Energy management of a Wireless Sensor Network at application level

Olesia Mokrenko

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :
Olesia MOKRENKO

Le 20 novembre 2015

Titre :

Gestion de l'énergie dans un réseau de capteurs au niveau application

---

EDSYS : Automatique 4200046

Unité de recherche :

CEA - LETI

Directeur(s) de Thèse :

Carolina ALBEA SANCHEZ, Dr., UPS, LAAS, CNRS
Suzanne LESECQ, Dr., HDR, CEA

Rapporteurs :

Didier GEORGES, Pr., Grenoble-INP
Marc JUNGERS, Dr., HDR, Université de Lorraine, CNRS

Autre(s) membre(s) du jury :

Mirko FIACCHINI, Dr., GIPSA-lab, CNRS
Fabio GOMEZ-ESTERN, Pr., Loyola University Andalusia
Diego PUSCHINI, Dr., CEA-LETI
Daniel QUEVEDO, Pr., University of Paderborn

Присвячується моїм батькам, сестрам і тобі, Адріан.

Contents

Remerciements

Cette thèse a été effectuée en collaboration entre le Laboratoire Infrastructure et Atelier Logiciel pour Puces (LIALP) du Département Architectures, Conception et Logiciels Embarqués (DACLE) du CEA-Leti, Grenoble, France ; et le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) de l'Université Toulouse III Paul Sabatier, France.

Tous mes remerciements vont aux membres du Jury. Merci aux rapporteurs, M. Didier Georges, professeur à l'Institut Polytechnique de Grenoble, France ; et M. Marc Jungers chargé de recherche CNRS à l'Université de Lorraine, France. Leurs critiques et remarques m'ont permis d'améliorer la présentation de ce travail. Mes très sincères reconnaissances vont également aux autres membres du Jury, particulièrement à M. Yann Labit, professeur à l'Université de Toulouse III, Président du Jury de thèse ; M. Mirko Fiacchini, chargé de recherche CNRS à l'Institut Polytechnique de Grenoble, examinateur de thèse ; M. Fabio Gomez-Estern, professeur au Loyola University Andalusia, Espagne, examinateur de thèse ; M. Diego Puschini, ingénieur chercheur au CEA-Leti, examinateur de thèse ; M. Daniel Quevedo, professeur au University of Paderborn, Allemagne, examinateur de thèse. Merci pour l'intérêt porté à mes travaux.

Je remercie vivement Mme. Suzanne Lesecq, ma co-directrice au CEA, une personne essentielle pour la réussite de cette thèse. Ses conseils, sa disponibilité, mais surtout son soutien et sa patience m'ont permis de mener cette thèse jusqu'au bout. Elle a toujours été présente pendant ces trois années de thèse. Un très grand merci à Mme. Carolina Albea Sanchez, directrice de cette thèse, pour ses conseils pertinents et sa rigueur scientifique. Je tiens très particulièrement à remercier encore M. Diego Puschini, mon encadrant au CEA. De nombreuses discutions avec lui ont contribué très activement à la réussite de ce travail. Ses conseils et son soutien, surtout dans les moments de doutes dont cette thèse a été parsemé, m'ont énormément aidé. Merci pour votre implication.

Un grand merci au chef de laboratoire LIALP M. Vincent Olive pour sa confiance. Merci également à M. Marc Belleville, directeur scientifique du DACLE, pour ses précieux conseils.

Merci également à tous les membres de ce laboratoire pour les conseils techniques et scientifiques, particulièrement à Warody, Isabel, Olivier D., François, Maxime et Youssouf. Merci encore aux autres thésards et stagiaires du labo, pas seulement pour leurs conseils et discutions pertinents mais aussi pour leur amitié. Merci particulièrement aux filles Isabel, Adja et Geneviève. Merci aussi aux autres personnes qui sont passés par le labo avec qui j'ai eu l'occasion de travailler.

Je remercie l'ensemble de l'équipe de secourisme de la Croix Rouge française pour tous ces bons moments passés à vos cotés, qui ont fait de ces trois années une période inoubliable. Un grand merci à Claire, Mélanie, Sarah et Mathilde.

Finalement, je tiens aussi à remercier ma famille pour son soutien et l'immesurable effort de supporter la distance. Enfin, je remercie affectueusement Adrien, qui a choisi de m'accompagner dans cette expérience et qui est toujours présent depuis le début, malgré le temps, la distance et mon caractère.

List of Acronyms

ADC Analog-to-Digital Converter 14
DPM Dynamic Power Management 18
DVS Dynamic Voltage Scaling 18
HDS Hybrid Dynamic System 53
HVAC Heating, Ventilation and Air Conditioning 4
LP Linear Programming 22
LTI Linear Time-Invariant 26
LLC Logical Link Control 11
MAC Media Access Control 11
MCU Micro-Controller Unit 15
MILP Mixed Integer Linear Programming 47
MIMO Multi-Input Multi-Output 22
MIP Mixed Integer Programming 22
MIQP Mixed Integer Quadratic Programming 22
MPC Model Predictive Control 22
OCV Open Circuit Voltage 86
OSI Open System Interconnection 10
QP Quadratic Programming 22
QoS Quality of Service 13
RAM Random Access Memory 15
RTC Real Time Clock 33
SISO Single-Input Single-Output 22
SOC State of Charge 86
WSN Wireless Sensor Network 1

Notation and Definitions

- $\mathbb{N}^*$ is the set of natural numbers such that $\mathbb{N}^* = \mathbb{N}\backslash\{0\}$.
- $\mathbb{R}^n$ is the real vector space of dimension n.
- $\mathbb{R}^{n\times m}$ is the set of real-valued matrices with dimension $n \times m$.
- $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers, including zero.
- $\mathbb{R}_{>0}$ is the set of non-negative real numbers, excluding zero.
- $\{0,1\}^n$ is the vector with binary components (0 or 1), of dimension n.
- $\{0,1\}^{n\times m}$ is the matrix with binary components (0 or 1), of dimension $n \times m$.
- $\mathbf{I}_n$ is the identity matrix of dimension $n \times n$.
- $\mathbf{0}$ denotes a vector or a matrix filled with 0, of appropriate dimension.
- $\text{diag}(A_1, \ldots, A_n)$ is a matrix with $A_1, \ldots, A_n$ as diagonal elements.
- e_h denotes column h of the identity matrix, or the h^{th} vector of the Euclidean basis.
- $\left\|x\right\| = \sqrt{x_1^2 + \cdots + x_n^2}$ is called l_2 norm or Euclidean norm where $x \in \mathbb{R}^n$ and $x = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix}^T$.
- $\left\|x\right\|_1 = \sum_{i=1}^{n} \left|x_i\right|$ is called l_1 norm or Taxicab norm where $x \in \mathbb{R}^n$ and $x = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix}^T$.
- $\left\|x\right\|_\infty = \max(|x_1|, ..., |x_n|)$ is called l_∞ norm where $x \in \mathbb{R}^n$ and $x = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix}^T$.

Introduction

"Radio has not future. "
"Wireless [telegraphy] is all very well but I'd rather send a massage by a boy on a pony!"

Lord Kelvin (Sir William Thomson),
British physicist originally of absolute zero.

Context

Technological and technical developments performed in the areas of wireless communication, micro-electronics and system integration have led to the advent of a new generation of large-scale sensor networks suitable for various applications [1]. Consider a set of small electronic devices (the so-called sensor nodes), autonomous, equipped with sensors and able to communicate with each other wirelessly. Together they form a Wireless Sensor Network (WSN) capable of monitoring a phenomenon of interest, and possibly react on the environment. They can provide high level information about this phenomenon to users by the combination of measurements taken by the various sensors and then communicated via the wireless medium.

This technology promises to revolutionize our way of life, work and interact with the physical environment around us. Sensor nodes, which able to communicate wirelessly together with distributed computing capabilities, allow developing new applications that were impractical or too expensive a few years ago. Today, tiny and inexpensive sensor nodes can be literally scattered on roads, bridges, buildings wings of planes or forests, creating a kind of "second digital skin" which can detect various physical phenomena such as vibrations created by earthquake or the change in the shape of a mechanical structure, or fire appearance and evolution in forests. As a consequence, many applications deal with the detection and monitoring of disasters (earthquake, flooding), environmental monitoring and mapping of biodiversity, intelligent building, advanced farming techniques, surveillance and preventive maintenance of machinery, medicine and health, logistics and intelligent transportation systems.

WSNs are often characterized by a dense deployment of nodes in large-scale environments with various limitations. These limitations are related to processing and memory capabilities, radio communication ranges, but also to energy resources as the sensor nodes may be powered by batteries. Note that even if sensor nodes are connected to power lines, they must be power-efficient because it is not acceptable from an ecology viewpoint, to drastically increase the number of power plants just to feed all these new devices. For sensor networks powered by batteries, changing the batteries has an extra cost related to their recharge and/or change that must be taken into account. Moreover, as the batteries in the network will certainly not be drained at the same time, the network maintenance teams change all the batteries during a unique intervention, leading to a suboptimal discharge of some of the batteries. Also, the sensor nodes may be placed in locations that are hard to access, for safety or economic reasons. Indeed, it is widely recognized that the energy limitation is an unavoidable issue in the design and deployment of WSNs because it imposes strict constraints on the network operation. Basically, the power consumption of the sensor nodes plays an important role in the life of the network. This aspect has become the predominant performance criterion for sensor networks. If we want the sensor network to perform its functionality satisfactorily as long as possible, these energy constraints implies compromises between different activities both at the node level and at the network layers.

Objective and Motivation

Energy management of WSN must be tackled at different levels, from node level to network level. Indeed, the sensor nodes themselves should be energy aware. In general, saving energy eventually consists in finding a compromise between the different energy consuming activities. The literature related to WSNs recognizes that the radio part of a sensor node is the most consuming part of the nodes [2]. The research community has also proposed various energy-aware protocols at all layers of the Open System Interconnection (OSI) model. They can naturally be split into two classes. The first class of protocols has been developed in the area of ad-hoc networks[1]. However, the application of these protocols to each specific WSN unsurprisingly leads to a high complexity and energy cost unaffordable for sensor nodes powered by batteries. The second class of protocols is dedicated to sensor networks. They are often designed and tuned for a given application. However, many of them are applied and deployed for any application context despite their drawbacks, especially related to their power consumption. Also, many progresses have been done for the different units of a sensor node. Now,

[1] A wireless ad-hoc network is a decentralized type of wireless network. The network is ad-hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks.

the low power techniques are used to design all parts of a node. For instance, the techniques relate to the advanced control of the supply voltage and clock frequency in the computing unit or the control of the radio power while watching quality of link in the communication unit. Their combinations provide a low-power future of whole sensor node.

For all these reasons, it is necessary to combine these achievements in different network layers and at node level, a global vision on the WSN and its consumption. It allows to manage the heterogeneity of used communication protocols and sensor nodes. Using this global vision, the energy management strategies placed on top of the sensor network can be developed to extend the lifespan of whole WSN. Indeed, in many deployments, the controller at the application (monitoring, control) receives from the nodes more measurements than necessary, or measurements that are strongly correlated. Therefore, control strategies must be designed and implemented for the energy management at the application level that takes into account the application requests in terms of data quantity (amount of measurements). Basically, an appropriate management of data amount and node switching that send data might provide further energy savings leading to the extend of the network lifespan. They may implement innovative conservation techniques to improve the network performance, including maximizing of the network lifespan.

Keeping this in mind, the work performed in this PhD thesis mainly deals with the reduction of the energy consumption of a (wireless) sensor network with strategies implemented at the application level, taking into account several imposed constraints. Most of this work consists in designing power/energy management strategies based on control approaches. The strategies proposed must ensure that the constraints imposed by the application in terms of amount of received data are fulfilled, while the network so-called lifespan is extended, compared current "basic" deployments [2]. The network lifespan is defined as the time period when the network provides the required amount of data.

Contributions

This PhD thesis has been performed in coooperation between the Laboratoire d'Analyse et d'Architecture Systemes (LAAS) and the Laboratoire de l'Électronique et des Technologies de l'Information from the French Atomic and Alternative Energy Commission (CEA-LETI).

As stated above, this PhD thesis deals with the minimization of the energy consumption of a set of nodes at the application level by the employing control theory, taking into account some constraints imposed by an application. The

[2] A "basic" deployment means the situation without control where any node is activated without any priority (any control strategy).

proposed strategies must ensure that the constraints imposed by the application in terms of amount of received data are fulfilled, while the network lifespan is extended, compared to "basic" scheme of current deployments.

The power/energy management strategies require first to model the power/ energy consumption of the set of sensor nodes, taking into account real-life phenomena such as the fact that some nodes may become unreachable (for instance due to radio link disturbances) and come back in a reachable condition. The model must also take into account current nodes features, such as their capability to work in different functioning modes with different power consumptions. Note that the nodes may autonomously switch from one mode to another one, depending for instance on the level of radio disturbances.

Two control strategies have been proposed:

- a first one based on Model Predictive Control (MPC). This approach naturally takes into accounts the constraints that must be fulfilled by the controlled system, that is the sensor network;

- a second one based on a Hybrid Dynamic System (HDS) approach. Indeed, the controlled system is *per se* hybrid: dynamics that illustrate the charge/discharge of batteries are continuous while dynamics related to the change of the sensor mode are discrete.

A relevant common assumption of both control approaches is that the information related to the energy that remains in the batteries is accessible by the controller. Actually, there is no sensor that can directly measure this remaining energy. To resolve this problem, an estimation method of the remaining energy in the node battery must be implemented. However, this estimation method must exhibit a low computational complexity, especially if it is embedded in the sensor nodes, in order not to increase the energy consumption of nodes. Basically, this estimation algorithm will consume extra energy, this explains why it must be as simple as possible.

The main contributions of this work are:

- modeling of the energy consumption in the whole set of sensors for both approaches. In the case of the MPC approach (Chapter 2), the model is a state space one in discrete time. The state represents the remaining energy in the node batteries. The control vector contains binary values with constraints which express the fact that a node can be in a unique functioning mode. For the HDS approach (Chapter 3), the continuous time model describes the energy evolution in the node batteries. The discrete part of the model describes when nodes swap their roles;

- development of two control strategies to reduce the energy consumption of the WSN. As mentioned before, they are based on the MPC approach

(Chapter 2) and on the HDS approach (Chapter 3). The common objective is to increase the WSN lifespan when a given service is assured. This is accomplished by the choice of nodes placed in the *Active* mode, while other nodes are either in the *Standby* mode or in the *Unreachable* condition. This choice is guided by a tread-off between the energy consumption and remaining energy in a node battery;

- implementation of a method to estimate the battery remaining capacity. After an analysis of the literature with respect to the estimation of the State-of-Charge of batteries, a low computation cost technique is chosen and adapted to be implemented. It has been successfully implemented in real-life experiments (Chapter 4);
- analyzes of various realistic scenarios in simulation in Matlab/Simulink environments for both MPC and HDS approaches (Chapters 2 to 4);
- implementation of both control strategies on a real test-bench. The results has been analyzed and compared (Chapter 4).

Report Organization

This report is composed of four chapters. Chapter 1 one summarizes the context of this PhD work, especially WSNs with some of their application fields. It introduces the energy consumption issues that are tackled in this PhD thesis at the application level. Then, the objectives are expressed and justified. In order to replace our contributions in the general context of power/energy management of a set of sensor nodes, different techniques that deal with the minimization of the energy consumption in a WSN at the different network layer are shortly discussed. Finally, this chapter presents the traditional architecture and power consumption of a sensor node.

The goal of the Chapter 2 is to propose an energy/power management strategy based on an MPC approach to increase the network lifespan. It firstly introduces the main MPC concepts. After that, the system is modeled, taking into account the different constraints that exist in real-life conditions. The MPC problem is then expressed in our context. Note that depending on the optimization cost, a Mixed Integer Quadratic Programming (MIQP) problem or a Mixed Integer Linear Programming (MILP) one is solved. Several realistic scenarios are considered to validate the proposed strategy.

Chapter 3 presents a second energy/power management strategy. Here, a HDS approach is considered. This chapter firstly introduces the hybrid dynamical phenomena. Then, the design of a hybrid scheduling law is used to describe the sensor network activity evolution with respect to its remaining energy. After that, the increase of the network lifespan and the minimization of the number

of switches using the HDS strategy are demonstrated. Simulations of a realistic test-case show the effectiveness of this second control approach.

Chapter 4 compares the both control strategies first in simulation, then on a real test-bench. This latter implements on-the-shelf devices. The assumptions common to the MPC and HDS approaches are discussed together with the peculiar assumptions to each control approach. The implementation details are also described. Then results are analyzed and compared.

Finally, the chapter "Conclusions and Perspectives" summarizes the main contributions and proposed future work directions.

Chapter 1

Energy in Wireless Sensor Networks

With the recent advances achieved over the last decade in highly integrated microelectronics, sensors and actuators, and low power wireless communication technologies, Wireless Sensor Network (WSN) have gained a significant interest from both academic and industrial communities due to their potential to monitor and control the physical world from remote locations which can be difficult to reach.

The WSN have an enormous potential for improving the functionalities of the different industrial fields, enhancing peoples daily lives, and monitoring ecological environment. They have already been used in a wide variety of applications such as industrial machine and process control, building and facility automation, fire rescue and medical monitoring, wild life tracking and agriculture monitoring. However, implementing such networks remains a challenging task due to hard requirements in terms of data processing and transmission capabilities as well as computational power and energy resources.

In this chapter, we summarize the state-of-the-art related to the application domains of WSNs, the requirements and challenges of WSN and the power consumption issues in a sensor node and at different layers of the WSN. From this the state-of-the-art, the problem formulation of this PhD thesis is presented.

1.1 WSNs and related applications

To understand the requirements and trade-offs of WSNs, it is helpful to briefly examine their origin and history. The early researches and developments in WSNs were initially driven by military applications. The first development of WSNs started in the 1980s by the distributed sensor networks program at the Defense Advanced Research Projects Agency (DARPA) in the USA [3]. This research program explored the challenges involved in implementing many spatially distributed low-cost and autonomous sensor nodes that collaborate with each other. The research results and testbeds generated many WSNs systems with uses mainly in the military domain. Examples of WSN applications include acoustic tracking sensors for low-flying aircraft and Remote Battlefield Sensor Systems (REMBASS) [4]. At this time, the solutions proposed were very expensive and dedicated only to military purposes. Even though researchers in the 1980s and early 1990s had in mind a global vision for a WSN, the processing workstations and communications technologies were not yet available to support their goals. The WSN research remained in the defense area until the end of 1990s and early 2000s when the first low-cost sensor nodes were developed [5].

From these years onward, the increasing miniaturization of micro-electromechanical systems and radio frequency devices paved the way to a new area of mobile networks and resulted in the emergence of many other potential applications, ranging from industrial to healthcare. One outcome of these advances was the Smart Dust project [5] developed at the University of California at Los Angeles, which focused on the development of highly miniature sensor nodes called "motes". This project demonstrated that a complete system composed of a micro-controller, wireless transceiver, sensors, and interface circuits can be integrated in a single tiny CMOS chip that has the size of a grain of sand. Another research effort was the PicoRadio project [6] initiated in 1999 to support the development of low-cost and low-energy ad-hoc wireless networks that can power themselves from the energy sources available in their operating environment, such as vibrational or solar energy.

Currently, WSNs are viewed as one of the most important technologies of the 21^{st} century due to the variety of possible applications [3]. A WSN may consist of many types of sensors that monitor physical information including temperature, light, movement, mechanical stress, etc. As shown in Figure 1.1, a huge number of applications are possible. We split these applications into four key domains: wearables, car and home automation, smart cities, and the industry. All these applications are encompassed by the concept of connected smart world [7, 8].

Wearable devices: Thanks to technological advances, intelligent sensor nodes may be integrated in devices such as wristwatch, wristband, glasses, shoes or an ordinary jacket [9, 10]. The recent developments in pervasive services

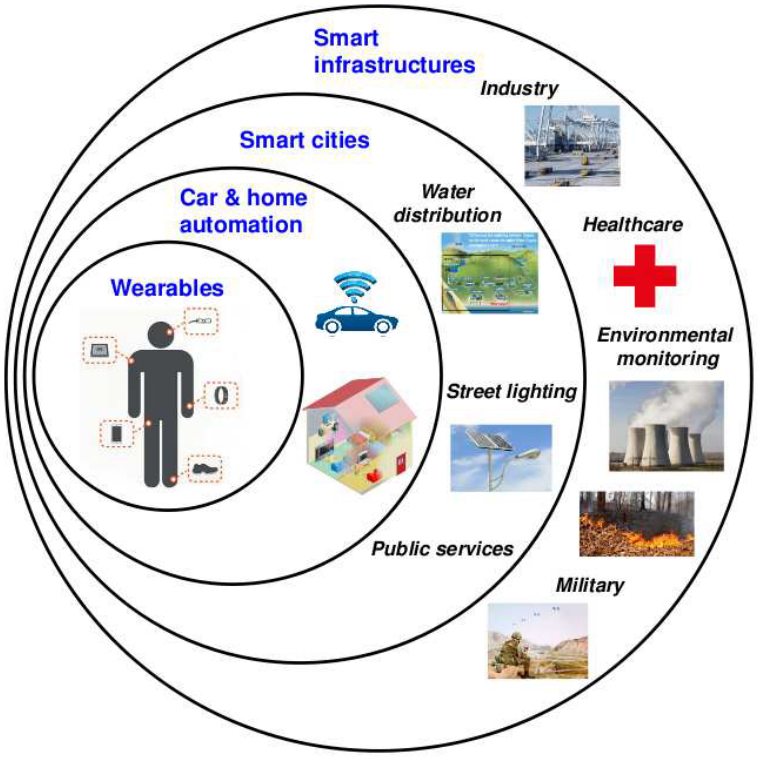


Figure 1.1: WSN application landscape

offer opportunities to provide an efficiency enhancement to human users. "Smart" devices (e.g., wristband, chest strap monitors, jacket) monitor the heart rate, compute calorie consumption and track (gestures) monitor [11]. Smart wristwatch and shoes are used by athletes to track sporting times, to monitor fitness tracking and distance [12]. Overlays navigation directions and information about points of interest directly on to the wearer's field of vision can be realized through glasses. Wearable devices can communicate with other WSNs deployed in a car or at home.

