Harte, “Tests for Hurst effect,” *Biometrika*, vol. 74, no. 1, pp. 95–101, 1987.

[78] D. Zordan, G. Quer, M. Zorzi, and M. Rossi, “Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, US, Dec. 2011.

[79] P. Abrahamsen, “A Review of Gaussian Random Fields and Correlation Functions,” Norwegian Computing Center, Box 114, Blindern, N-0314 Oslo, Norway, Tech. Rep., 1997.

[80] L. Bierl, “MSP430 Family Mixed-Signal Microcontroller Application Reports,” Texas Instruments Incorporated, Tech. Rep., 2000.

[81] Chipcon, “SmartRF CC2420: 2.4 GHz IEEEE 802.15.4/ZigBee-ready RF Transceiver,” Texas Instruments Incorporated, Tech. Rep., 2007.

[82] IEEE P802.15 Working Group, “Std. IEEE 802.15.4-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs),” in *IEEE Standard*, Oct. 2003, pp. 1–89.

[83] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Prentice Hall, 1971.

[84] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, “Sensing, Compression and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[85] I. Vilajosana, G. Khazaradze, E. Suriñach, E. Lied, and K. Kristensen, “Snow avalanche speed determination using seismic methods,” *Elsevier Cold Regions Science and Technology*, vol. 49, no. 1, pp. 2–10, 2007.

[86] K. D. Donohue, “Wav data files for class examples and laboratory experiments,” http://www.engr.uky.edu/~donohue/ee422/Data/DataEE422.htm, 2013.

[87] G. Mao, B. D. O. Anderson, and B. Fidan, “Online Calibration of Path Loss Exponent in Wireless Sensor Networks,” in *Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE*, San Francisco, CA, US, Nov. 2006.

[88] R. Crepaldi, S. Friso, A. F. Harris III, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi, “The Design, Deployment, and Analysis of SignetLab: A Sensor Network Testbed and Interactive Management Tool,” in *IEEE Tridentcom*, Orlando, FL, US, May 2007.

[89] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, “SYNAPSE++: Code Dissemination in Wireless Sensor Networks using Fountain Codes,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 12, pp. 1749–1765, Dec. 2010.

[90] O. Yang and W. Heinzelman, “Modeling and Performance Analysis for Duty-Cycled MAC Protocols with Applications to S-MAC and X-MAC,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 905–921, Jun. 2012.

[91] T. Wang, W. Heinzelman, and A. Seyedi, "Link Energy Minimization for Wireless Sensor Networks," *Elsevier Ad Hoc Networks*, vol. 10, no. 3, pp. 569–585, 2012.

[92] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking," *ACM Transactions on Sensor Networks*, 2014, Accepted for Publication.

[93] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Salt Lake City, Utah, US, Jun. 2011, pp. 46–54.

[94] J. Yang and S. Ulukus, "Optimal Packet Scheduling in an Energy Harvesting Communication System," *Communications, IEEE Transactions on*, vol. 60, no. 1, pp. 220–230, 2012.

[95] P. Castiglione, O. Simeone, E. Erkip, and T. Zemen, "Energy Management Policies for Energy-Neutral Source-Channel Coding," *Communications, IEEE Transactions on*, vol. 60, no. 9, pp. 2668–2678, 2012.

[96] C. Tapparello, O. Simeone, and M. Rossi, "Dynamic Compression-Transmission for Energy-Harvesting Multihop Networks With Correlated Sources," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–13, 2013, Accepted for Publication.

[97] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Vol. II*, 4th ed. Athena Scientific, 2012.

[98] E. Altman, *Constrained Markov Decision Processes: Stochastic Modeling.* London: Chapman and Hall CRC, 1999.

[99] D. Djonin and V. Krishnamurthy, "Structural Results on Optimal Transmission Scheduling over Dynamical Fading Channels: A Constrained Markov Decision Process Approach," in *Wireless Communications*, ser. The IMA Volumes in Mathematics and its Applications. Springer New York, 2007, vol. 143, pp. 75–98.

[100] C. Sun, E. Stevens-Navarro, V. Shah-Mansouri, and V. W. Wong, “A Constrained MDP-based Vertical Handoff Decision Algorithm for 4G Heterogeneous Wireless Networks,” *Wireless Networks*, vol. 17, no. 4, pp. 1063–1081, May 2011.

[101] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Jon Wiley & Sons, Inc , 2005.

[102] F. J. Beutlerand and K. W. Ross, “Optimal Policies for Controlled Markov Chains with a Constraint,” *Journal of Mathematical Analysis and Applications*, vol. 112, no. 1, pp. 236–252, Nov. 1985.

[103] A. Goldsmith, *Wireless Communications.* Cambridge University Press, 2005.

[104] C. Tan and N. Beaulieu, “On first-order Markov modeling for the Rayleigh fading channel,” *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2032–2040, 2000.

[105] M. Rossi, L. Badia, and M. Zorzi, “SR ARQ Delay Statistics on N-State Markov Channels with Non-Instantaneous Feedback,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1526–1536, Jun. 2006.

[106] M. Miozzo, D. Zordan, P. Dini, and M. Rossi, “SolarStat: Modeling Photovoltaic Sources through Stochastic Markov Processes,” Submitted for Publication, 2013.