Car and building automation: A smart car is a car that is equipped with Internet access, and usually with a wireless local area network [13]. This allows the car to access Internet with other devices both inside and outside the vehicle. This car is also outfitted with special technologies that provide security and comfort to the driver, for example, automatic notification of crashes, notification of speeding and safety alerts. Other functionalities include entertainment, navigation, roadside assistance, voice commands, contextual help/offers, parking apps, engine controls and car diagnosis [14, 15].

Building automation represents another area for using WSNs. Sensor nodes

can transform a traditional building into an "intelligent" one that is aware of its occupants special needs and responds accordingly. The term "intelligent building" includes both the concept of individual networked home (Smart home) and positive energy building (Smart Building) [16]. Building automation may include control of lighting, Heating, Ventilation and Air Conditioning (HVAC), appliances (e.g. multimedia devices [17]), security lock of gates and doors and other systems, to provide improved comfort, energy efficiency [18, 19, 20] and security [21].

The popularity of building automation has greatly increased in recent years due to their affordability and simplicity of access through smart-phone and tablet connectivity. A home automation system connects various type of systems and devices in a house with each other. The techniques employed in home automation include those in building automation as well as the control of domestic activities, such as home entertainment systems, houseplant and yard watering, pet feeding, ambiance change for different events (such as dinners or parties), and the use of domestic robots [22]. Home automation for the elderly and disabled can increase the quality of life for these persons who might otherwise require caregivers or institutional care. The flexibility and ease of installation of WSNs can speed up the implementation of such buildings and contribute to the digital future of humanity.

Smart cities: WSN technology is envisioned to play a key role in instrumenting and interconnecting such smart urban environments. The objectives for a smart city are to assure a more efficient resource management and a better quality of life for the citizens by means of applications such as Smart Parking, Traffic Management, Noise and Pollution Monitoring, Smart Lightning (see Figure 1.2) [23, 24, 25].

Across the globe, some cities are currently revamping their critical infrastructure and services. At the same time, new cities appear that can incorporate the smart city vision from scratch [26]. Examples of such cities are Songdo IBD in South Korea, Masdar City in Abu Dhabi, Besançon in France, PlanIT Valley in Portugal, and SmartCity in Malta [27, 28].

Approximately 75% of the population of the European Union lives in urban areas [29]. The smart city concept as the next stage in urbanization has gained ground with policy makers, leading to huge investments. Resource management and environmental sustainability are part of the smart city concept. A smart city can be considered as an ecosystem, albeit one with highly technical components. This type of urban metabolism is an open and dynamic system that consumes, transforms, and releases materials and

Figure 1.2: Smart city concept

energy, develops and adapts to changes, and interacts with humans and other ecosystems.

Smart infrastructures: The sensor networks also have applications in other smart infrastructures, among which: the industry, healthcare, environmental monitoring, military.

Industry: The industrial applications are continuously looking for more efficient ways to improve production quality and reduce costs while minimizing waste of energy and materials. Thus, it is not surprising that several organizations, such as CISCO [30] work on the possibilities of using WSNs to monitor and enhance each step of a product including manufacture, delivery and consumption. Sensor nodes can offer real-time access to information about the equipment of plants, and prevent disruption of infrastructures [31]. Unlike wired networks, WSNs are excellent candidates for monitoring the entire life of a product, step by step from raw material provision used for its fabrication to its final assembling. The equipment to be monitored by WSNs can include assembly machines [32], product parts, logistics and transportation (containers or vehicles) [33], warehouses palettes localization [34], and end-user assets.

Healthcare: Health professionals are always looking for new techniques that provide the best care to patients. Thus, WSNs may change the interaction between the medical staff and their patients. The medical domain has produced many implementations of sensor networks ranging from equipment and patient health monitoring to wireless medicine injection and control [35]. Many health monitoring products implement WSNs such as SmartVest [36], AMON [37], Wealthy [38]. These systems transmit wirelessly and continuously physiological parameters about the health of the patient (e.g. they monitor vital signs, blood glucose, organ monitor, cancer detector

[39], etc.) so that healthcare professionals can diagnose and assure timely interventions.

Environmental monitoring: The growing concerns about ecology and the impact of humanity on climate changes and global warming is pushing scientists to use WSNs for environment protection. WSNs can ease the measurement of environmental data [40, 41] for a huge number of applications such as agriculture, meteorology, geology, zoology, etc. Today, WSNs are also used for the detection of forest fires [42] or floods and air pollution monitoring [43]. The advantage of using WSNs in such applications is mainly due to the need for acquiring large amounts of data in a region that would be costly to obtain using wired technologies. Based on this data, researchers and engineers build models that can describe the behavior of the environment and potentially predict disasters.

Military: The beginning of WSNs development focused on military applications [44]. One of the most important characteristics of WSNs that make them prominent for military applications is their ability to autonomously reorganize themselves [45, 46] to form a network capable of routing measurements to the commanders. From defense's perspective, WSNs represent an important technology mandatory for maintaining soldiers safe in the battlefield [47]. In addition, WSNs could provide useful information such as enemy troop movement, monitor the health of soldiers, coordinate resources and defense, direct gunfire origin, monitor critical equipments, etc.

Today, many WSN applications described in this section are widespread. Given the usefulness of such networks in our life, WSNs will continue their proliferation in every domain [48] that will raise the requirements to the WSN.

1.2 Requirements and challenges of WSNs

The WSNs are currently receiving significant attention due to their unlimited potential [49]. However, it is still very early in the lifespan of such systems and many research challenges exist. Sensor nodes are typically equipped with processing, sensing, power management and communication capabilities. In order to handle a large variety of applications, a sensor node has to meet the following requirements [50]:

Low Cost: usually, WSN are deployed using a large number of nodes, which demands the development of low cost sensor node platforms that address the specific WSN issues, such as those mentioned above. Low cost sensor node platforms allow the deployment of large scale WSN, which is the existing demand nowadays. The total cost contains the initial deployment

and maintenance costs are two key factors that will drive the adoption of WSN technologies. In data collection networks for research purposes, the primary goal will be to collect data from many locations (without exceeding their fixed budget). A reduction in per-node cost will result in the ability to purchase more nodes, deploy a collection network with higher density, and thus, collect more data.

Small Size: the node size also impacts the ease of network deployment. Smaller nodes can be placed in more locations and used in more scenarios. It will also reduce the power consumption and cost of sensor nodes.

Low Power: battery replacement can be difficult, expensive, or even impossible in harsh environments. To meet the multi-year application requirement, individual sensor nodes must be low-power. To develop the low-power wireless sensor applications, the several essential issues such as efficiently harvest, convert, and store energy as well as the use of the available energy in the most efficient way, without compromising performance (communication range, data rate, latency, and/or standards compliance) are crucial. The average power consumption of WSN nodes should be in micro amps. This ultra-low-power operation can only be achieved by combining both low-power hardware components and low duty-cycle operation techniques.

Environmental Impacts: while telecommunication technologies are still seen as "clean technologies", they can have some harmful impacts on the environment [51]. This situation becomes even more concerning given the fast advancement of wireless systems creating real electromagnetic turbulence of non-ionizing radiation on the habitat in which we live. For the reduction of the possible negative impacts to humans and the environment, we must define the rules and procedures for development, installation and operation of the wireless technologies.

The requirements and challenges of a WSN highly depend on the application domain. Nevertheless, many applications are concerned with energy consumption challenges. This point will be addressed in this thesis. The energy consumption of each sensor node separately and the WSN in its entirety can be considered. Since most sensor nodes are battery-powered, it is important to reduce or completely remove the battery replacement operation. Thus, to guarantee energy autonomy for sensor nodes, energy harvesting represents an efficient solution that may results in a self-powered node. Reducing the excessive power consumption of the nodes and/or of the number of active nodes by guaranteeing a given service requested by the application is of primary importance to assure a long lifespan to networks. The sensor nodes are unable to harvest energy from their environment or that harvest very little energy or in an intermittent manner [52].

1.3 Architecture and communication of WSNs

A sensor network usually contains (a lot of) small and possibly battery powered nodes scattered in a sensor field [53] as shown in Figure 1.3. Each of these sensor nodes has the capabilities to collect data and route it back to the gateway(s) (or sink) directly (single-hop network architecture) or *via* other sensor nodes playing the role of "repeaters" (multi-hop network architecture) [54], depending on the network topology [55]. The gateway is used for interfacing with networks that use different communication protocols. After, the gateway transmits data to the supervisor and, in some cases, to end users. It is responsible for interconnecting to a Wide Area Network such as the Internet [56, 57]. The supervisor monitor and control (task management) of the area where sensor nodes are spread.

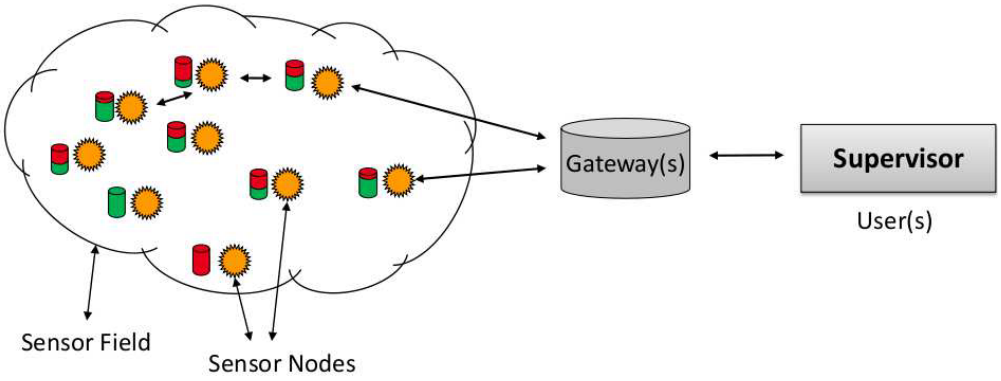


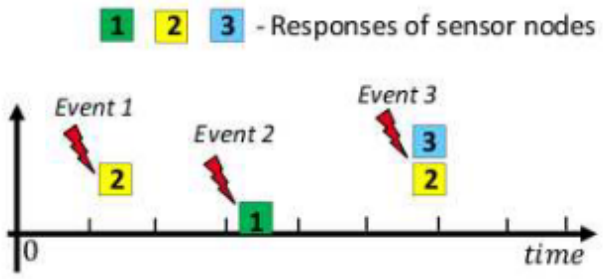
Figure 1.3: Architecture of the WSN

Three main communication schemes can be found for sensor networks [58, 59]:

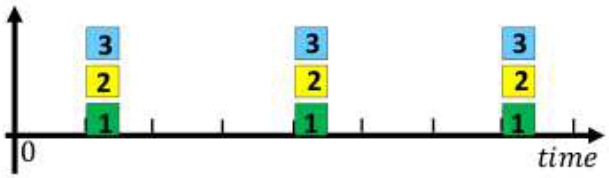
Event-driven communication: The sensor nodes send data to the supervisor when a specific event is detected (see Figure 1.4(a)). For example, for the application of waste and glass container management [60], when the sensors detect that the container level has crossed a threshold, this event is sent to the supervisor. This latter will inform the authorities to send a truck to unload the container.

Periodical communication: Sensor nodes periodically send information to the supervisor (see Figure 1.4(b)). A typical example is in agriculture area [61] where the nodes are deployed in a field to monitor the condition of the soil. They periodically send information about soil conditions to optimize water supply and nutrients.

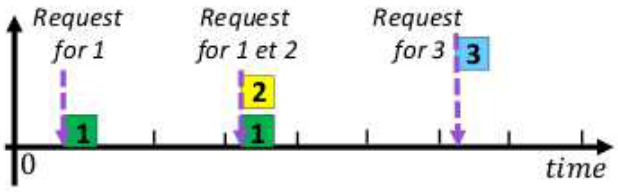
Communication on request: The supervisor sends a request to one or a set of nodes that must send their data (see Figure 1.4(c)). A typical example of this scheme is the remote meter reading of water meters [62]. In this



(a) Event-driven communication



(b) Periodical communication



(c) Communication on request

Figure 1.4: The communication architectures for WSNs

application, sensor nodes are deployed with water meters to monitor the usage of customers. The supervisor can be seen as a data collection device. Once in front of the building where he must collect information, the technician sends with his supervisor a request to collect information. When the request is received by the destination node(s), it(they) will respond by sending the information required by the supervisor.

These communication schemes can be present in the WSN separately or in combination. The choice of the communication scheme depends on the applications and on the used hardware capability.

1.4 Protocol stack for WSNs

A protocol stack model is commonly used in communication theory to describe the processes taking place in a networked communication (between a supervisor, gateway(s) and all sensor nodes) [63]. The protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes. The standard Open System Interconnection (OSI) model is composed of seven layers [64]. In the context of WSN, the stack is only composed of five layers [17]. It is also made up of cross planes/layers which include the Task Management, Connection Management and Power Management planes [53]. These are used to manage the network and make the sensors work together in the correct order to increase efficiency.

1.4.1 Protocol stack description

The protocol stack for WSNs comprises the Application, Transport, Network, Data Link and Physical layers [17, 65, 66]. Each layer has a set of functions that it must perform in order for data packets to travel from a source to a destination on the network. Below is a brief description of each layer in the WSN model as shown in Figures 1.5.

WSN layers	Function	Data type
Application layer	- User Application Services, allows access to network services that support Applications - Handles network access, flow control and error recovery	User data
Transport layer	- Additional connection below the Application layer - Manages the flow control of data between parties across the network, - Responsible for end-to-end error recovery and flow control; Ensures complete data transfer	Datagram's /Segments
Network layer	- Creates logical paths for transmitting data from node to node - Translates logical network address and names to their physical address (e.g. computer name to MAC address) - Logical Addressing; Routing; Datagram Encapsulation; Fragmentation and Reassembly; Error Handling and Diagnostics	Datagram's /Packets
Data Link layer	- Handles data frames between the Network and Physical layers - Two sub layers: MAC, LLC - The receiving end packages raw data from the Physical layer into data frames for delivery to the Network layer - Logical Link Control; Media Access Control; Data Framing; Addressing Error Detection and Handling; Defining Requirements of Physical Layer	Frames
Physical layer	- Transmits raw bit stream over wireless connection - Defines wireless connections, cards, and physical aspects - Encoding and Signaling; Wireless Date Transmission; Hardware Specifications; Topology and Design	Bits

Figure 1.5: The protocol stack of WSNs

Application layer: This is the layer at which communication partners are identified. User authentication and privacy are considered, and any constraint on data syntax is identified. This layer is not the application itself, although some applications may perform application layer functions. It represents the services that directly support applications such as software for file transfers, database access, etc.

Transport layer: Standards at this layer ensure that all packets have arrived. This layer also isolates the upper layer – which handle user and application requirements from the details that are required to manage the end-to-end connection.

Network (or Routing) layer: This layer handles the routing of the data, addresses messages and translates logical addresses and names into physical addresses. It also determines the route from the source to the destination computer and manages traffic problems (flow control), such as switching, routing, and controlling the congestion of data packets.

Data Link layer: It is the layer that makes the link between software and hardware. Data Link layer are divided into two sub-layers: I) the upper sublayer – Logical Link Control (LLC) layer and II) the lower layer – Media Access Control (MAC) layer. The LLC layer acts as an interface between the MAC layer and the Network layer. Its main responsibility is to provide services to the Network layer protocols and particularly to control frame synchronization, flow control and error check. The MAC layer determines the media access controlling on hardware devices and provides data analyzing based on physical signaling requirements of the medium. Since the medium in WSNs is made of wireless technologies, enhancing the radio design and effective MAC protocols that will decrease the power consumption in a radio is a main channeling issue in protocol design in MAC sublayer. Functionalities of MAC protocol are changed depending on the Network layer requirements, hardware specifications and network topologies [67].

Physical layer: It defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems. Such characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other, similar, attributes are defined by the Physical layer specifications.

The configuration of the protocol stack depends on the application and devices in a WSN. The protocols used at each layer strongly influence the communication part and, as a consequence, the energy consumption of whole WSN.

1.4.2 Plane description

We can consider that the five layers are split into three planes [68], namely, Task Management, Connection Management and Power Management planes, as shown in Figure 1.6.

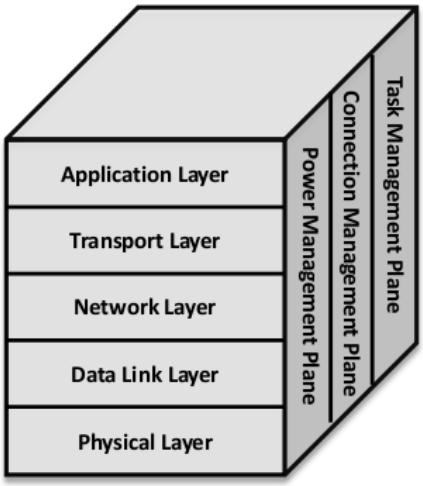


Figure 1.6: Protocol stack for a WSN

Power Management plane: The main goal for the power management plan is to take charge of managing the power supplies for all the different parts of the sensor node such as sensing data, processing, broadcasting and responses that depend on a resourceful power management scheme at every phase of the protocol layers [69]. For instance, at the MAC layer, to avoid getting duplicated messages, a sensor node may turn off it receiver after receiving a message from one of its neighbors. Also, a sensor node broadcasts to its neighbors that it is low in power and can not take part in routing messages. The remaining power is reserved for sensing and detecting tasks. At the network layer, a sensor node may select a neighbor node with the most residual energy as its next hop to the gateway [70].

The power management can be used to maximize the node lifespan or/and WSN (or network) lifespan. The WSN lifespan is the amount of time a WSN would be fully operative. This means the amount of time till a certain percentage of sensor nodes fails or survives in the network [71].

Connection Management plane: It is responsible for the configuration or reconfiguration of the sensor nodes in an attempt to establish or maintain the network connectivity.

This plane is responsible for realizing the communication between the nodes and the supervisor. The handling of configuration and re-configuration of the sensor nodes are through the connection management plan which ensure continuous connectivity and node maintenance of the network whenever changes to the topology due to breakdown of nodes, movement occurrence and node addition.

In the connection management plane, a Quality of Service (QoS) is treated. In sensor networks, the "traditional" QoS reflects the capacity of the communication system to guarantee the performance required by the application (speed, time packet transmission from end-to-end, packet loss rate) [72]. The guarantees offered in terms of QoS are therefore dependent on the intended application because of the different requirements and specific characteristics of each type of data traffic. The metric used to ensure QoS are the latency, the packet loss rate, the bandwidth and the data rate. Note that it is not the QoS of a particular node that is relevant, but due to the redundancy of the sensor nodes, rather QoS of the set of nodes assigned to the same task.

In the literature, there are mainly two different approaches for supporting the QoS in sensor networks. The first approach is related to the MAC layer that operates its access control mechanisms, channel reservation and scheduling data transmissions to provide the desired QoS [73]. The second approach exploits the temporal variability of the radio channel at the physical layer to improve the QoS. For example, the approach taken in [74] allows for an allocation of the energy resources to ensure a bounded time depending on the transmission conditions. While in [75], the modulation is adjusted to increase the flow to determine, according to QoS constraints, the best compromise time/energy consumed. While most sensor networks research work (e.g. [76, 77]) focuses on the energetic efficiency, it therefore seems essential to find the right compromise between QoS and energy efficiency at the risk of reducing the reliability of transmissions and thus actually reduce the energy efficiency by increasing retransmissions of packets.

Task Management plane: It is responsible for the distribution of tasks among sensor nodes to increase the network lifetime and improve the energy efficiency. The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time. As a result, some sensor nodes may perform the task more often than others, depending on their power level.

The Task Management plane is more related to the sensor nodes as they

have to collaborate very efficiently with each other in order to perform a certain task, working sequentially or concurrently. Allocation of tasks or scheduling of the sensing between the sensor nodes is the main duty of the task management plane. This procedure ensures energy efficiency improvement. Thereby the network lifespan is increased. Thus management techniques can be applied to perform sharing of schedules for several sensors nodes [78].

The communication protocols at different layer can be oriented towards the energy, link quality or/and task management at the same times. In addition, each sensor node and other devices of a WSN can have its own communication protocols. However in the end, they must work in harmony, i.e. without conflicts between the different layers and devices.

1.5 Traditional architecture and power consumption of a sensor node

The WSN lifespan strongly depends on the lifetime of the single nodes that constitute the WSN. The lifetime of a sensor node begins when it first boots and ends when it is no longer able to communicate or perform its other basic tasks [79]. While the sensor network community develops algorithms and methods for power management of WSNs at the different layers, others from the hardware development community proposes solutions to minimize the power consumption of each unit that forms the sensor node.

1.5.1 Traditional architecture description

A sensor node consists of several units which correspond to a particular role, i.e. acquisition, processing, or communication. It also includes a Power Supply unit [53]. The functional units of a node are shown in Figure 1.7.

As can be seen, the node is split into four main units:

Sensing unit: It consists of one or several sensors and a read/write interface (e.g., an Analog-to-Digital Converter (ADC)). Sensors collect information from the physical world and convert it to electric signals. The analogue signals produced by sensors are converted to digital signals by ADC. There exists a plethora of sensors that measure environmental parameters such as light, temperature, humidity, sound, etc.

The sensors are classified into three categories: (I) passive, omni-directional sensors; (II) passive, narrow-beam sensors; and (III) active sensors. The passive sensors sense data without actually manipulating the environment by active probing. They are self-powered; that is, energy is needed only to

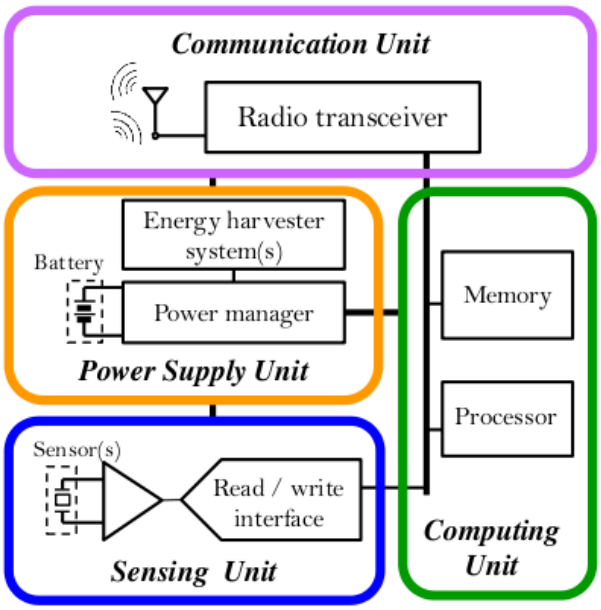


Figure 1.7: Functional units of a wireless sensor node (adapted from [80])

amplify their analog signal. The active sensors actively probe the environment, e.g. a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omni-directional sensors have no notion of direction involved in their measurements. Host of the theoretical work on WSNs deals with passive, omni-directional sensors. Each sensor node has a certain area of coverage for which it can reliably and accurately report the particular quantity of interest that it is observing.

Computing unit: This unit ensures intelligence to the node *via* a microprocessor or Micro-Controller Unit (MCU). The computing unit performs tasks, processes data and controls the functionality of other components in the sensor node. It can be used also for the management of the power supply unit. In the computing unit, the state of each unit (such as *On, Idle, Sleep, Deep sleep, Off*) and the functioning mode (such as *Active, Standby*) of the whole node are controlled. Also, the wake-up of the sensor node is launched.

Memory can be either volatile and non-volatile memory. Volatile memory is memory that loses its contents when the computer or hardware device loses power. Computer Random Access Memory (RAM) is an example of a volatile memory and is why if our computer freezes or reboots when working on a program, you lose anything that hasn't been saved. Non-volatile memory, sometimes abbreviated as NVRAM, is memory that keeps

its contents even if the power is lost. EPROM is an example of a non-volatile memory.

Communication unit: The communication unit comprises mainly the radio transceiver (RF transceiver) with the amplifiers and associated electronics. The RF transceiver enables the wireless communication with the gateway and also between nodes. It implements all the necessary functionalities to convert bits to radio waves and *vice-versa*. In the radio transceivers, the different wireless technologies (as shown in Table 1.1) based on standards and operating in different frequency bands are embedded. The communication unit can cause the node wake-up [81].

Power supply unit This unit supplies the energy to the node. It typically consists of a battery with a DC-DC converter. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. They are also classified according to the electrochemical material used for the electrodes such as NiCd (nickel-cadmium), NiZn (nickel-zinc), NiMH (nickel-metal hydride), and lithium-ion. Some sensor nodes are able to increase their energy from harvesting systems. Indeed, there are various sources of ambient energy available for harvesting. These energy sources can be categorized into five types: mechanical, thermal, solar, wind, and radio-frequency [52]. Environmental energy harvesting represents a viable option to increase the autonomy of a node when the battery is hard or impossible to be recharged or replaced.

Table 1.1: Wireless technologies [82]

Technology	**Standard**	**Industrial organization**	**RF Frequency**
Zigbee	IEEE 802.15.4	Zigbee Alliance	2.4 GHz, 868 / 915 MHz
Blutooth	IEEE 802.15.1	Bluetooth SIG	2.4 GHz (79 channels - 1.6 MHz)
UWB	IEEE 802.15.3a	UWB Forum and WiMedia^{TM} Alliance	3.1-10.6 GHz
Wi-Fi	IEEE 802.11b IEEE 802.11g IEEE 802.11a IEEE 802.11n	Wi-Fi Alliance	2.4 GHz - 5.8 GHz
Proprietary standards	Proprietary	N/A	433/868/900 MHz, 2.4 GHz

1.5.2 Power consumption of a sensor node

The energy consumption of a sensor node is typically distributed amongst Communication, Computing and Sensing units as show in Figure 1.8 [83, 84, 79, 85, 86, 87].

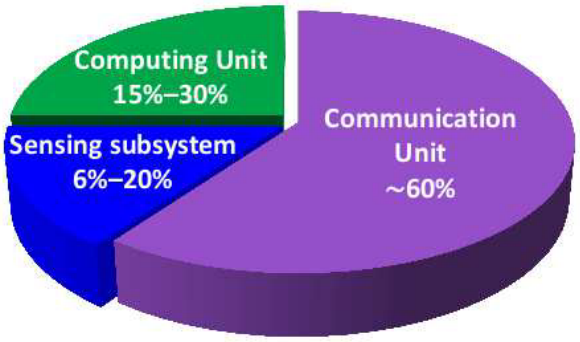


Figure 1.8: Power consumption distribution for a wireless sensor node

Power consumption of the Sensing unit: The power consumed in the Sensing unit is used in sensor sampling, which includes the wake-up and stabilization time associated with the sensor and the data acquisition time. At all other times, the sensors are completely off and they consume no power. The power consumption of the ADC is typically proportional to the amount of samples acquired and the sampling node used [83], see the communication schemes in Section 1.3. With a low rate ADC and passive sensors, the Sensing unit will be one of the least energy consumers. However, if higher rate ADCs and/or energy hungry sensors are used for a particular application, the power consumption of the Sensing unit can drastically increase up to the consumption of the Communications unit [88]. That is why low-energy ADCs (e.g. [89, 90]) and low-energy sensors (e.g. [91, 92]) for the different applications are mainly used.

Power consumption of the Computing unit: The different processor hardware parameters influence the node overall power consumption and thus also the node lifetime. This is why a low power circuit design is important [93]. Today's MCUs, e.g., TI's MSP430 family or Atmel's XMEGA 256 controllers, offer unprecedented economic operation/sleep states to minimize the power consumption [94, 95]. The influence of the supply voltage, clock frequency, emplacement of the program in the MCU memory for MCU and others have been studied in different works (e.g. [96, 97]).

Power consumption of the Communication unit: The Communication unit – transmit, receive, and listen – dominates the power budget with $\sim 60\%$ of

the total available energy. The energy consumption of this unit strongly depend of the communication protocols (see Section 1.4) used at the different layer of the network. It greatly influences the lifetime of the sensor node. Its power consumption is evaluated by parameters such as voltage supply, transmitting current, receiving current, current at power-down mode, etc. A large number of works (e.g. [98, 80, 99, 100, 101]) try to reduce the power consumption of the Communication unit.

Energy in the Power Supply unit Using energy stored in the batteries or energy harvested from the outside world are the two options for the Power supply unit. The energy storage may be achieved with the use of batteries or alternative devices such as fuel cells or miniaturized heat engines, whereas energy-scavenging opportunities [102] are provided by solar power, vibrations, acoustic noise, and piezoelectric effects [103]. Each battery type has characteristics that make it appropriate, or inappropriate for a sensor node in a given application. Knowing the specific characteristics of each cell chemistry in terms of voltage, cycles, load current, energy density, charge time, and discharge rates is the first step in selecting a cell for a sensor node [104]. The discussions in the works [105, 104, 106] give an overview of the characteristics, strengths, and weaknesses of the different battery types.

Two power saving policies can be used in the Power supply unit with the influence on other units: Dynamic Power Management (DPM) and Dynamic Voltage Scaling (DVS) [107]. DPM conserves power by shutting down parts of the sensor node which are not currently used or active. By varying the supply voltage along with the clock frequency of the node, it is possible to obtain quadratic reduction in the power consumption. This DVS scheme is applied depending on the active node workload and tasks to be performed.

As we could see, each unit influences in more or less degree on the energy consumption of a node. Now, there are solutions to make the different parts of the node few consuming. However, research does not stop on these achievements, it continues to move forward with new proposals of more efficient hardware. Table 1.2 summarizes the sensor nodes that are more used in the WSN applications with its characteristics on the computing, communication parts, and also its current consumption in the different states. This example of the node platforms confirms the large consumption of communication and shows the need to reduce the number of connections between the nodes and the supervisor.

1.6 Conclusion

Wireless sensor networks hold great promise in applications ranging from environmental and structural monitoring to industrial control and human health

Table 1.2: Available wireless sensor nodes

Name	**MCU**	**RAM** $[kB]$	**Communication**	**Consumption**			
				Slep $[\mu A]$	Normal $[mA]$	RX, $[mA]$	TX, $[mA]$
Arduino BT [108]	Atmel ATmega328, 8-bit, 20MHz	32	Bluetooth	N/A	N/A	N/A	N/A
eZ430-RF2500 [109]	MSP430F2274, 16-bit, 16MHz	1	Proprietary	N/A	N/A	13.3-18.8	11.1-21.2
iMote2 [110]	Intel PXA271 Xscale, 13-416MHz	32	802.15.4	390	31	44-66	44-66
Mica2 [111]	Atmel ATmega128L, 8-bit, 16MHz	128	802.15.4	15	8	9.6	16.5
Flyport Wi-Fi 802.11g [112]	PIC24F, 16-bit, 32MHz	16	Wi-Fi	1.44 mA	28.21	162.7	282.5
SunSpot [113]	Atmel ARM920T, 16-bit, 180 MHz	512	802.15.4	36	35	18.8-19.7	17.4
Telos/Tmote [114]	TI MSP430F149, 16-bit, 8MHz	10	802.15.4	5.1-21	1.8-2.4	18.8-23	17.4-21
WiSMote Dev [115]	TI MSP430F5437, 16-bit, 8MHz	16	802.15.4	2	2.2	18.5	25.8-33.6
WiSMote mini [116]	Atmel ATmega128RFA2, 8-bit, 16MHz	16	802.15.4	20 nA	N/A	12.5-16.6	18.6

monitoring. However, almost all WSNs available on the market still operate on batteries. Such a situation presents a substantial roadblock to the proliferation of WSNs due to the need of long lifespan for the network.

In the literature and in the industry, a large number of protocols based on different approaches and algorithms are proposed at different layers to increase the network lifespan. There are also diverse hardware solutions in each sensor node to reduce its energy consumption. Therefore, maximizing the WSN lifespan comes down to reducing the energy consumption not only of nodes but of the different parts of the network.

To combine these achievements in different network layers and at node level, a global vision on the WSN and its consumption is necessary. Thanks to this global vision, the communications protocols can be chosen depending on the hardware capability of a sensor node. For instance, when a node has different state as sleep, idle, TX, RX etc., we need to have a protocol which can handle this state in order to ensure a given service (e.g. send a measurement) and at the same time to minimize the energy consumption. Therefore, the energy management of the whole network at application level must be searched during the development of autonomous WSNs to find a trade-off between performance and energy consumption. It means that some information from nodes and layers

such as its consumption, remaining energy, work to do, is sent to a centralized controller. This information allows to manage the nodes and the whole network depending on the control objectives which are highly interconnected with the service to assure and the energy consumption to minimize.

In the next Chapter, we present a strategy to minimize the WSN energy consumption at application level by using Model Predictive Control approach.

Chapter 2

Energy Management using a Predictive Control Approach

The lifespan is probably the most important metric in assessing the performance of a WSN. Indeed, in a constrained environment, any limited resource has to be taken into account. The lifespan of the network, related to the energy consumption, has become the key characteristic for evaluating sensor networks in an application-specific way.

This chapter aims at proposing a novel control strategy at application level to minimize the energy consumption based on a Model Predictive Control (MPC) approach. This choice has been motivated by:

- the model of the energy consumption for the set of nodes;
- the global objectives that are to minimize the energy consumption of the set of nodes and to assure a given service;
- the set of constraints that must be fulfilled.

We first summarize the basic concepts of the MPC approach. In order to manage the energy savings in the WSN, the modeling of the WSN energy consumption is presented. Then, we propose a control strategy at the application level that will be evaluated through a realistic benchmark in simulation. The results drive to the first conclusions and validations. The research activities presented in this chapter have been published in [117, 118].

2.1 Introduction to Model Predictive Control

2.1.1 Historical context

Model Predictive Control (MPC), also referred to as Receding Horizon Control or Moving Horizon Optimal Control, has been discovered and re-invented several times. The ideas of Receding Horizon Control and Model Predictive Control can be traced back to the 1960s [119]. Receding Horizon approaches were used in the 1960s and 70s to define computational methods for optimal control problems that have no closed-form solution [120]. MPC reappeared in the entirely different context of industrial process control in the 1980s to exploit continual improvements of computational resources so as to improve controlled system performances. More recently, the approach has been used as a general technique for deriving stabilizing controllers for constrained systems [121]. At the same time, the availability of faster computers and improvements in computational efficiency of predictive controllers (including non-linear and robust MPC schemes) have extended its range of applications to include fast sampling systems [122].

More than 30 years after MPC appeared in the industry [123, 124] as an effective means to deal with multi-variables constrained control problems, a large theoretical basis for this technique exists [125, 119]. It is based on predicting the system trajectories over a receding horizon, while calculating an optimal control policy, possibly with respect to a set of constraints [121]. It can be applied to linear Single-Input Single-Output (SISO) and Multi-Input Multi-Output (MIMO) systems [126, 120], non-linear [127, 128] and hybrid [129] systems. The systems can present continuous (and sampled) states (real-set variables), discrete and state-machine states (integer-set variables) and logic rules (binary-set variables) [130]. Many practical problems such as control objective prioritization and symptom-aided diagnosis can be integrated systematically and effectively into the MPC framework by expanding the problem formulation to include the real, integer or/and binary variables, leading to Linear Programming (LP), Quadratic Programming (QP), Mixed Integer Programming (MIP) or Mixed Integer Quadratic Programming (MIQP) problems. Efficient optimization techniques for solving these problems are available [119].

2.1.2 Basic concepts of MPC

The conceptual structure of MPC is depicted in Figure 2.1. The name MPC stems from the idea of employing an explicit model of the system to be controlled which is used to *predict* the future output behavior. This prediction capability allows solving *optimal* control problems on-line, where the tracking error, namely the difference between the predicted output and the desired reference, is minimized over a future horizon, possibly subject to constraints on the manipulated inputs and outputs. When the model is linear, then the optimization problem is

quadratic if the performance index is expressed through the l_2-norm, or linear if the performance index expressed through the l_1/l_∞-norm [131]. The result of the optimization problem is applied according to a *receding horizon* philosophy: at time k only the first input of the optimal command sequence is actually applied to the system. The remaining optimal inputs are discarded, and a new optimal control problem is solved at time $k+1$. This idea is illustrated in Figure 2.2.

As new measurements are collected from the system at time k, the receding horizon mechanism provides the controller with the desired feedback characteristics. A MPC approach contains the basic components of prediction, optimization and receding horizon implementation [120]. A summary of each of these ingredients is given below.

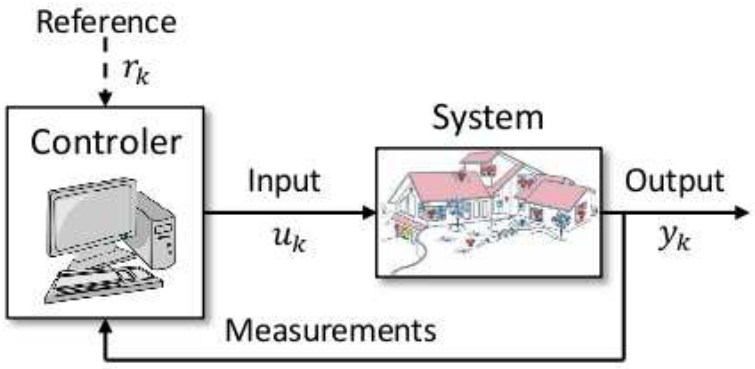


Figure 2.1: Basic structure of Model Predictive Control

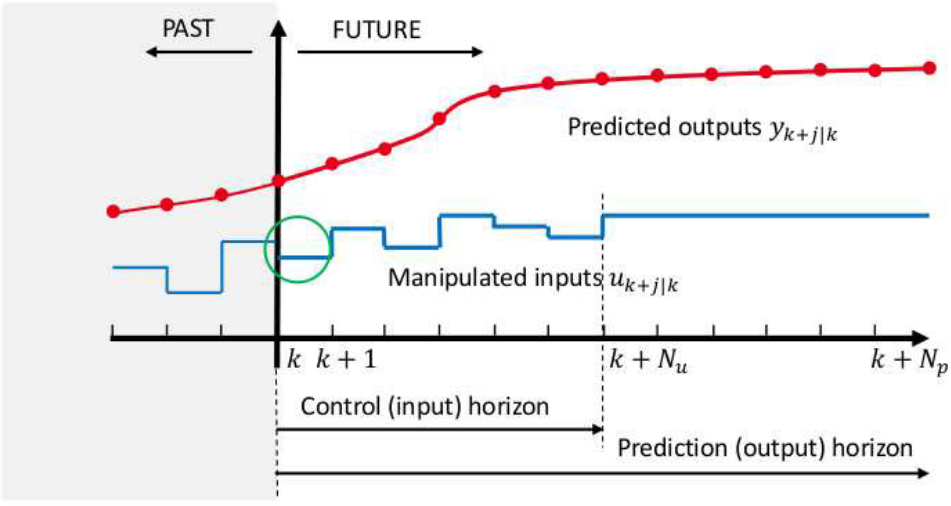


Figure 2.2: Receding horizon strategy: only the first computed control (input of the system) u_k is applied to the system to be controlled

Prediction

The future response of the controlled system is predicted using a dynamic model. We consider the case of discrete-time linear systems with state-space representation

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k, \tag{2.1}$$

where $\mathbf{x}_k$, $\mathbf{u}_k$ denote the state and control vectors at the k^{th} sampling instant, respectively. Note that, in the present study, the output vector (measurable) is $\mathbf{y}_k = \mathbf{x}_k$. Given a predicted input sequence, the corresponding sequence of state predictions is generated by simulating the model forward over the prediction horizon, i.e. N_p for the output prediction and N_u for the input prediction sampling intervals. For convenience, these predicted sequences are often stacked into vectors $\mathbf{x}$, $\mathbf{u}$ defined by:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{k+1|k} \\ \mathbf{x}_{k+2|k} \\ \vdots \\ \mathbf{x}_{k+N_p|k} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_{k|k} \\ \mathbf{u}_{k+1|k} \\ \vdots \\ \mathbf{u}_{k+N_u-1|k} \end{bmatrix}. \tag{2.2}$$

$\mathbf{x}_{k+j|k}$ and $\mathbf{u}_{k+j|k}$ denote the state and input vectors at time $k+j$ that are predicted at time k. As shown in Figure 2.2, N_p denotes the length of the *prediction horizon* or output horizon, and N_u denotes the length of the *control horizon* or input horizon with $N_u \leqslant N_p$. When $N_p = \infty$, we deal with the classical infinite horizon problem. Similarly, when N_p is finite, we refer as a finite horizon problem. $\mathbf{x}_{k+i|k}$ evolves according to the prediction model:

$$\mathbf{x}_{k+j+1|k} = A\mathbf{x}_{k+j|k} + B\mathbf{u}_{k+j|k},\ j = 0, 1, ... \tag{2.3}$$

with the initial condition defined by $\mathbf{x}_{k|k} = \mathbf{x}_k$ (which is here suposed to be measured).

Optimization

The predictive control feedback law is computed by minimizing a performance cost, which is defined in terms of the predicted sequences $\mathbf{u}$, $\mathbf{x}$. Usually, the quadratic cost J_k is chosen:

$$J_k = \sum_{j=0}^{N_p} \mathbf{x}_{k+j|k}^T Q \mathbf{x}_{k+j|k} + \sum_{j=0}^{N_u-1} \mathbf{u}_{k+j|k}^T R \mathbf{u}_{k+j|k}, \tag{2.4}$$

where Q and R are symmetric positive semi-definite and symmetric positive definite matrices respectively. We consider that $\mathbf{u}_{k+j|k} = \mathbf{u}_{k+N_u-1|k}$, where $j = N_u, ..., N_p - 1$. Clearly, J_k is a function of $\mathbf{u}$, and the optimal input sequence for the problem of minimizing J_k is denoted $\mathbf{u}^*$:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} J_k. \tag{2.5}$$

If the system is subject to control and state constraints, then they can be naturally taken into account in the optimization problem.

Receding horizon implementation

As stated above, only the first element of the optimal predicted input sequence $\mathbf{u}^*$ is applied to the system:

$$\mathbf{u}_k = \mathbf{u}_{k|k}. \tag{2.6}$$

The state predictions $\mathbf{x}$ and hence the optimal control sequence $\mathbf{u}^*$ depend on the current state $\mathbf{x}_k$. This procedure introduces feedback into the MPC law, thus providing a degree of robustness to modeling errors and uncertainty.

A second feature of the receding horizon approach is that, by continually shifting the horizon over which future inputs are optimized, it attempts to compensate for the fact that this horizon is finite. Provided the cost and constraints are properly designed, a receding horizon strategy can ensure that the performance of the closed-loop system is at least as good as the one of the optimal prediction [131].

2.1.3 MPC approach for energy management of a WSN

The MPC techniques provide a methodology to handle constraints in a systematic way during the design and implementation of the controller. Moreover, in its most general form, MPC is not restricted in terms of the model, objective function and/or constraints [119]. For these reasons, the MPC approach seems a good candidate for energy management of a WSN at application level.

2.2 WSN system modeling

The objective of the present work is to control (minimize) the energy consumption of a set of nodes while ensuring that a pre-defined service offered by these nodes is fulfilled. Thus, consider a WSN that contains $n \in \mathbb{N}^*$ sensor nodes S_i, $i = 1, \ldots, n$, each node being powered by a battery. Here, we consider one battery, but in general, the node may have several batteries. The nodes may also be equipped with harvesting systems. All nodes are functionally equivalent: they are interchangeable but their hardware can differ, e.g. batteries, processors may be different. The communication can be multi-hop or single-hop clustered (see Section 1.3). Moreover, each node has to send data through a gateway to the supervisor.

The nodes exhibit different functioning modes M_h, $h = 1, \ldots, m$, $m \in \mathbb{N}^*$, which are related to the state (on, sleep, off, etc.) of each node unit (see Section 1.5), characterized by a known energy consumption over a given period of time. Typically, the functioning modes are *Active*, *Standby*. Note that these modes can

be split in sub-modes with their associated energy consumption. For instance, the *Active* mode can be split in *Transition*, *Ready*, *Monitor*, *Observe*; *Standby* – *Sleep*, *Deep sleep* [132, 133]. The supervisor chooses the functioning mode of each node thanks to an energy control strategy presented hereafter.

In order to control energy savings in the WSN, the remaining energy in the nodes is modeled with a discrete-time Linear Time-Invariant (LTI) state-space model:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + E\mathbf{w}_k, \tag{2.7}$$

where $\mathbf{x}_k \in \mathbb{R}^n_{>0}$ is the remaining energy in the batteries of the nodes at each sampling time k. The state matrix is $A = \mathbf{I}_n \in \mathbb{R}^{n\times n}$. $B\mathbf{u}_k$ represents the energy that will be consumed during the time interval $[kT_c, (k+1)T_c]$. T_c is the control period. $E\mathbf{w}_k$ in (2.7) corresponds to the energy gained from the harvesting system.

Normally, in real-life experiments, T_c may be variable, depending on the environmental and node conditions. Thus, it may be adjusted thanks to the occurrence of events in the system, for instance the disappearance of a node that is supposed to be in the *Active* mode at time $kT_c < T_f < (k+1)T_c$. As a consequence, the control law should be run before the end of the initially planned control period T_c is elapsed to ensure that the service the WSN has to provide is fulfilled.

The initial battery capacity (i.e. at $k = 0$) is denoted $\mathbf{x}_0$. Moreover, for each node S_i, the remaining energy is constrained:

$$0 < X_i^{min} \leqslant x_i \leqslant X_i^{max}, \tag{2.8}$$

where $X_i^{min} = \gamma_i X_i^{max}$ corresponds to the minimal energy level in the battery of the sensor node S_i. The value γ_i is provided by the battery manufacturer to avoid damages to the battery [134]. X_i^{max} is the nominal energy capacity, supposed constant. In the case the sensor node is plugged into the main supply, it has no battery and its energy capacity is unlimited. However, it can still exhibit different functioning modes with their associated energy consumption. Note that the remaining energy in the node battery is supposed to be delivered by each node at sampling time k, i.e. we have:

$$\mathbf{y}_k = \mathbf{x}_k. \tag{2.9}$$

$\mathbf{u}_k = \begin{bmatrix} u_1^T & \cdots & u_i^T & \cdots & u_n^T \end{bmatrix}^T \in \{0,1\}^{nm}$ is the control input. Each sub-vector $u_i = \begin{bmatrix} u_{i1} & \cdots & u_{ih} & \cdots & u_{im} \end{bmatrix}^T$ is related to the functioning mode of S_i, where $u_{ih} \in \{0,1\}$. As each node S_i has a unique functioning mode at each time instant, a set of constraints must be defined:

$$\forall i = 1, \ldots, n : \sum_{h=1}^{m} u_{ih} = 1, u_{ih} \in \{0,1\}. \tag{2.10}$$

It is also possible to impose additional constraints on $\mathbf{u}_k$. For instance, one can require that a node with a specific geographic location is active/inactive all the time (or at given periods of time). We can also force a more accurate sensor node to be active during the night period, if we consider that we need only one precise active sensor during the night period to perform the application that turns on top of the sensor network.

The control matrix is $B = \text{diag}(-B_1, \ldots, -B_n) \in \mathbb{R}^{n \times nm}$. Each component b_{ih} of $B_i \in \mathbb{R}^m$ represents the amount of energy consumed by S_i working in mode M_h (see Table 2.1). The component b_{ih} can contain:

- an average value of the energy consumption over the time period T_c;
- the exact amount of consumed energy, which is measured and possibly updated (leading to a time variable matrix B_k) by the supervisor (or by each sensor node itself) in real-time.

The choice of the numerical value of b_{ih} depends on the application and on the hardware node at hand. Note that a switch of S_i from mode M_h to mode M_l imposed by the supervisor has an extra energy cost that is supposed to be integrated in b_{il}.

Table 2.1: Energy consumption of each sensor node in different functioning modes

Sensor node	**Mode** M_1	...	**Mode** M_h	...	**Mode** M_m	
S_1	$[b_{11}$	...	b_{1h}	...	$b_{1m}]$	$= B_1$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
S_i	$[b_{i1}$	...	b_{ih}	...	$b_{im}]$	$= B_i$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
S_n	$[b_{n1}$	...	b_{nh}	...	$b_{nm}]$	$= B_n$

$\mathbf{w}_k = \begin{bmatrix} w_1 & \cdots & w_i & \cdots & w_n \end{bmatrix}^T \in \{0,1\}^n$ can be seen as a disturbance input that cannot be controlled but may be predicted in some situations. Basically, w_i corresponds to the ability for node S_i to harvest energy over the time period $[kT_c, (k+1)T_c]$. 1 (resp. 0) is associated to the *Active* (resp. *Inactive*) state of the harvesting system. $E \in \mathbb{R}^{n \times n}$ is the so-called disturbance matrix:

$$E = \text{diag}(E_1, \ldots, E_i, \ldots, E_n), \tag{2.11}$$

where E_i corresponds to the amount of energy harvested by S_i during the period T_c. Note that matrix E is in essence a time-variant matrix in real-life conditions.

Consider that the lifespan at time k of the *reachable* sensor node S_i is defined by:

$$L_i = \frac{x_i + E_i w_i - X_i^{min}}{B_i u_i}, \tag{2.12}$$

if there is no changes at time k in the mode of the node. A *reachable* sensor node S_i is a node that is "seen" by the supervisor: this latter received information regarding the remaining energy y_i of S_i at time k. The notation that appears in (2.12) are illustrated on Figure 2.3.

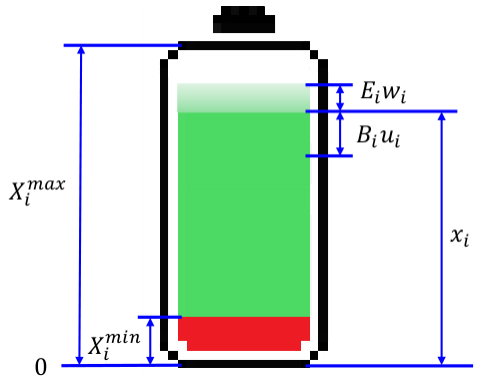


Figure 2.3: Definitions related to the battery energy at time t, where X_i^{max} is the maximum energy in the battery; X_i^{min} is the minimum energy in the battery; x_i is the remaining energy in the battery; $E_i w_i$ is the energy possibly harvested during $[kT_c, (k+1)T_c]$; and $B_i u_i$ is the energy consumed during $[kT_c, (k+1)T_c]$

2.3 Control objectives

In the present work, a dynamic energy saving control policy is proposed at application level in order to increase the WSN lifespan while meeting a given *service* requested by the application built on top of the WSN. This application makes use of the measurements provided by the nodes.

This service is expressed for instance as a given (minimum) number of measurements that must be provided by the WSN over a defined time period. Hereafter, this service is called *mission*. We define the *WSN lifespan* as the time interval until which the mission is fulfilled.

Typically, the mission has to be guaranteed over a geographically limited area, where each node can reverse its role with another one without decreasing the performance of the whole network. In order to fulfill the mission while decreasing the energy consumption, a subset of $d_h \in \mathbb{N}^*$ nodes is assigned to a given mode M_h. Thus, another set of constraints is defined:

$$\sum_{i=1}^{n} u_{ih} = d_h. \tag{2.13}$$

Basically, the idea is to reduce the number of active nodes if the measurements provided by all the nodes are not all mandatory. This information can be seen as an external reference.

As indicated above, the control vector $\mathbf{u}_k$ takes its values in $\{0,1\}^{nm}$. It fixes the functioning mode of each node under constraints (2.10). The constraints in (2.13) are used to define the mission. Consider a network with n nodes that deliver strongly correlated measurements. The whole network energy consumption can be decreased if the nodes are managed in order to provide just enough information for the application implemented on top of the WSN to run properly. Broadly speaking, this means that if the application requires $n_{act} < n$ active nodes that send their measurements, $n - n_{act}$ nodes can be placed in the *Standby* mode that consumes less energy than the *Active* mode. In this situation, the WSN lifespan is the length of time the control strategy can guarantee $n_{act} = d_1$ nodes in the *Active* mode M_1.

The constraints that define the mission (2.13) can be dynamically changed, depending on a time schedule or on external events. This *dynamic mission* allows to adjust the needs of the application during the system evolution. For instance, during the day period, when people are present in an office, the mission can be defined as $d_1 = a < n$ nodes in the *Active* mode while during the night period, when there is nobody in the office, the mission becomes $d_1 = b < a$.

2.4 MPC design

The minimization of the energy consumption (discrete-time model) of (2.7) can be done using a MPC approach that is caustically solved *via* a QP problem. MPC can be also used for controlling systems that involve a mix of real-valued dynamics and logical rules, see e.g. [130]. Unfortunately, when this problem is formulated as an optimization one, it is no longer a QP problem but a MIQP one. This latter involves optimization variables that are real values, but also integer values or even binary values, which makes the problem harder to solve than an ordinary QP problem. It must be noted that MIQP leads to the optimization of a quadratic function over points in a polyhedral set where some of the components are restricted to be integer. Thus, MIQP is in NP[1] (non-deterministic polynomial time), see [135]. This complexity has a direct influence on the size of the problem solvable in real time for a given computer.

At each decision time kT_c, the current state (assumed to be available) $\mathbf{x}_k = \mathbf{x}_{k|k}$ can be used to find the optimal control sequence $\mathbf{u}^* = [\mathbf{u}_{k|k}^T, \ldots, \mathbf{u}_{k+N_p-1|k}^T]^T$,

[1] In computational complexity theory, NP is one of the complexity class.

where $\mathbf{u}_{k|k} = \mathbf{u}_k$, by means of the following minimization problem:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \left\{ \sum_{j=0}^{N_p} (\mathbf{X}^{max} - \mathbf{x}_{k+j|k})^T Q (\mathbf{X}^{max} - \mathbf{x}_{k+j|k}) + \sum_{j=0}^{N_u-1} \mathbf{u}_{k+j|k}^T R \mathbf{u}_{k+j|k} \right\}$$

subject to:

$$\begin{cases} \mathbf{x}_{k+j+1|k} = A\mathbf{x}_{k+j|k} + B\mathbf{u}_{k+j|k} + E\mathbf{w}_{k+j|k}, \;\; j = 0, 1, \ldots, N_p, \\ \mathbf{X}^{min} \leqslant \mathbf{x}_{k+j|k} \leqslant \mathbf{X}^{max}, \;\; j = 0, 1, \ldots, N_p, \\ \sum_{i=1}^{n} u_{ih} = d_h \text{ for each } (k+j|k), \\ \sum_{h=1}^{m} u_{ih} = 1 \text{ for each } (k+j|k), \\ \mathbf{u}_{k+j|k} \in \{0,1\}^{nm}, \;\; j = 0, 1, \ldots, N_u - 1, \end{cases} \tag{2.14}$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{nm \times nm}$ are symmetric positive semi-definite and symmetric positive definite weighting matrices, respectively.

N_p and $N_u \leqslant N_p$ are the prediction and control horizons respectively. Here, we choose $N_u = N_p$, in order to not violate the constraints (2.10) and (2.13). $\mathbf{X}^{min} = \begin{bmatrix} X_1^{min} & \cdots & X_n^{min} \end{bmatrix}^T$ and $\mathbf{X}^{max} = \begin{bmatrix} X_1^{max} & \cdots & X_n^{max} \end{bmatrix}^T$ are the lower and upper bounds on the battery state, given in vector form, when all nodes are considered.

The inequality constraints on the system state $\mathbf{x}_{k+j|k}$ and the equality constraints on the control $\mathbf{u}_{k+j|k}$ can be fully described with matrices $F_{in_x} \in \mathbb{R}^{n \times 2n}$, $G_{in_x} \in \mathbb{R}^n$ and $F_{eq_u} \in \mathbb{R}^{(n+m) \times nm}$, $G_{eq_u} \in \mathbb{R}^{n+m}$ respectively. Basically, the inequality constraints (2.8) are rewritten as follows:

$$\underbrace{\begin{bmatrix} \mathbf{I}_n \\ -\mathbf{I}_n \end{bmatrix}}_{F_{in_x}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq \underbrace{\begin{bmatrix} \mathbf{X}^{max} \\ \mathbf{X}^{min} \end{bmatrix}}_{G_{in_x}}, \tag{2.15}$$

while the equality constraints (2.10) and (2.13) are rewritten as:

$$\underbrace{\begin{bmatrix} [1\,1\cdots1] & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & [1\,1\cdots1] \end{bmatrix}}_{F^1_{eq_u}} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}_{G^1_{eq_u}}, \tag{2.16}$$

$$\underbrace{\begin{bmatrix} [1\,0\cdots0] & [1\,0\cdots0] & \cdots & [1\,0\cdots0] \\ \vdots & \vdots & \ddots & \vdots \\ [0\,0\cdots1] & [0\,0\cdots1] & \cdots & [0\,0\cdots1] \end{bmatrix}}_{F^2_{eq_u}} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \underbrace{\begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix}}_{G^2_{eq_u}}, \tag{2.17}$$

where d_h, $h = 1, ..., m$, corresponds to the number of nodes in mode M_h. The matrices F_{eq_u} and G_{eq_u} are defined as $F_{eq_u} = \begin{bmatrix} F^1_{eq_u} & F^2_{eq_u} \end{bmatrix}^T$, $G_{eq_u} = \begin{bmatrix} G^1_{eq_u} & G^2_{eq_u} \end{bmatrix}^T$.

By compacting the states involved in the optimization problem (2.14) as $\mathbf{x} = [\mathbf{x}^T_{k+1|k}, \ldots, \mathbf{x}^T_{k+N_p|k}]^T$ with $\mathbf{X}^{max}_{ex} = [(\mathbf{X}^{max})^T, \ldots, (\mathbf{X}^{max})^T]^T \in \mathbb{R}^{nN_p}$, the cost function is rewritten in a matrix form as:

$$\begin{aligned} &\arg\min_{\mathbf{u}} \ \left\{(\mathbf{X}^{max}_{ex} - \mathbf{x})^T \bar{Q} (\mathbf{X}^{max}_{ex} - \mathbf{x}) + \mathbf{u}^T \bar{R} \mathbf{u}\right\} \\ &\text{subject to: } \begin{cases} \bar{F}_{in_x} \mathbf{x} \leqslant \bar{G}_{in_x}, \\ \bar{F}_{eq_u} \mathbf{u} = \bar{G}_{eq_u}, \\ \mathbf{u} \in \{0,1\}^{N_u nm}, \end{cases} \end{aligned} \tag{2.18}$$

where $\bar{Q} = \text{diag}(Q, \ldots, Q) \in \mathbb{R}^{N_p n \times N_p n}$, $\bar{R} = \text{diag}(R, \ldots, R) \in \mathbb{R}^{N_u nm \times N_u nm}$. Now, the inequality and equality constraints that express the constraints (2.8), (2.10) and (2.13) on $\mathbf{x}$ and $\mathbf{u}$, are presented as:

$$\begin{aligned} \bar{F}_{in_x} &= \text{diag}(F_{in_x}, \ldots, F_{in_x}) \in \mathbb{R}^{s \times p}, \\ \bar{G}_{in_x} &= \text{diag}(G_{in_x}, \ldots, G_{in_x}) \in \mathbb{R}^{s}, \\ \bar{F}_{eq_u} &= \text{diag}(F_{eq_u}, \ldots, F_{eq_u}) \in \mathbb{R}^{r \times q}, \\ \bar{G}_{eq_u} &= \text{diag}(G_{eq_u}, \ldots, G_{eq_u}) \in \mathbb{R}^{r}, \end{aligned} \tag{2.19}$$

where $s = 2N_p n$, $p = N_p n$, $r = N_u(n+m)$, $q = N_u nm$, respectively.

The state-space system can now be written in a compact matrix form:

$$\begin{aligned} &\mathbf{x} = \Phi \mathbf{x}_{k|k} + \Gamma \mathbf{u} = \Phi x + \Gamma \mathbf{u}, \\ &\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \ \Gamma = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1} & A^{N_p-2} & \ldots & A^{N_p-N_u} \end{bmatrix}, \end{aligned} \tag{2.20}$$

where $x = \mathbf{x}_{k|k}$.

Then, (2.18) is expressed as (see e.g. [120]):

$$\begin{aligned} &\arg\min_{\mathbf{u}} \ \left\{\mathbf{u}^T H \mathbf{u} + 2\mathbf{u}^T F(x - \mathbf{X}^{max}) + (x - 2\mathbf{X}^{max})^T G x + (\mathbf{X}^{max})^T \bar{Q} (\mathbf{X}^{max})\right\} \\ &\text{subject to: } \begin{cases} \bar{F}_{in_x} \Gamma \mathbf{u} \leqslant \bar{G}_{in_x} - \bar{F}_{in_x} \Phi x, \\ \bar{F}_{eq_u} \mathbf{u} = \bar{G}_{eq_u}, \end{cases} \end{aligned} \tag{2.21}$$

where $2\mathbf{u}^T F(x - \mathbf{X}^{max}) \geq 0$ and

$$\begin{aligned} H &= \Gamma^T \bar{Q} \Gamma + \bar{R}, \\ F &= \Gamma^T \bar{Q} \Phi, \\ G &= \Phi^T \bar{Q} \Phi. \end{aligned} \tag{2.22}$$

Note that $(x - 2\mathbf{X}^{max})^T Gx$ and $(\mathbf{X}^{max})^T \bar{Q}(\mathbf{X}^{max})$ are constant, thus they are not influencing the result. As a consequence, they will be further omitted.

The prediction N_p and control N_u horizons, together with matrices Q and R are the degrees of freedom of the control design strategy, specified by the user. The choice of the weighting matrices depends on the control objectives, which are motivated here from the requirement to minimize the energy consumption of the set of nodes while fulfilling a pre-defined mission.

2.5 Simulation results with the MPC strategy

The MPC strategy is now applied in simulation on a realistic benchmark. Note that the numerical values involved in the simulations come from the data-sheets of the devices at hand (sensor nodes, batteries) or from laboratory measurements. The simulation benchmark is first described and results are discussed.

2.5.1 Benchmark description

The proposed energy management strategy is exemplified with $n = 6$ nodes, possibly embedding harvesting systems. We consider that each node can be in one of three states: $m = 2$ functioning modes and the *Unreachable* condition. This choice is conducted by the actual capabilities of the real nodes at hand: they can be placed in *Active* and *Standby* modes. Moreover due to environmental disturbances (e.g. communication breakdown) or energy shortage, an *Unreachable* condition, described hereafter, is defined. This choice is similar to the one in [136].

The modes are defined in accordance with the state of the node units, as shown in Table 2.2.

Table 2.2: Functioning modes for node S_i

Mode	Comput. unit	Com. unit	Sensing unit
M_1	On	Radio (Tx, Rx)	On
M_2	Sleep	Off	Off

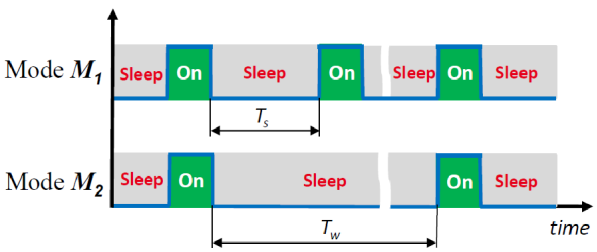


Figure 2.4: Duty cycles for modes M_1 and M_2

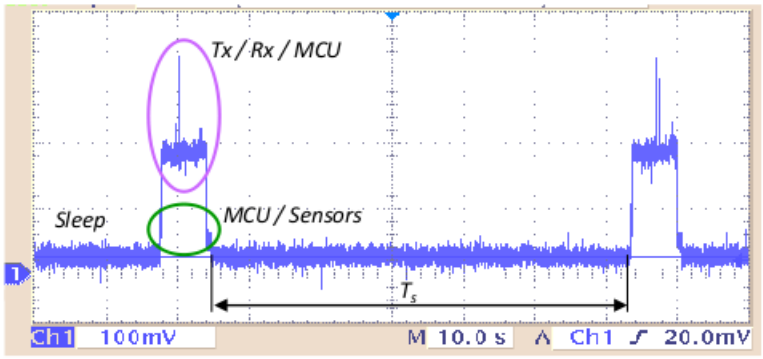


Figure 2.5: Typical waveform of the current consumption for a node working in mode M_1 (current consumption measured with a shunt resistance of 1Ω)

The three states of the nodes, namely, two functioning modes and one *Unreachable* condition, are described as follows:

- M_1 is the *Active* mode. In this mode, sensing, computing and communication units are "duty cycled" [137] as shown in Fig. 2.4. Each unit is asleep by default and it wakes up periodically with a sampling period $T_s = 1\ min$[2] to sense, process and exchange data with the supervisor. When a sensor node sleeps, only the Real Time Clock (RTC) Quartz system (one part of the computing unit) is active. Indeed, the RTC wakes up the node at each sampling time kT_s. The choice of T_s depends of the application and on the dynamics of the physical phenomenon that is monitored by the node. It can be modified by the supervisor. For instance, when an event such as "smoke in a room" is detected by a sensor node, it is first analyzed and pre-validated by the sensor node. Then, it is transmitted to the supervisor. Based on this measurement, the supervisor can decide to reduce the sampling period T_s of the node and even place more nodes in the *Active* mode to monitor and analyze more precisely the presence of smoke so as to choose the appropriate action(s) [138] that must be performed (e.g. raise an alarm, send a message to the firefighter brigade, etc.). Therefore, events can trigger *via* the supervisor a reconfiguration of the control strategy. Fig. 2.5 shows the current consumption waveform of a sensor node working in mode M_1 (laboratory measurements). The waveform is typical of a duty cycled node: the node is awake from the sleep state. It collects data and prepares the packets to be transmitted. Then, the packets are sent to the supervisor, and the node goes back to the sleep state;

- M_2 corresponds to the *Standby* mode. In the present case, M_2 is also duty

[2]Note that T_s can be modified, depending on the application run on top of the WSN.

cycled, but with a much larger sampling period $T_w = 1\ hour^3 \gg T_s$ as depicted in Fig. 2.4. In this mode, the node remains in the sleep state and it wakes up each $T_w = T_c$ to monitor the remaining energy level in the batteries, to sens it the supervisor, and to receive the control (i.e. the mode in which it will work during the next control period T_c) from the supervisor;

- the *Unreachable* condition corresponds to the situation when the sensor node is not seen by the supervisor: this latter has not received information regarding the remaining energy level y_i of node S_i at time k. This can occur because of node physical damages or/and strong perturbations of the radio channel. It can also be caused by a lack of power resources $x_i \leq X_i^{min}$. When $x_i = X_i^{min}$, the lifespan of node S_i (see (2.12)) is equal to zero. In this situation we consider that the node is unreachable. The supervisor can also decide to place a sensor node in the *Unreachable* condition based on data analysis, for instance, if the measurements sent by the sensor node are inconsistent with the measurements that are expected. However, this is a particular situation that is not considered here. The sensor node can exit from this condition (become reachable again) when for instance, the battery is recharged by the harvesting system ($x_i > X_i^{min}$), the physical damages are repaired or/and the radio channel is back to normal condition.

  During the simulation of communication perturbations, we consider that the unreachable node still consumes the same amount of energy as in mode M_1. This corresponds to the so-called "best effort" situation.

  In the real-life conditions, we know neither the remaining energy level $\mathbf{y}_k$ nor the real consumption of an unreachable node. However, when it comes back (i.e. it is again reachable by the supervisor), it sends its remaining energy and the system state is updated. Then, the MPC strategy will be able to consider this node as a node to be placed in *Active* or *Standby* mode.

Remark 1. *When a sensor node becomes unreachable, it is "removed" from the optimization problem, i.e. it is not considered in the problem* (2.21)*. This means that the dimension of the problem will change each time a sensor node becomes unreachable/reachable, as show in Figure 2.6. In Chapter 3 we will show a different way to deal with this problem using Hybrid Dynamic theory.*

In the present simulation, the mission is defined with $d_1 = 3$ nodes placed in the *Active* mode. These three nodes in mode M_1 (active nodes) are considered sufficient to guarantee the service the WSN has to provide to the application built on top of the network.

[3]Here also, T_w can be modified, depending on the application.

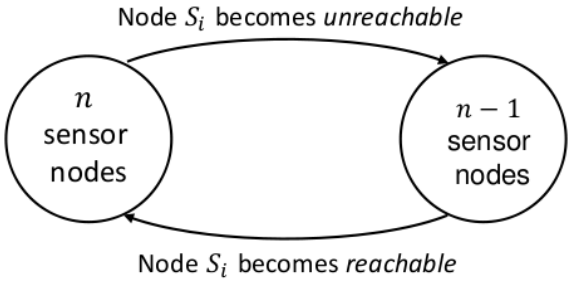


Figure 2.6: Problem dimension changes

2.5.2 Control configuration

For the system (2.7), $A = \mathbf{I}_6$. The components of matrix B are calculated from the values in Table 2.3 multiplied by the nominal voltage of the battery of the corresponding node, see Table 2.4 where the battery characteristics associated with each node are summarized. Note that the numerical values are derived from the technical data-sheet of the OpenPicus node [112] and from laboratory measurements. Table 2.4 also provides the initial energy capacity of the batteries associated with each sensor node. These latter numerical values are obtained from the technical data-sheet of Li-polymer rechargeable batteries [139]. We consider that the value related to the minimal energy level of the sensor node S_i is $\gamma_i = 0.05$ for all these batteries[4].

Table 2.3: Average current consumption b_{ij}/V_i for the sensor nodes in modes M_1 and M_2

Sensor node	Average current consumption in mode M_1, $[mA \cdot hour]$	Average current consumption in mode M_2, $[mA \cdot hour]$
S_1	9.42	1.58
S_2	9.86	1.63
S_3	9.89	1.65
S_4	9.86	1.63
S_5	9.88	1.65
S_6	8.93	1.55

For the harvesting systems, we consider that the average energy of a solar cell [140] can be recovered during the day period. The availability of the harvesting systems is depicted in Figure 2.7 while the numerical values are given in Table 2.4. The state constraints (2.8), i.e. the maximal and minimal energy level in

[4]The values γ_i, $i = , ..., 6$, are not specified in the the technical data-sheet of Li-polymer rechargeable batteries. However from our knowhow for the simulations, we have chosen $\gamma_i = 0.05$ of all node batteries.

Table 2.4: Node battery and harvesting system characteristics

Sensor node	Battery type	Nominal voltage V_i, $[V]$	Nominal battery capacity P_i, $[mA \cdot hour]$	Harvesting availability E_i/V_i, $[mA \cdot hour]$	Energy coef. ξ_i, [1]	Harvesting period, *per* 24 *hours*
S_1	LiPo	3.7	1050	missing	1	–
S_2	LiPo	3.7	1050	missing	0.8	–
S_3	LiPo	3.7	1050	21	0.9	7h-12h
S_4	LiPo	3.7	950	missing	0.7	–
S_5	LiPo	3.7	950	27	1	13h-18h
S_6	LiPo	3.7	2300	missing	1	–

the battery of node S_i, are defined as $X_i^{max} = P_i \cdot V_i$ and $X_i^{min} = \gamma_i \cdot X_i^{max}$, respectively. P_i is the nominal capacity and V_i is the nominal voltage of the node battery. The initial state value $\mathbf{x}_0$ is also provided in Table 2.4: for each node S_i, it is equal to $\xi_i \cdot X_i^{max}$, where ξ_i is any energy coefficient.

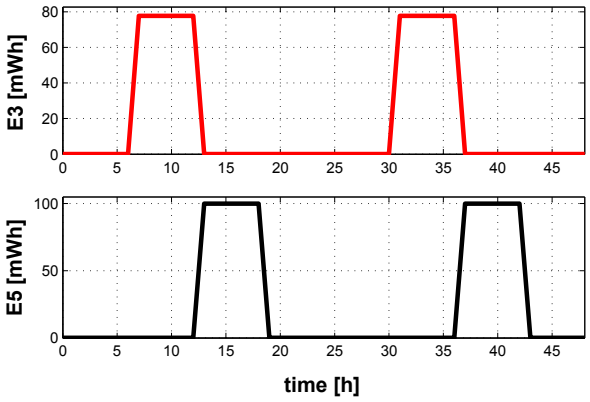


Figure 2.7: Availability of the energy harvesting systems (over 48 hours)

The weighting matrices Q and R that appear in the definitions of $\bar{Q}$ and $\bar{R}$ in (2.21) are chosen depending on the control objectives as follows:

1. To penalize the sensor nodes with the higher consumption compared to the other nodes, Q and R are defined as $Q = \mathbf{0}_{6\times 6}$, $R = B_{cons}^T \times B_{cons}$, where $B_{cons} = \text{diag}(b_{11}, b_{12}, ..., b_{61}, b_{62})$. B_{cons} is derived from B (see Table 2.3), where the components b_{ih}, $i = 1, ..., 6$, $h = 1, 2$, of B have been placed diagonally to ensure that R is a symmetric positive definite matrix. The

choice $Q = \mathbf{0}_{6\times 6}$ lies in the fact that the state dynamics should evolve as smoothly as possible (see e.g. [141]);

2. To ensure a trade-off between the smaller energy consumption and the larger battery energy level for the nodes to be activated, Q and R are defined as $Q = \text{diag}(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$, $R = B_{cons}^T \times B_{cons}$.

It is also possible to choose other weighting matrices Q and R in order to cope with other control objectives.

The inequality constraints (2.8) are rewritten as follows:

$$\underbrace{\begin{bmatrix}\mathbf{I}_6\\ -\mathbf{I}_6\end{bmatrix}}_{F_{in_x}}\begin{bmatrix}x_1\\ \vdots\\ x_6\end{bmatrix} \leq \underbrace{\begin{bmatrix}X_1^{max} & \cdots & X_6^{max} & X_1^{min} & \cdots & X_6^{min}\end{bmatrix}^T}_{G_{in_x}}, \tag{2.23}$$

while the equality constraints (2.10) and (2.13) become:

$$\underbrace{\begin{bmatrix}1\,1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\,1\end{bmatrix}}_{F_{equ}^1}\begin{bmatrix}u_{1h}\\ \vdots\\ u_{6h}\end{bmatrix} = \underbrace{\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}}_{G_{equ}^1}, \tag{2.24}$$

$$\underbrace{\begin{bmatrix}1\,0 & 1\,0 & \cdots & 1\,0\\ 0\,1 & 0\,1 & \cdots & 0\,1\end{bmatrix}}_{F_{equ}^2}\begin{bmatrix}u_{1h}\\ \vdots\\ u_{6h}\end{bmatrix} = \underbrace{\begin{bmatrix}d_1\\ d_2\end{bmatrix}}_{G_{equ}^2}, \tag{2.25}$$

where d_1 and d_2 correspond to the number of nodes in modes M_1 and M_2, respectively. During this simulation $d_1 = 3$, the other nodes being placed in mode M_2, except if they fall in the *Unreachable* condition. Suppose that n_u nodes are in the *Unreachable* condition. These n_u nodes are removed from the optimization problem. Therefore the number of nodes placed in mode M_2 is equal to $d_2 = n - d_1 - n_u$. Note that if $n - n_u > d_1$, the system cannot be properly controlled because we do not have enough measurements. This situation also means that the mission cannot be fulfilled. The matrices F_{equ} and G_{equ} are defined as $F_{equ} = \begin{bmatrix}F_{equ}^1 & F_{equ}^2\end{bmatrix}^T$, $G_{equ} = \begin{bmatrix}G_{equ}^1 & G_{equ}^2\end{bmatrix}^T$. The prediction and control horizons are chosen equal to $N_p = 2$, $N_u = N_p$ respectively. The system considered during the simulation presents slow dynamics. However, a node can become unreachable at any moment, so we consider that the prediction horizon is long enough.

The decision period (i.e. the time period when the control strategy is run) is initially $T_c = 1$ *hour*. Thus, the MIQP problem is solved on-line at each decision time kT_c, except if events occur. For instance, when a node in the

Active mode disappears (i.e it falls in the *Unreachable* condition) at time t_1, $kT_c < t_1 < (k+1)T_c$, a reconfiguration of the set of nodes is required to fulfill the mission. Then, the supervisor wakes up one node from the *Standby* mode (i.e. the sampling period of the *Standby* mode is equal to $T_w = t_1$), based on the computation of the new control, to replace the node fallen in the *Unreachable* condition. Note that this mechanism means that the node in the *Standby* mode can be waked up when mandatory. This mechanism is not currently implemented in most of the industrial node platforms. More, this mechanism is not existing on the OpenPicus platform that will be used for the experiments.

2.5.3 Simulation results

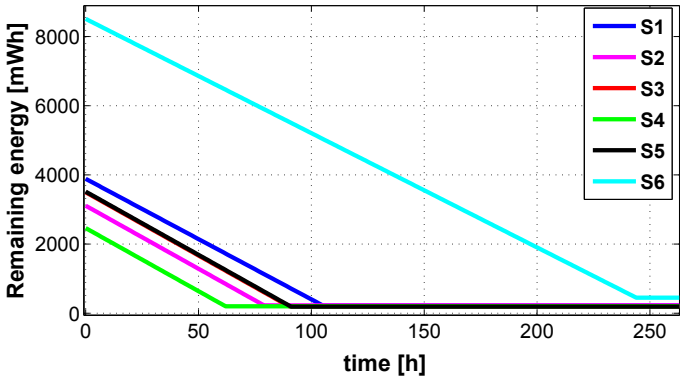
The energy management of the sensor nodes at application level is evaluated in simulation in the Matlab environment. The MIQP problem is solved with Mosek [142] solver using YALMIP [143]. YALMIP is a modeling language for the solution of convex and non-convex optimization problems. The language is consistent with standard Matlab syntax, thus making it extremely simple to use for anyone familiar with Matlab.

Four scenarios are considered. The first one corresponds to the case without any control of the nodes: all nodes are placed in the *Active* mode, and no reconfiguration is performed. The second one corresponds to a finite-state automaton. Three nodes are in the active mode and as soon as a node becomes unreachable another one becomes active. The other two scenarios implement the proposed energy management strategy with two different choices of the weighting matrices Q, R. The simulation results obtained for the four scenarios are now presented and discussed.

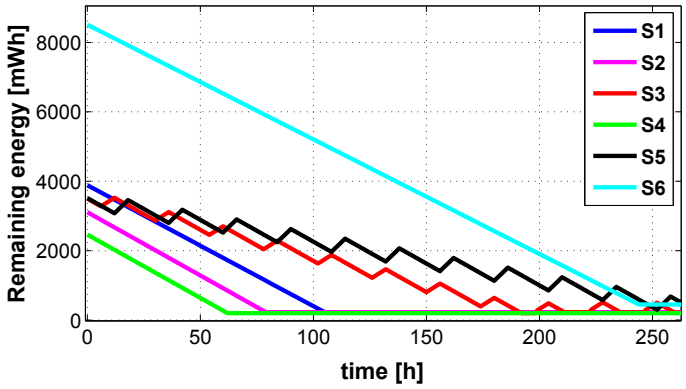
Scenario 2.1: without control of the modes of the nodes at application level ("dummy" situation)

In this scenario, all sensor nodes are in the *Active* mode at the beginning of the simulation. Their mode is not modified and they evolve from the *Active* mode till the *Unreachable* condition, i.e. when the node battery attains its minimum energy level X_i^{min}. This situation is basically what happens with most of the commercial nodes in today WSN deployments. Figure 2.8(a) and Figure 2.8(b) show the remaining energy evolution in each node battery without and with harvesting systems respectively. Note that harvesting systems are considered for nodes 3 and 5 (see Figure 2.7).

The network lifespan (corresponding to the time period when the mission is fulfilled) is equal to 91 *hours* without harvesting system, and to 192 *hours* with harvesting systems. We can observe that the remaining energy in the sensor nodes is never equal to 0, see e.g. Figure 2.8(a). This limit has been introduced



(a) Without harvesting systems

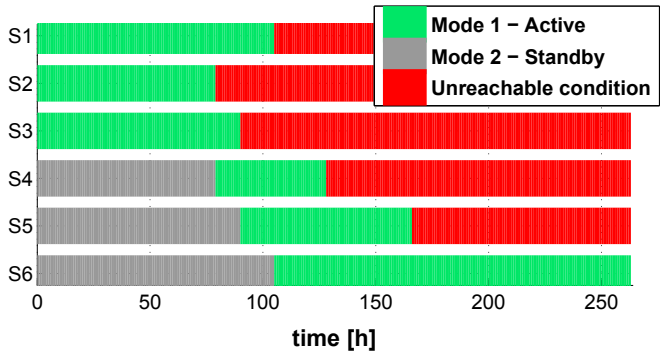


(b) With harvesting systems

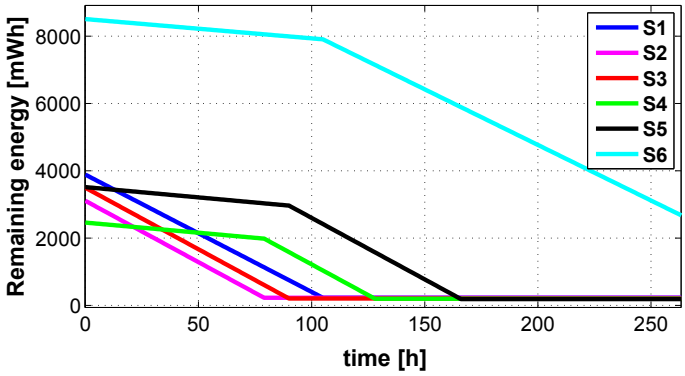
Figure 2.8: Scenario 2.1: evolution of the remaining energy in each node battery without control

in the simulation via the constraint in (2.8) in order to be consistent with real-life constraints on the batteries. Indeed, batteries embed energy limiters to avoid any damage when they are drained. The oscillatory behavior of the remaining energy in the batteries of nodes S_3 and S_5 is consistent with the harvesting profile.

Scenario 2.2: control based on a finite-state automaton ("basic" situation)



(a) Evolution of the functioning modes of the nodes



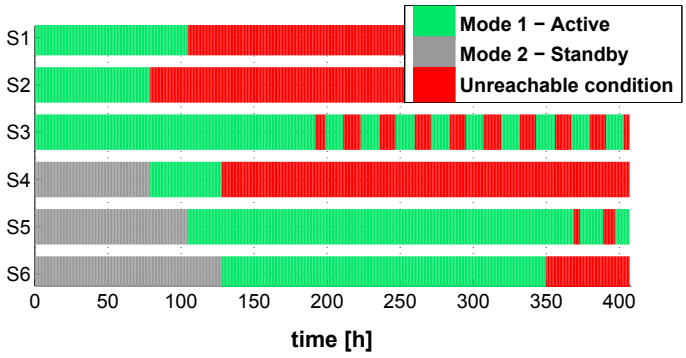
(b) Evolution of the remaining energy in each node battery

Figure 2.9: Scenario 2.2: energy management based on a finite-state automaton without harvesting systems

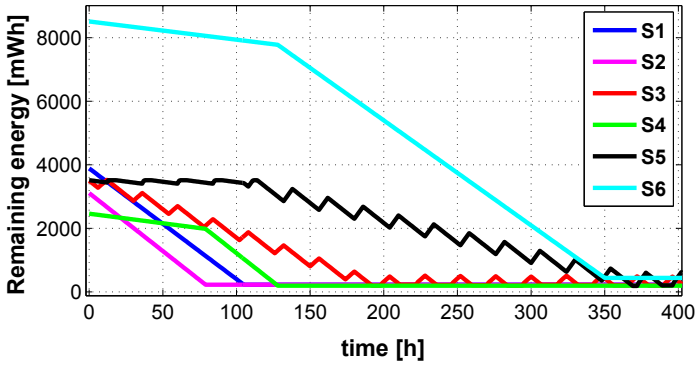
This scenario presents the management of the node activity to fulfill the mission (2.13) based on a finite-state automaton, so that there are any $d_1 = 3$ sensor nodes that are active at any time. In this simulation, the first three nodes are placed in the *Active* mode. Figure 2.9(a) and Figure 2.9(b) (resp. Figure 2.10(a) and Figure 2.10(b)) show the evolution of the functioning modes and the remaining energy for each node without (resp. with) harvesting systems. The WSN lifespan is equal to 128 *hours* (resp. 192 *hours*) without (resp. with) harvesting systems.

We can see that these nodes can exit from the *Unreachable* condition when they have enough energy to be placed in mode M_1. In the case with harvesting systems, the node S_3 runs out of energy at time $k = 192$ *hours*, i.e. it falls in

the *Unreachable* condition. Then it gains a bit of energy at time $k = 199$ *hours* and the mission can be fulfilled again during 13 *hours*. This process of fulfilling of the mission is repeated with the period of 24 *hours* until $k = 350$ *hours*, when the node S_6 falls in the *Unreachable* condition.



(a) Evolution of the functioning modes of the nodes



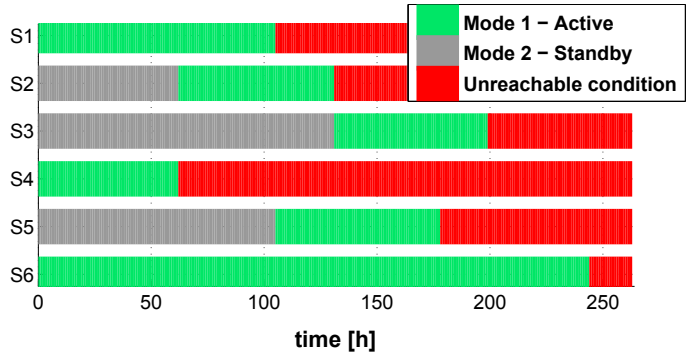
(b) Evolution of the remaining energy in each node battery

Figure 2.10: Scenario 2.2: energy management based on a finite-state automaton with harvesting systems

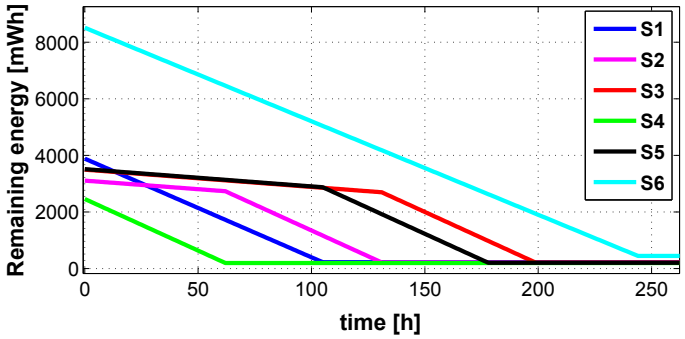
Scenario 2.3: energy management strategy based on the MPC approach, $Q = \mathbf{0}_{6\times6}$ and $R = B_{cons}^T \times B_{cons}$

In this scenario, we evaluate the proposed control strategy with $Q = \mathbf{0}_{6\times6}$ and $R = B_{cons}^T \times B_{cons}$ as it is marked in Section 2.5.2. This implies that the nodes with larger energy consumption are more penalized and thus less used.

Figure 2.11(a) shows the evolution of the functioning modes for each node without harvesting systems, i.e. $\mathbf{w}_k = \mathbf{0}_n$ in (2.7). The energy in the battery of each node is given in Figure 2.11(b). As can be seen, the mission (3 nodes in mode M_1) is fulfilled until $k = 178$ *hours*, that represents the WSN lifespan. Note that, as expected, the sensor nodes which are less consuming are automatically placed in the *Active* mode.



(a) Evolution of the functioning modes of the nodes

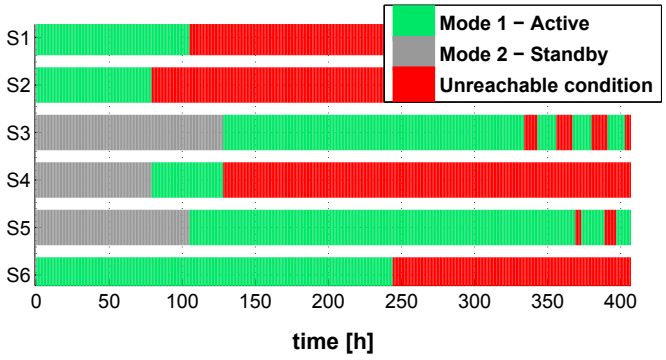


(b) Evolution of the remaining energy in each node battery

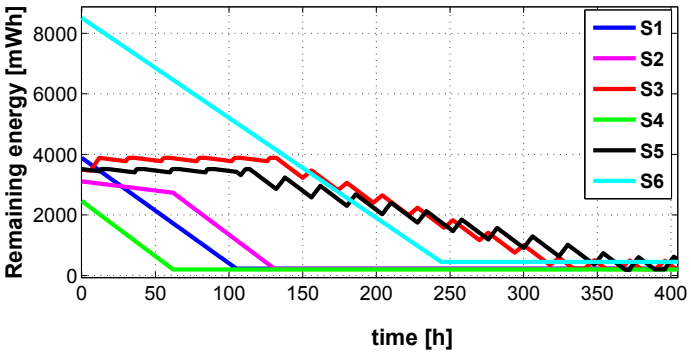
Figure 2.11: Scenario 2.3: energy management strategy based on MPC, $Q = \mathbf{0}_{6\times6}$ and $R = B_{cons}^T \times B_{cons}$, without harvesting systems

The results obtained with harvesting systems are shown in Figure 2.12(a) and Figure 2.12(b). The WSN lifespan is here equal to 244 *hours*. We can

observe that a node can exit from the *Unreachable* condition when it has enough energy to be placed in mode M_1. This situation appears for node S_3 after time $k = 343$ $hours$ and for node S_5 after time $k = 373$ $hours$.



(a) Evolution of the functioning modes of the nodes

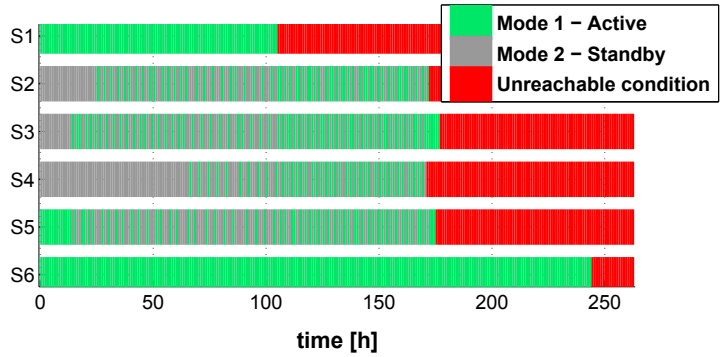


(b) Evolution of the remaining energy in each node battery

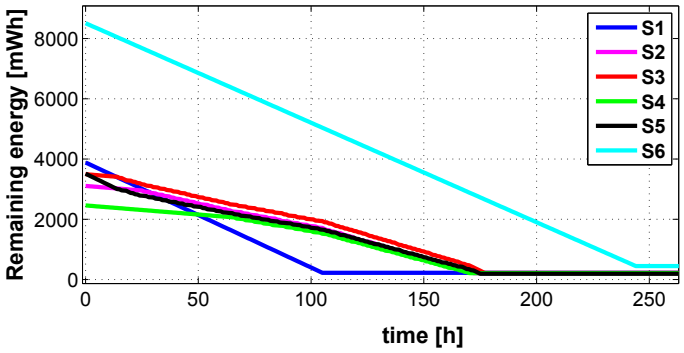
Figure 2.12: Scenario 2.3: energy management strategy based on MPC , $Q = \mathbf{0}_{6\times6}$ and $R = B_{cons}^T \times B_{cons}$, with harvesting systems

Notice that the initial choice of the active nodes (S_1, S_2, S_6) made by the control strategy is equivalent from a consumption perspective to the one (S_1, S_4, S_6) that appears in Scenario 2.3 without harvesting systems, see Figure 2.11.

Scenario 2.4: energy management strategy based on the MPC approach, $Q = diag(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$ and $R = B_{cons}^T \times B_{cons}$



(a) Evolution of the functioning modes of the nodes



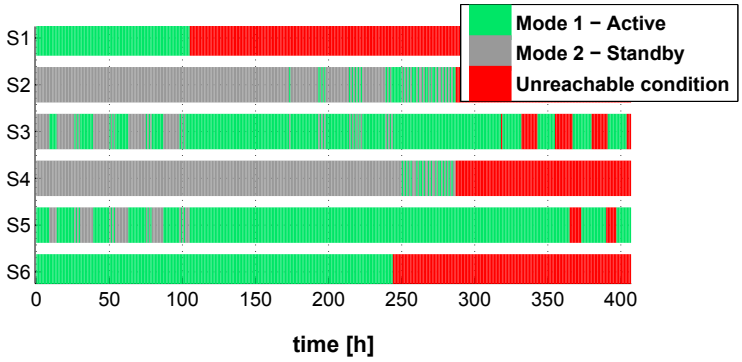
(b) Evolution of the remaining energy in each node battery

Figure 2.13: Scenario 2.4: energy management strategy based on MPC, $Q = \text{diag}(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$ and $R = B_{cons}^T \times B_{cons}$ without harvesting systems

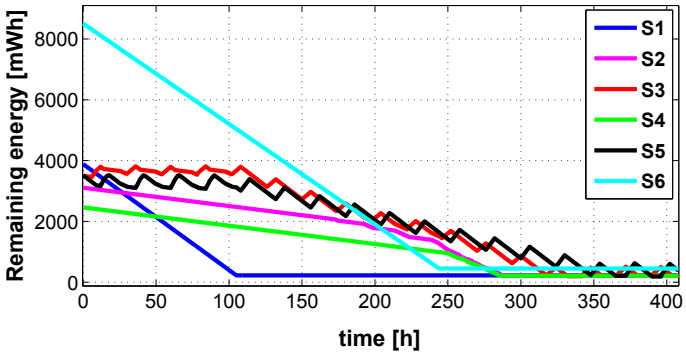
Scenario 2.4 is similar to Scenario 2.3 but with $Q = \text{diag}(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$, $R = B_{cons}^T \times B_{cons}$. This choice allows for a trade-off between less consuming sensor nodes and larger amount of energy in their battery. Figure 2.13(a) and Figure 2.13(b) show the evolution of the functioning modes and the remaining energy for each node without harvesting systems, respectively. The WSN lifespan is equal to 175 *hours*. In Figure 2.13(b), we can see that the remaining energy in the nodes is quite balanced after a transition period. However, to ensure this well balanced situation, a large number of switches (the order of 10^2) has been needed. These extra switches cause additional consumption corresponding to the energy switch cost. Note that this latter is coarsely taken into account in the

system modeling (2.7) as it is supposed to be taken into account in matrix B. Actually, a model that implicitly takes into account this switching cost should be developed.

Scenario 2.4 is more interesting when the harvesting systems are used, because the sensor nodes with harvesting systems are used more often even if their consumption is larger than the consumption of the other nodes. In this case, the network lifespan is increased up to 287 *hours*. The system evolution with the harvesting systems is displayed in Figure 2.14.



(a) Evolution of the functioning modes of the nodes



(b) Evolution of the remaining energy in each node battery

Figure 2.14: Scenario 2.4: energy management strategy based on MPC, $Q = \text{diag}(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$ and $R = B_{cons}^T \times B_{cons}$ with harvesting systems

Comparison of the results and discussion

A summary of the four scenarios is presented in Table 2.5. With the proposed control strategy (Scenario 2.3 and 2.4), it is possible to increase the network lifespan by a factor of 2 without harvesting systems and by a factor of 1.5 with harvesting systems, when compared to the "dummy" scheme in Scenario 2.1.

The WSN lifespan is larger by a factor of 1.4 without harvesting systems, and by a factor of 1.5 with harvesting systems, when compared to the basic situation from the Scenario 2.2. Using the harvesting systems with their harvesting profiles given in Figure 2.7, the lifespan can be extended a bit more. Note that this lifespan extension depends on the harvesting profiles. Other profiles will give a different system evolution and, consequently, a different WSN lifespan.

Table 2.5: Summary of scenarios

Scenarios	WSN lifespan without harv. sys., $[hours]$	WSN lifespan with harv. sys., $[hours]$	Number of switches, $[n.u.]$
Scenario 2.1: dummy	91	192	0
Scenario 2.2: basic	128	192	3
Scenario 2.3: MIQP	178	244	3
Scenario 2.4: MIQP	175	287	$\approx 10^2$

The proposed energy management strategy applied to the WSN is effective with $n > d_1$ nodes. Otherwise, if $n = d_1$, the control has no flexibility, leading to the situation in Scenario 2.1.

Comparing scenarios 2.3 and 2.4, we can see that Scenario 2.3 is best (the WSN lifespan is larger) without harvesting systems and/or when the nominal battery capacities are pretty much the same. Scenario 2.4 is clearly effective in the case of using harvesting systems (the WSN lifespan is 15% larger compared to Scenario 2.3). However, the number of switches in Scenario 2.4 is very important (10^2 compared to 3 in Scenario 2.3). If we consider that the energy cost of the switch from mode M_1 to mode M_2 (resp. from mode M_2 to mode M_1) is equal to $\delta_i^{1\to 2} = 0.01\ mWh$ (resp. $\delta_i^{2\to 1} = 0.05\ mWh$), the energy cost of 10^2 switches is approximately equivalent to 1 *hour* of WSN lifespan. Note that the switching energy cost is not given in the data-sheet. It can be measured but it strongly depends on environmental conditions.

In real-life conditions, a significant amount of energy is consumed when switching from mode M_2 to mode M_1 (from *Standby* mode to *Active* mode) to turn on the radio [144, 145]. Therefore, reducing the number of switches is mandatory to maximize the WSN lifespan.

2.6 Linear Programming problem formulation

The energy management strategy with the MPC formulation (2.14) described above is now expressed as a Mixed Integer Linear Programming (MILP) problem solution [146, 147], or more precisely as a Binary Linear Programming problem solution.

It makes the control strategy less complex in comparison to the MPC formulation solved with MIQP. In this section, the mathematical formulation of the MILP problem for the energy management of the nodes is first presented. Based on the MILP problem formulation, two additional scenarios are considered in simulation.

2.6.1 Mathematical formulation

The objective of the present work is to maximize the network lifespan while fulfilling a given mission. This is achieved via the minimization of the energy consumption. This can also be achieved by the maximization of the remaining energy in the batteries of the sensor nodes. We now reformulate our objectives in this way.

We need to maximize the remaining energy in the set of sensor nodes $\mathbf{x}_{k+1}$ in (2.7) at time $k+1$ subject to (2.8), (2.10), (2.13):

$$\max_{\mathbf{u}_k} \|\mathbf{x}_{k+1}\|_1 = \max_{\mathbf{u}_k} \|\mathbf{x}_k + B\mathbf{u}_k + E\mathbf{w}_k\|_1 , \tag{2.26}$$

where $\mathbf{x}_k$ is the remaining energy in the whole set of nodes measured at time k. Note that the components of $\mathbf{x}_k$ have positive values. As above, $\mathbf{w}_k$ corresponds to the ability for each node to harvest energy. Basically, $\mathbf{x}_k$ at time k has no influence on the minimization problem, because $\mathbf{x}_k$ is measured. Moreover, $\mathbf{w}_k$ is unknown thus it is not taken into account in the optimization problem. Therefore, the problem (2.26) for the LTI system (2.8) can be rewritten so as to choose the control $\mathbf{u}_k$ at each time k such that the energy consumption of the whole WSN system is minimized:

$$\min_{\mathbf{u}_k} \|-B\mathbf{u}_k\|_1 , \tag{2.27}$$

where $\mathbf{u}_k \in \{0,1\}^{nm}$. Note that each component of $-B\mathbf{u}_k$ is positive.

This problem is indeed a linear optimization problem with the minimization of the energy consumption over the next period of control, with equality and inequality constraints that must be fulfilled. The MPC formulation (2.14) can be rewritten as a MILP by using the approach described in [148]. At each decision time kT_c, the current state $x = \mathbf{x}_k$ can be used to find the optimal control sequence $\mathbf{u}^* = [\mathbf{u}_{k|k}^T, \ldots, \mathbf{u}_{k+N_p-1|k}^T]^T$, where $\mathbf{u}_{k|k} = \mathbf{u}_k$, by means of the following

minimization problem:

$$\begin{aligned} &\mathbf{u}^* = \arg\min_{\mathbf{u}} \bar{c}^T \mathbf{u} \\ &\text{subject to: } \begin{cases} \bar{F}_{in_x} \Gamma \mathbf{u} \leqslant \bar{G}_{in_x} - \bar{F}_{in_x} \Phi x, \\ \bar{F}_{eq_u} \mathbf{u} = \bar{G}_{eq_u}, \end{cases} \end{aligned} \tag{2.28}$$

where $\bar{c} = [c^T \ \cdots \ c^T]^T \in \mathbb{R}^{N_u nm}$ contains the coefficients of the objective function of the problem with $c \in \mathbb{R}^{nm}$. The constrains are defined in Section 2.4.

Vector $\bar{c}$ is the degree of freedom of the control strategy design, specified by the user. Its choice depends on the control objectives, which is here to minimize the energy consumption.

2.6.2 Simulation results

To analyze the energy management strategy described as a MILP problem, two additional scenarios with different problem costs are considered. The prediction and control horizons are chosen equal to $N_p = N_u = 2$ as in all scenarios of Section 2.5.

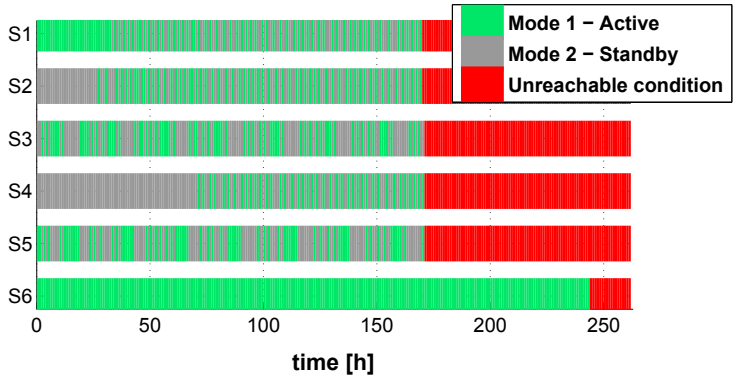
Scenario 2.5: energy management strategy when a MILP problem is solved, with $c = [b_{11} \ b_{12} \ \cdots \ b_{62}]^T$

In this scenario, the objective function is defined with $c = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{62} \end{bmatrix}^T$. This implies that the nodes with larger energy consumption are more penalized and thus, less used. The control objective is similar to the one in Scenario 2.3 (see Section 2.5.3). The network lifespans for both problems are identical. Moreover, the mode evolution of the nodes is equivalent in two solved problems. Namely, the WSN lifespan is equal to 178 *hours* without harvesting systems and to 244 *hours* with harvesting systems.

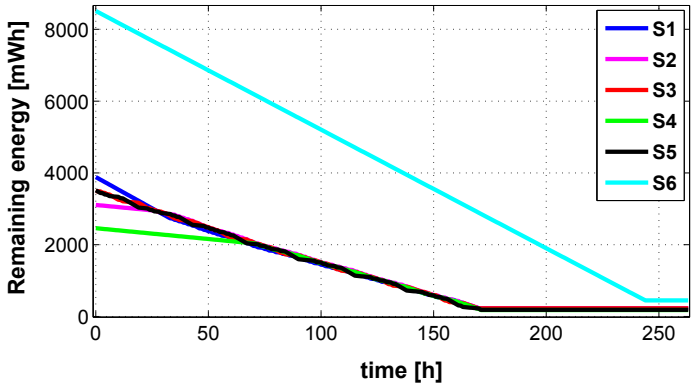
Scenario 2.6: energy management strategy when a MILP problem is solved, with $c = [\frac{b_{11}}{x_1+E_1w_1-X_1^{min}} \ \frac{b_{12}}{x_1+E_1w_1-X_1^{min}} \ \cdots \ \frac{b_{62}}{x_6+E_6w_6-X_6^{min}}]^T$

To penalize and thus less use the sensor nodes with the smaller lifespan (see the definition (2.12)) compared to other nodes, the objective function is defined with $c = \begin{bmatrix} \frac{b_{11}}{x_1+E_1w_1-X_1^{min}} & \frac{b_{12}}{x_1+E_1w_1-X_1^{min}} & \cdots & \frac{b_{62}}{x_6+E_6w_6-X_6^{min}} \end{bmatrix}^T$, where $x_i + E_i w_i - X_i^{min} > 0$. Actually, if $x_i + E_i w_i - X_i^{min} \leq 0$, the sensor node S_i has fallen in the *Unreachable* condition and it has been removed from the optimization problem (2.28).

Here, we can take into account the value of the state $\mathbf{x}_k$ in the cost of the problem, because we consider that $\mathbf{x}_k$ is known. Indeed, it is measured. We can



(a) Evolution of the functioning modes of the nodes

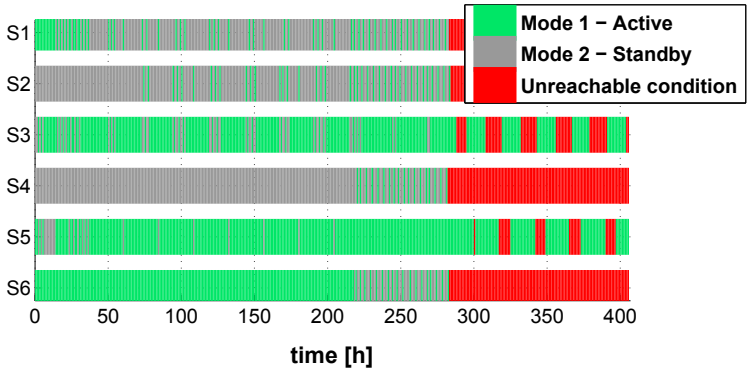


(b) Evolution of the remaining energy in each node battery

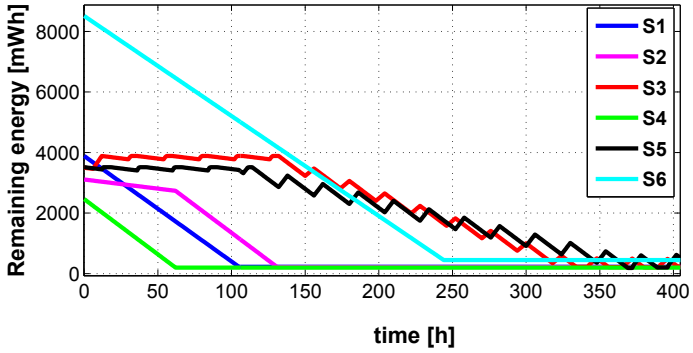
Figure 2.15: Scenario 2.6: with the energy management strategy described as a MILP problem, without harvesting systems

also take into account the amount of harvested energy $E\mathbf{w}_k$ if it can be predicted for the next period of time $[kT_c, (k+1)T_c]$.

Figures 2.15 and 2.16 show the evolution of the functioning modes and the remaining energy of each node without and with harvesting systems, respectively. The WSN lifespan is equal to 171 *hours* (resp. 284 *hours*) without (resp. with) harvesting systems. The number of switches is in the range of 10^2. We can see that the control tries to equalize the lifespan of nodes by preferentially using the sensor nodes with larger lifespan. However, this implies a large number of switches with their associated switching costs.



(a) Evolution of the functioning modes of the nodes



(b) Evolution of the remaining energy in each node battery

Figure 2.16: Scenario 2.6: with the energy management strategy described as a MILP problem, with harvesting systems

Comparison of the results and discussion

The simulation results of all scenarios are summarized in Table 2.6. Scenario 2.6 seems better than Scenario 2.5 from a WSN lifespan point of view, with an increase of 14% of the network lifespan when the harvesting systems are used. Scenarios 2.5 and 2.6 have almost the same lifespan without harvesting systems. However, the number of switches in Scenario 2.6 is very high, in the range of 10^2, that is equivalent to 1 *hour* of WSN lifespan, when compared to 3 switches in Scenario 2.5.

The simulation results for the control strategy, when the optimization problem is solved with MILP, give similar WSN lifespans and similar number of switches compared to the results where the optimization problem is solved with MIQP.

The remaining energy in the nodes never becomes smaller than the given min-

Table 2.6: Comparison of scenarios

Scenarios	WSN lifespan without harv. sys., [*hours*]	WSN lifespan with harv. sys., [*hours*]	Number of switches, [*n.u.*]
Scenario 2.1: dummy	91	192	0
Scenario 2.2: basic	128	192	3
Scenario 2.5: MILP	178	244	3
Scenario 2.6: MILP	171	284	$\approx 10^2$

imal energy level, see (2.8). The sensor nodes can also exit from the *Unreachable* condition when they have regained enough energy to perform their task.

The MILP problem complexity is also NP [149] but the resolution time is shorter. Actually, the MILP problem is solved in ≈ 0.04 *sec* while it takes ≈ 0.48 *sec* when the MIQP problem is solved. These values are obtained in the Matlab environment using the *tic-toc* function. Matlab turns on one core of the Intel Xeon Processor E5620 [150]. Thus, it can be concluded that, due to its computational cost, the control based on the MPC approach and solved with MIQP is effective even if the scale-up in the number of nodes deployed might raise extra challenge-uses, because the solution of this optimization problem requires significant time and energy cost. The control strategy solved with a MILP formulation is less "heavy". A more precise comparison of this control strategy solved with MIQP and with MILP will be presented in Chapter 4.

2.7 Conclusion

This chapter proposes an energy management strategy at application level for a WSN based on MPC. The remaining energy in the node batteries is modeled using a linear state-space representation. Harvesting capabilities of the nodes are also taken into account.

The WSN has to provide a given service (named *mission*), expressed with a set of constraints. The control problem is tackled using MPC, leading to a MIQP problem, when a quadratic cost function is considered, and to a MILP problem, when a l_1 norm cost function is used. A unique functioning mode is imposed to each node at each decision time.

Simulation results on a realistic test-bench have been used to compare the proposed strategy with "dummy" and basic situations. The comparison of the different scenarios show the effectiveness of the proposed strategy solved using MIQP or MILP. Indeed, for a particular scenario, the WSN lifespan can by extended by a factor of 1.4 without harvesting systems and by a factor of 1.5

with harvesting systems, compared to the basic situation from Scenario 2.2.

These simulation results make our control strategy promising when the WSN lifespan increase is considered. Moreover, this control strategy offers a solution for the energy management problem when a set of constraints has to be explicitly taken into account.

However, in the present work the switching cost from one mode to another one is coarsely modeled and a model that explicitly takes this cost into account should be developed.

The problem dimension changes each time a sensor node becomes unreachable/reachable. This dimension change naturally introduces jumps between systems of different structures. As a consequence, a control strategy that naturally takes into account the switching cost and the change in the system structure must be searched. Such a possible solution might be attained using Hybrid Dynamic System theory. This will be presented in the next chapter.

Chapter 3

Energy Management using a Hybrid Control Approach

Hybrid systems are dynamical systems that consist in components with continuous and discrete behaviors. Modeling, analysis, and design of such systems raise severe methodological questions, because they require the combination of continuous system descriptions, like differential and difference equations, with discrete-event models, like automata or Petri nets [151].

The hybrid nature inherent in power control of a WSN system at application level comes from the combination of

- continuous physical processes, namely, the charge/discharge of the energy in the node batteries;
- finite-states such as the functioning modes and the *Unreachable*[1] condition.

Therefore, the Hybrid Dynamic System (HDS) approach is natural to deal with the WSN power management, and also to take into account the switching cost between two states of the node.

In this chapter, we introduce the hybrid dynamical phenomena. The WSN system presented in the previous chapter is now modeled using the HDS approach. The design of the scheduling law is realized for the simple case with two nodes. Then, it is extended to n nodes. The proposed control strategy based on the HDS approach is evaluated in simulation on a realistic test-case similar to the one in Chapter 2. We analyze and compare the simulation results. The research activity presented in this chapter will be published in [152].

[1] *Unreachable* condition corresponds to the situation when the supervisor did not receive the information regarding the remaining energy level of node a time kT_c

3.1 Introduction to HDS

3.1.1 Historical development

Basically, Hybrid Dynamic Systems Theory (the term was used for the first time in [153]) studies the behavior of dynamical systems, which are modeled through mathematical formalisms and that involve both continuous models, such as differential or difference equations, and discrete models, such as finite state machines or automata. The term "hybrid system" is used to indicate that the dynamical system under study is modeled using a mixture of heterogeneous modeling formalisms, namely continuous models as adopted in control theory, and discrete models as used in computer science.

The growing complexity of technological systems required a deeper understanding of the intricate interactions between the discrete and continuous dynamics in the early 1990s. As a result, the academic world started investigating the rich behavior that is produced by these hybrid systems. This is evidenced by the start of a workshop series [154] and several special issues published in that period [155, 156].

With advanced technology development, HDSs are now implemented in various applications [151]. Examples of such systems include devices with impacts and friction in mechanics, walking and hopping machines in robotics, power electronic circuits in electrical engineering, control systems [157] and more recently general regulatory networks [158] and neurons in computational neuroscience and biology [159].

3.1.2 Hybrid dynamical phenomena

An appropriate modeling of hybrid systems can often be obtained by adding new dynamical phenomena to the classical description formats of the monodisciplinary research areas [151]. Indeed, continuous models represented by differential or difference equations, as adopted by the dynamics and control community, can by extended to suitably describe hybrid systems. On the other hand, the discrete models used in computer science, such as automata or finite-state machines, can be extended by concepts like time, clocks, and continuous evolution in order to capture the mixed discrete and continuous evolution of hybrid systems.

The model of a hybrid system can be represented in the following form [160]:

$$\begin{cases} z \in \mathcal{C} & \dot{z} \in F(z), \\ z \in \mathcal{D} & z^+ \in G(z), \end{cases} \tag{3.1}$$

or with equations:

$$\begin{cases} z \in \mathcal{C} & \dot{z} = f(z), \\ z \in \mathcal{D} & z^+ = g(z). \end{cases} \tag{3.2}$$

These representations suggest that the state of the hybrid system, represented by z, can change according to a differential inclusion $\dot{z} \in F(z)$ or according to a differential equation $\dot{z} = f(z)$ while in the set $\mathcal{C}$. It can change according to a difference inclusion $z^+ \in G(z)$ or according to a difference equation $z^+ = g(z)$ while in the set $\mathcal{D}$. The notation $\dot{z}$ represents the derivative of the state z over time, while z^+ represents the value of the state after an instantaneous change.

To shorten the terminology, the behavior of a dynamical system that can be described by a differential equation or inclusion is referred to as *flow*. The behavior of the dynamical system that can be described by a difference equation or inclusion is refereed to as *jump*. This leads to the following names for the four objects involved in (3.1) or (3.2):

- $\mathcal{C}$ is the *flow set*;
- F (or f) is the *flow map*;
- $\mathcal{D}$ is the *jump set*;
- G (or g) is the *jump map*.

3.1.3 Simple example of a hybrid dynamical system

Various examples of hybrid systems from different application areas can be found in the literature [161, 162]. We consider here the simple example of a temperature control system with the aim of illustrating the notions of hybrid models.

In a simple model used for the temperature control of a room [162, 163], we have one continuous variable, namely, the room temperature denoted by x and taking values in $\mathbb{R}$, and one discrete variable, that is the state of the heater denoted by q and taking values in $\{1; 0\}$. "1" stands for the heater is on while "0" means that the heater is off. This system is modeled by including a logical variable q in the state:

$$\dot{z} = \begin{bmatrix} q \\ x \end{bmatrix} \in \mathbb{R}^2. \tag{3.3}$$

The continuous dynamics in the system are described by the differential equation:

$$\begin{cases} \dot{x} &= -x + x_0 + x_\Delta q, \\ \dot{q} &= 0, \end{cases} \tag{3.4}$$

where x_0 represents the natural temperature of the room, x_Δ the capacity of the heater to raise the temperature in the room by being always on. Typically, it is desired to keep the temperature between two specified values x_{min} and x_{max}, given in degree Celsius units, satisfying the following relationship:

$$x_0 < x_{min} < x_{max} < x_0 + x_\Delta. \tag{3.5}$$

For purposes of illustration, consider the case when $x_{min} = 19C^{\circ}$ and $x_{max} = 20C^{\circ}$. A control algorithm that attempts to keep the temperature between such thresholds is shown in Figure 3.1. this algorithm implements the following logic: if the heater is "on" and the temperature is larger then $20C^{\circ}$, then the heater is turn off on the contrary, if the heater is "off" and the temperature is smaller than $19C^{\circ}$, then the heater is turn on.

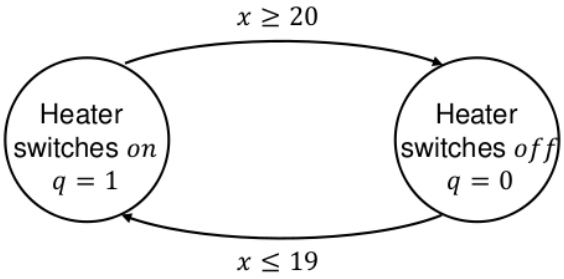


Figure 3.1: Temperature control as a hybrid automaton

The form to represent such as a system in relation with (3.1), is

$$\begin{cases} z \in C & \dot{z} \in F(z), \\ z \in D & (q, z)^{+} \in G(z), \end{cases} \tag{3.6}$$

where $C = (\{0\} \times C_0) \bigcup (\{1\} \times C_1)$ is a flow set, F is a flow map, $D = (\{0\} \times D_0) \bigcup (\{1\} \times D_1)$ is a jump set, and G is a jump map, described as:

$$F(z) := -x + x_0 + x_{\Delta} q, \qquad G(z) := \begin{pmatrix} 1 - q \\ x \end{pmatrix},$$

$$C_0 := \{x : x > 19\}, \qquad D_0 := \{x : x \leq 19\},$$

$$C_1 := \{x : x < 20\}, \qquad D_1 := \{x : x \geq 20\}.$$

As can be seen, this modeling approach is quite natural. It takes into account the continuous dynamics and the jumps.

3.2 WSN system modeling

We consider a WSN system similar to the one presented in Section 2.2, which is modeled as an HDS with continuous part of energy evolution in a set of nodes and discontinues pars of jump evolution. We show hereafter, that the hybrid representation can naturally take into account the continuous evolution of the discharge battery of the node and the discrete evolution of the node jumping in the different states.

3.2.1 System description and control objectives

Consider a WSN system similar to the one presented in Section 2.2 with $n \in N^*$ sensor nodes, $m \in N^*$ functioning modes for each node. During the WSN lifespan, some nodes S_i may fall (and exit) in the *Unreachable* condition. This *Unreachable* condition can occur because of node physical damages, strong perturbations of the radio channel or when $x_i \leq X_i^{min}$ (for more detail see Section 2.5.1). Due to the effect of this "dynamicity" (i.e. appearance/disappearance of nodes), it is mandatory to supervise the number of available nodes at every time instant in order to collect enough measurements from the application viewpoint at the supervisor side, so as to fulfill the requested mission.

The energy consumed when switching from one mode to another was supposed to be integrated in b_{il} in the model in Section 2.2. In this chapter, we consider that the switching power consumption $\delta_i^{h\to l}$ is explicitly taken into account when the node S_i switches from mode M_h to mode M_l.

The WSN system is therefore characterized by two states x_i and u_i for each node S_i and a system output y_i:

- $x_i \in \mathbb{R}_{>0}$ is the remaining energy in the node battery with the same constraint as is (2.8) on the state x_i:

$$0 < X_i^{min} \leqslant x_i \leqslant X_i^{max}. \tag{3.7}$$

  $x_i(t, j) = x_i(0, 0)$ (i.e. at $t = 0$ and $j = 0$) denotes the initial battery capacity, where t is the continuous time and j is the total number of jumps of the solution;

- y_i is the measurement of the remaining energy in the node battery delivered by each node at each sampling time kT_c to the supervisor (as in (2.9)). Note that in real-life conditions, the supervisor needs this measurement to calculate the control. Therefore, at time kT_c:

$$X_i^{min} \leqslant y_i \leqslant X_i^{max}. \tag{3.8}$$

- $u_i = \begin{bmatrix} u_{i1} & \cdots & u_{ih} & \cdots & u_{im} \end{bmatrix}^T \in \{0, 1\}^m$ denotes the control states, related to the functioning mode of the node. The components of u_i are equal to 1 or 0. Thus, $u_i = e_h$ means that S_i is in mode M_h. e_h is the h^{th} column of the identity matrix, or the h^{th} vector of the Euclidean basis. $u_i = \mathbf{0}_m$ denotes the *Unreachable* condition of node S_i. $u_i(0, 0)$ denotes the initial control state. The values of $u_i(0, 0)$ are chosen so as to take into account the mission (described hereafter) and to penalize the sensor nodes with smaller lifespan. Remember that the lifespan of the reachable sensor node S_i at time t is equal to

$$L_i = \frac{x_i - X_i^{min}}{B_i u_i}. \tag{3.9}$$

Basically, the sensor node S_i is considered reachable if the supervisor has received information on its remaining energy y_i at time kT_c. Therefore (3.9) becomes at time kT_c:

$$L_i = \frac{y_i - X_i^{min}}{B_i u_i}. \tag{3.10}$$

Moreover the node S_i is characterized by:

- a power consumption (line) vector $B_i \in \mathbb{R}^m_{>0}$. The component b_{ih} of B_i denotes the power consumed by S_i when operating in mode M_h, that is when $u_i = e_h$. Note that in the system model for MPC strategy, the component b_{ih} denotes the energy consumed by S_i in mode M_h. From a practical viewpoint, the components of B_i are assumed unequal which is consistent with what is observed for today commercial nodes. Moreover, now, the consumption when the node switches from one mode to other one is not taken into account in b_{ih};

- a disturbance input $w_i \in \{0; 1\}$ that corresponds to the ability for the node S_i to harvest energy (as in Section 2.2). This disturbance input cannot be controlled but may be predicted in some situation. $w_i = 1$ (resp. $w_i = 0$) is associated to the capability for the harvesting system to harvest (resp. not to harvest) energy from the environment;

- the harvested power value E_i corresponds to the amount of power provided by the harvesting system of node S_i (as in Section 2.2). Note that E_i is in essence a time-variant value in real-life conditions. E_i may be in some situations predicted or even measured;

- the switching power consumption $\delta_i^{h\to l} \in \mathbb{R}_{>0}$ between two functioning modes taken into account the fact that switching node S_i from mode M_h to mode M_l has an energy cost. Moreover, $\delta_i^{h\to h} = 0$;

- an exogenous input $\alpha_i \in \{0, 1\}$ that is related to the *Unreachable* condition defined above. α_i is equal to 0 (resp. 1) if S_i is unreachable (resp. reachable). In the sequel, we consider $y_i = \alpha_i x_i$. Thus, if node is unreachable, $y_i = 0$ which is inconsistent with the bounds in (3.8), meaning that this measurement has not been received by the supervisor.

Remark 2. *In the real-life conditions, we know neither the remaining energy level y_i nor the actual consumption of an unreachable node. However, when it comes back, i.e. it is again reachable by the supervisor, it sends the measurements of its remaining energy so that the control strategy will be able to consider this node as a potential node to be placed in a functioning mode.*

The control objectives are to extend the WSN lifespan by reducing the overall power consumption of the nodes via an appropriate management of the functioning mode of each node while fulfilling a given mission for the whole set of nodes. During the continuous-time evolution of each solution $(\mathbf{x}, \mathbf{u})$, we must fulfill the mission defined as:

$$\sum_{i=1}^{n} u_i^T e_h = d_h, \tag{3.11}$$

where $d_h \in \mathbb{N}^*$ defines the exact number of nodes in mode M_h. For instance, d_1 defines the number of nodes in the *Active* mode. This information can be seen as an external reference.

3.2.2 Hybrid representation and pairwise jump rules

Within the above setting, the flow dynamics of the n nodes are given by:

$$\left\{ \begin{array}{lcll} \dot{x}_i & = & -B_i u_i + E_i w_i, & \\ y_i & = & \alpha_i x_i, & i = 1, ..., n, \qquad (\mathbf{x}, \mathbf{u}, \mathbf{w}, \alpha, \mathbf{y}) \in \mathcal{C}, \\ \dot{u}_i & = & \mathbf{0}^m, & \end{array} \right. \tag{3.12}$$

where the flow set $\mathcal{C}$ is to be designed. The state dynamics $\mathbf{x}$ can be compactly written as:

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}}(t) & = & -B\mathbf{u}(t) + E\mathbf{w}(t), \\ \mathbf{y}(t) & = & \alpha(t)\mathbf{x}(t), \\ \dot{\mathbf{u}}(t) & = & \mathbf{0}^{nm}, \end{array} \right. \tag{3.13}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T \in \mathbb{R}^n_{>0}$, $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}^T \in \{0,1\}^{nm}$, $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T \in \{0,1\}^n$, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T \in \mathbb{R}^n_{>0}$, $\alpha = \text{diag}(\alpha_1, \alpha_2, ..., \alpha_n) \in \{0,1\}^{n \times n}$, $B = \text{diag}(B_1, B_2, \cdots, B_n) \in \mathbb{R}^{n \times nm}$, and $E = \text{diag}(E_1, E_2, \cdots, E_n) \in \mathbb{R}^{n \times n}$.

The jump dynamics of the n nodes comprise the possibility that the available (reachable) nodes autonomously decide to swap their respective role within the network. Given (3.12), one readily understands that swapping role means simply swapping the values of u_i. Then, we may define the sets $\mathcal{D}_{il}$ to provide conditions under which two nodes S_i and S_l are required to swap their roles, under the straightforward assumption that $\mathcal{D}_{il} = \mathcal{D}_{li}$, so that swapping is simultaneously enabled from both sides.

The adopted paradigm intrinsically defines a distributed scheduling paradigm, as long as one restricts the sets $\mathcal{D}_{il}$ to be non-empty (therefore relevant for the *potential swap* evaluation) only for pairs (i, l), belonging to the edges of a suitable undirected interconnection graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ characterizing the nodes.

In the general case, the following sets will be designed:

$$\mathcal{D}_{il} = \mathcal{D}_{li},\ \forall\ nodes\ S_i, S_l :\ (i,l) \in \mathcal{E}, \tag{3.14}$$

where $\mathcal{E}$ is the set of all edges in the interconnection graph. Based on the sets in (3.14), which will be designed in Section 3.3, we can represent the swapping as an instantaneous update of the states of the nodes S_i and S_l, corresponding to:

$$\begin{cases} \begin{bmatrix} x_i^+ \\ x_l^+ \end{bmatrix} = \begin{bmatrix} x_i - (u_i^+)^T \Delta_i u_i \\ x_l - (u_l^+)^T \Delta_l u_l \end{bmatrix}, \\ \qquad\qquad\qquad\qquad (\mathbf{x},\mathbf{u}) \in \mathcal{D}_{il}, \\ \begin{bmatrix} u_i^+ \\ u_l^+ \end{bmatrix} = \begin{bmatrix} u_l \\ u_i \end{bmatrix}, \end{cases} \tag{3.15}$$

with the switching consumption matrix Δ_i defined as:

$$\Delta_i = \begin{bmatrix} 0 & \delta_i^{2\to 1} & \cdots & \delta_i^{h\to 1} & \cdots & \delta_i^{m\to 1} \\ \vdots & & \ddots & & & \vdots \\ \delta_i^{1\to m} & \delta_i^{2\to m} & \cdots & \delta_i^{h\to m} & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m\times m}. \tag{3.16}$$

Equation (3.15) only indicates the instantaneous swap for a pair of nodes, but for a complete description of the dynamics, we should also specify that across those jumps all other nodes S_i, $i = 1, ..., n-2$ do not experience any change of their x_i and u_i states. In particular, the relations in (3.15) may be compactly represented as:

$$\begin{bmatrix} \mathbf{x}^+ \\ \mathbf{u}^+ \end{bmatrix} = \begin{bmatrix} g_x^{il}(\mathbf{x},\mathbf{u}) \\ g_u^{il}(\mathbf{x},\mathbf{u}) \end{bmatrix} =: g_{il}(\mathbf{x},\mathbf{u}), \quad (\mathbf{x},\mathbf{u}) \in \mathcal{D}_{il}, \tag{3.17}$$

where $g_{il} : (\mathbb{R}\times\{0,1\}^m)^n \to (\mathbb{R}\times\{0,1\}^m)^n$ can be straightforwardly expressed from (3.15). With the pairwise rules in (3.17) associated to a jump set $\mathcal{D}_{sw}$ corresponding to the set where at least one pair of nodes is ready for a swap operation (namely $\mathbf{x} \in \mathcal{D}_{sw}$ if $\mathbf{x} \in \mathcal{D}_{il}$ for at least one pair i, l):

$$\mathcal{D}_{sw} = \bigcup_{i\neq l} \mathcal{D}_{il}, \quad G_{sw}(\mathbf{x},\mathbf{u}) = \bigcup_{il:\ (\mathbf{x},\mathbf{u})\in\mathcal{D}_{il}} g_{il}(\mathbf{x},\mathbf{u}), \tag{3.18}$$

where, by construction, G_{sw} is a set-valued mapping (multiple pairs may be ready to swap at the same time) that possesses the useful property of having a closed graph because its graph is the union of the (closed) graphs of g_{il}.

A different type of jump must be defined to take into account the presence of the exogenous input α, that captures the possibility for nodes to fall into the *Unreachable* condition, and of the dynamically changed mission during the system evolution.

Thus, for each node, four new jump rules are defined. The first one corresponds to the situation when the node S_i is in the *Active* mode (e.g., mode M_1 with $u_i = e_1$) or in the *Standby* one (e.g., mode M_2 with $u_i = e_2$) and the corresponding α_i goes to zero (the node becomes unreachable). Then, the input u_i is automatically switched to the *Unreachable* value, i.e. $u_i = \mathbf{0}_m$:

$$\begin{cases} x_i^+ = x_i \\ u_i^+ = \mathbf{0}_m, \end{cases} \quad (\mathbf{x}, \mathbf{u}) \in \mathcal{D}_0^i, \tag{3.19a}$$

$$\mathcal{D}_0^i := \{(\mathbf{x}, \mathbf{u}, \mathbf{y}) : \ (u_i = e_1 \text{ or } u_i = e_2) \text{ and } y_i \leq X_i^{min}\}, \tag{3.19b}$$

$$\mathcal{D}_0 := \bigcup_{i=1,\ldots,n} \mathcal{D}_0^i, \quad G_0(\mathbf{x}, \mathbf{u}) = \bigcup_{i:\ (\mathbf{x},\mathbf{u}) \in \mathcal{D}_0^i} g_0^i(\mathbf{x}, \mathbf{u}). \tag{3.19c}$$

In this case, note that the supervisor did not receive information related to the remaining energy y_i for the unreachable node ($\alpha = 1$). $g_0^i(\mathbf{x}, \mathbf{u})$ has a similar structure to functions $g_{il}(\mathbf{x}, \mathbf{u})$ introduced in (3.17). $g_0^i(\mathbf{x}, \mathbf{u}) : (\mathbb{R} \times \{0, 1\}^m)^n \to (\mathbb{R} \times \{0\}^m)^n$ is a compact way of representing the jump map in (3.19a) essentially comprising identities in all entries except for those related to node S_i.

The second jump rule corresponds to a more sophisticated action ensuring that the mission (e.g. d_1 active nodes, see the mission definition in (3.11)) is fulfilled even when an active node becomes unreachable. The second jump rule forces any reachable node (without any pre-specified priority) to become active if the mission is no more satisfied. This will typically happen if an active node falls in the *Unreachable* condition because of a jump arising from (3.19). This jump rule is given by:

$$\begin{cases} x_i^+ = x_i \\ u_i^+ = e_1, \end{cases} \quad (\mathbf{x}, \mathbf{u}) \in \mathcal{D}_1^i, \tag{3.20a}$$

$$\mathcal{D}_1^i := \{(\mathbf{x}, \mathbf{u}, \mathbf{y}) : \ u_i = e_2 \text{ and } \sum_{i=1}^{n} \alpha_i u_i^T e_1 \leq d_1 - 1\}, \tag{3.20b}$$

$$\mathcal{D}_1 := \bigcup_{i=1,\ldots,n} \mathcal{D}_1^i, \quad G_1(\mathbf{x}, \mathbf{u}) = \bigcup_{i:\ (\mathbf{x},\mathbf{u}) \in \mathcal{D}_1^i} g_1^i(\mathbf{x}, \mathbf{u}), \tag{3.20c}$$

where $g_1^i(\mathbf{x}, \mathbf{u}) : (\mathbb{R} \times \{0, 1\}^m)^n \to (\mathbb{R} \times \{0, 1\}^m)^n$ is a compact way of representing the jump map in (3.20a) essentially comprising identities in all entries except for those related to node i. Basically, $g_1^i(\mathbf{x}, \mathbf{u})$ has a similar structure to functions g_{il} introduced in (3.17).

The third jump rule corresponds to the situation when node S_i becomes

reachable with the ability to jump to *Standby* mode:

$$\begin{cases} x_i^+ = [X_i^{min}, X_i^{max}] \\ u_i^+ = e_2, \end{cases} \quad (\mathbf{x}, \mathbf{u}) \in \mathcal{D}_2^i, \tag{3.21a}$$

$$\mathcal{D}_2^i := \{(\mathbf{x}, \mathbf{u}, \mathbf{y}) : \ u_i = \mathbf{0}_m \text{ and } y_i \geq X_i^{min}\}, \tag{3.21b}$$

$$\mathcal{D}_2 := \bigcup_{i=1,\dots,n} \mathcal{D}_2^i, \quad G_2(\mathbf{x}, \mathbf{u}) = \bigcup_{i:\ (\mathbf{x},\mathbf{u}) \in \mathcal{D}_2^i} g_2^i(\mathbf{x}, \mathbf{u}), \tag{3.21c}$$

where $g_2^i(\mathbf{x}, \mathbf{u}) : (\mathbb{R} \times \{0\}^m)^n \to (\mathbb{R} \times \{0, 1\}^m)^n$ is a compact way of representing the jump map in (3.21a) essentially comprising identities in all entries except for those related to node i. $g_2^i(\mathbf{x}, \mathbf{u})$ has a similar structure to functions g_{il} introduced in (3.17). In this case, when a node becomes reachable ($\alpha = 1$), it sends its measured renaming energy $y_i = x_i$ to the supervisor, where x_i is found in $[X_i^{min}, X_i^{max}]$. Note that the mission is not more fulfilled, after the jump (3.21), a node S_i will switch in the *Active* mode as defined in (3.20).

The fourth jump rule express to the situation when the mission changes dynamically, i.e. $d_1(t_1) > d_1(t_2)$, where $t_1 < t_2$. In this case, node S_i jumps to the *Standby* mode from the *Active* one:

$$\begin{cases} x_i^+ = x_i \\ u_i^+ = e_2, \end{cases} \quad (\mathbf{x}, \mathbf{u}) \in \mathcal{D}_3^i, \tag{3.22a}$$

$$\mathcal{D}_3^i := \{(\mathbf{x}, \mathbf{u}, \mathbf{y}) : \ u_i = e_1 \text{ and } \sum_{i=1}^{n} u_i^T e_1 \leq d_1 - 1\}, \tag{3.22b}$$

$$\mathcal{D}_3 := \bigcup_{i=1,\dots,n} \mathcal{D}_3^i, \quad G_3(\mathbf{x}, \mathbf{u}) = \bigcup_{i:\ (\mathbf{x},\mathbf{u}) \in \mathcal{D}_3^i} g_3^i(\mathbf{x}, \mathbf{u}), \tag{3.22c}$$

where $g_3^i(\mathbf{x}, \mathbf{u}) : (\mathbb{R} \times \{0\}^m)^n \to (\mathbb{R} \times \{0, 1\}^m)^n$ is a compact way of representing the jump map in (3.22a) essentially comprising identities in all entries except for those related to node i. It has a very similar structure to function g_{il} introduced in (3.17). In the case, where $d_1(t_1) < d_1(t_2)$ with $t_1 < t_2$, the jump rule (3.20) is executed.

For these jumps rules (3.17)-(3.22), the energy in the battery of node S_i does not experience any discontinuity, thus $x_i^+ = x_i$, where $x_i \in [X_i^{min}, X_i^{max}]$. Remember that if $x_i = X_i^{min}$ and $w_i = 0$, the node S_i has the lifespan $L_i = 0$ (see (3.9)), therefore it is considered as a unreachable node.

With the definitions in (3.18), (3.19c), (3.20c), (3.21c), (3.22c), the complete jump dynamics can be compactly represented as:

$$\begin{bmatrix} \mathbf{x}^+ \\ \mathbf{u}^+ \end{bmatrix} \in G(\mathbf{x}, \mathbf{u}), \quad (\mathbf{x}, \mathbf{u}, \mathbf{y}) \in \mathcal{D}, \tag{3.23}$$

where $\mathcal{D} = \mathcal{D}_{sw} \cup \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, and $G = G_{sw} \cup G_0 \cup G_1 \cup G_2 \cup G_3$ is a set-valued map. The discrete-time dynamics (3.23) together with (3.12) correspond to a hybrid description of the WSN system under the specific control action.

Now, we focus on the solutions to (3.12) and (3.23), insisting that they evolve in a specific set $\mathcal{O}$ where the remaining energy in the battery of each node is positive for all nodes, and the input vector $\mathbf{u}$ has components equal to 0 or 1. Within this set, the flow set $\mathcal{C}$ is the closed complement of the flow set $\mathcal{D}$ relative to $\mathcal{O}$. More specifically:

$$\mathcal{O} = \mathbb{R}^n_{>0} \times \{0,1\}^{nm}, \quad \mathcal{C} = \left(\overline{\mathcal{O} \setminus \mathcal{D}}\right) \bigcap \mathcal{O}. \tag{3.24}$$

Within the set $\mathcal{O}$, due to the positivity of the entries in the vectors B_i and the constraints on the state values x_i, it is evident that all solutions will be bounded and not complete (their domain is bounded). Thus, the objective is to design the jump sets $\mathcal{D}_{il}$ in an intuitive way, the goal being to maximize the length of the solution domain in the ordinary time direction, i.e. the WSN lifespan, until d_1. The WSN lifespan is named hereafter the *lifespan of the solution*.

We focus on the solutions to (3.12) and (3.23), insisting that they evolve in a specific set $\mathcal{O}$ where the remaining energy in the battery of each node is positive for all nodes, and the input vector $\mathbf{u}$ has components equal to 0 or 1. Within this set, the flow set $\mathcal{C}$ is the closed complement of the flow set $\mathcal{D}$ relative to $\mathcal{O}$. More specifically:

$$\mathcal{O} = \mathbb{R}^n_{>0} \times \{0,1\}^{nm}, \quad \mathcal{C} = \left(\overline{\mathcal{O} \setminus \mathcal{D}}\right) \bigcap \mathcal{O}. \tag{3.25}$$

Within the set $\mathcal{O}$, due to the positivity of the entries in the vectors B_i and the constraints on the state values x_i, it is evident that all solutions will be bounded and not complete (their domain is bounded). Thus, the objective is to design the jump sets $\mathcal{D}_{il}$ in an intuitive way, the goal being to maximize the length of the solution domain in the ordinary time direction, i.e. the WSN lifespan, until d_1. The WSN lifespan is named hereafter the *lifespan of the solution*.

3.3 Design of the scheduling law

Now, the scheduling law will be designed for the simple case with two nodes and without harvesting systems. For this case, the optimality of the result will be proved. Then, the strategy based on a HDS approach will be extended to the case with $n > 2$ nodes and with harvesting systems.

3.3.1 Fault-free case with two nodes

In the previous section, the WSN power management has been expressed as a control problem that decides the functioning mode for each node, to the design of the pairwise sets $\mathcal{D}_{il}$. Indeed, dynamics (3.12), (3.23) already clarify what is happening during the flowing (i.e. when the batteries are discharging) and what should happen at each reconfiguration of the network. Basically, when the state

$(\mathbf{x}, \mathbf{u})$ belongs to $\mathcal{D}_{il}$, the nodes S_i and S_l swap their roles. To suitably design $\mathcal{D}_{il}$ for all $i \neq l$, the adopted control paradigm focuses on a pairwise reconfiguration rule.

This rule is described for the simplified case of two nodes S_1, S_2 without harvesting systems ($\mathbf{w}(t) = \mathbf{0}^n$) and two modes M_1, M_2 (resp. *Active* and *Standby*), with $\mathcal{D}_{12} = \mathcal{D}_{21} = \mathcal{D}$. For this specific case, an optimal result is proved hereafter. Moreover, in this case, it does not make sense to have an unreachable node because no replacement node is available. Therefore, we focus on the fault-free two-nodes case, with $\alpha_i = 1 \; \forall \; i = 1, 2$. Moreover, $d_1 = 1$.

The rationale behind the choice of $\mathcal{D}_{12}$ is that we would like to design an algebraic condition on $\mathbf{x}$ and $\mathbf{u}$ that expresses when it is convenient to swap roles between both nodes in such a way to maximize the *lifespan of the solution*. This quantity may be expressed as a cost function J to be maximized. When designing $\mathcal{D}_{12}$, the expected lifespan if the solution does not perform further jumps:

$$J(x, u) = \min_{i:u_i \neq \mathbf{0}^m} \frac{y_i - X_i^{min}}{B_i u_i} \tag{3.26}$$

is introduced. A first condition to encode in the flow set is that it is not convenient to jump (or swap roles) whenever $J(x^+, u^+) < J(x, u)$. Thus, one must ensure that:

$$\begin{aligned} J^+(x, u) := J(g_x(x, u), g_u(x, u)) < J(x, u) \\ \Rightarrow \; (x, u) \notin \mathcal{D}_{12}. \end{aligned} \tag{3.27}$$

Intuitively, the condition (3.27) means that no switch is performed if it reduces the lifespan. Note that this condition is a function of both x and u.

Even though condition in (3.27) is reasonable, it may still induce undesirable behaviours in some cases. For instance, consider two identical nodes, namely $\Delta_1 = \Delta_2$, $B_1 = B_2$. The *Active* (resp. *Standby*) mode is supposed associated with a large (resp. small) power consumption. Assume also that Δ_i are relatively small when compared to the power consumption of the *Active* mode. In this case, the best strategy is clearly to keep one node in the *Active* mode until its battery is drained, and then swap once and only once along the solution. However, picking $\mathcal{D}_{12}$ as the closed complement of the left hand condition of (3.27) will enforce extra unnecessary jumps as soon as the energy level in the active node becomes sufficiently small compared to the energy in the other node, and then *vice-versa*, and so on.

One way to avoid this situation is to encode in $\mathcal{D}_{12}$ another condition: nodes will not swap if there is no "emergency" to do so. Indeed, when the left hand condition in (3.27) holds, solutions will still keep flowing (thus, no switch is applied) if waiting any further does not cause any reduction in the expected lifespan after the potential switch. To characterize this reduction, denote by i^*

the index (or the set of indexes) that minimizes J^+, namely:

$$i^* = \arg\min_{i=1,2} \frac{x_i^+}{B_i u_i^+} = \arg\min_{i:u_i \neq \mathbf{0}^m} \frac{y_i - u_{3-i}^T \Delta_i u_i}{B_i u_{3-i}}. \tag{3.28}$$

Note that i^* may contain both indexes if the two expected lifespans coincide. Then, characterize the reduction as the derivative of J^+ during flow (if no jump would be performed):

$$\delta J(x,u) = \min_{l \in i^*} \frac{\dot{x}_l^+}{B_l u_l^+} = \min_{l \in i^*} \frac{-B_l u_l}{B_l u_{3-l}}. \tag{3.29}$$

Note that the derivative of $u_{3-i}^T \Delta_i u_i$ is zero along the flow. The function in (3.29) captures the idea of how damageable it is to postpone the swap. In other words (3.29) expresses how much smaller J^+ will become if the solution flows for some extra time, before jumping. Since the two components in B_i are supposed not equal, then $\delta J(x,u) \neq -1$.

Clearly, if we keep flowing, the decrease rate of J will simply be 1 as the lifespan decreases linearly as time flows. Therefore, one extra criterion for the selection of $\mathcal{C}$ may be:

$$\delta J(x,u) \geq -1 \quad \Rightarrow \quad (x,u) \notin \mathcal{D}_{12}, \tag{3.30}$$

which is well defined, as indicated above, because $\delta J(x,u) \neq -1$. In particular, what happens along solutions, as long as $J^+ \geq J$ is that $\delta J(x,u) > -1$ whenever there is no urge to jump, and then once the "argmin" in (3.28) changes, we start getting $\delta J(x,u) \leq -1$ and a jump occurs.

To summarize, one selects:

$$\mathcal{D}_{12} = \{(x,u) : \ J^+(x,u) \geq J(x,u) \text{ and } \delta J(x,u) \leq -1\}. \tag{3.31}$$

Note that the definition is commutative that is, there is not specific role of nodes S_1 and S_2 in the selection of $\mathcal{D}_{12}$. The selection (3.31) enables to prove the following optimality result. Notice that hereafter $X_i^{min} = 0 \leq x_i \leq X_i^{max}$. If this is not the case, a simple change in the variable is performed.

Theorem 1. *Assume that B_i, $i = 1,2$, has positive and distinct components and that matrices Δ_i, $i = 1,2$ have strictly positive off diagonal elements. Consider any initial condition with positive values of $x_i(0,0)$ and with $u_i(0,0)$, $i = 1,2$ being two independent columns of the identity matrix. The solution of* (3.12), (3.23), (3.31) *is unique and it has a lifespan equal to the maximum lifespan that can be obtained by selecting arbitrary jump times for* (3.12), (3.15) *starting from the same initial conditions.*

Proof. The proof is carried out by first (step 1) showing that the (optimal) solution providing the maximum lifespan performs at most one jump (one swap between both nodes), and then (step 2) showing that the solution generated by the proposed jump rule is optimal.

Step 1. Assume that the optimal solution $\varphi_{opt} = (x_{opt}, u_{opt})$, having lifespan T, performs more than one jump (or swap) and denote by t_1 and t_2 the first 2 jump times. These jump times clearly satisfy $0 \leq t_1 \leq t_2 \leq t_3$, where we denote by t_3 either the next jump time, or the lifespan time $t_3 = T$. Since T is the lifespan of the solution, we must have that $x_{opt,1}(T, j) > 0$, $x_{opt,2}(T, j) > 0$, and that at least one of them is zero, where j is the total number of jumps of the solution. Consider now another solution (x^*_{opt}, u^*_{opt}) starting from the same initial condition $(x(0,0), u(0,0)) = (x_{10}, x_{20}, u_{10}, u_{20})$ and performing one jump at time $t^*_1 = t_1 + (t_3 - t_2)$, and then performing the same jumps as the original solution does, namely $t^*_2 = t_3$, ..., $t^*_{j-1} = t_j$. Note that this new solution performs one more jump less than the original one. Then, it can be proved that at ordinary time $t^*_2 = t_3$ (and before the possible corresponding jump), both states x_1 and x_2 of the "*" solution are strictly larger than the corresponding states of the original *opt* solution. As a consequence:

$$0 < x_i(T, j) < x^*_i(T, j-1), \quad i = 1, 2, \tag{3.32}$$

which implies that the lifespan of the "*" solution is larger, thus completing the proof of this step. We only prove (3.32) for the case $i = 1$ as the case $i = 2$ is identical. To this end, consider:

$$\begin{aligned} x^*_{opt,1}(t^*_2, 1) &= x_{10} - (t^*_1 - 0)B_1u_{10} - (t^*_2 - t^*_1)B_1Su_{10} - (Su_{10})^T\Delta_1u_{10} \\ &= x_{10} - (t_1 + (t_3 - t_2))B_1u_{10} - (t_2 - t_1)B_1Su_{10} - (Su_{10})^T\Delta_1u_{10}, \end{aligned}$$

where the swaping matrix is $S := \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ and $t^*_2 - t^*_1 = t_2 - t_1$ is chosen. Instead, for the original solution we have:

$$\begin{aligned} x_{opt,1}(t_3, 2) &= x_{10} - (t_1 - 0)B_1u_{10} - (t_2 - t_1)B_1Su_{10} \\ &\quad - (t_3 - t_2)B_1S^2u_{10} - (Su_{10})^T\Delta_1u_{10} - u^T_{10}\Delta_1Su_{10} \\ &= x_{10} - (t_1 + (t_3 - t_2))B_1u_{10} - (t_2 - t_1)B_1Su_{10} \\ &\quad - (Su_{10})^T\Delta_1u_{10} - u^T_{10}\Delta_1Su_{10}, \end{aligned}$$

with $S^2 = I$. This clearly shows that the second state is smaller because it performs one extra jump and the off diagonal terms of Δ_i, $i = 1, 2$ are all positive.

Step 2. Consider now the structure of the jump set in (3.31) and note that all along any solution $\varphi(t, j) = (x(t, j), u(t, j))$, the function $\psi(t, j) := t + J(\varphi(t, j))$ denotes the envisioned lifespan (in the future) of the solution in the case when no other jump happens, while $\psi^+(t, j) := t + J^+(\varphi(t, j))$ denotes the envisioned lifespan of the solution if a single jump happens at the current time. This is easily

understandable from the fact that J and J^+ measure the remaining lifespan, so the total amount of the envisioned future flow. Then, at each hybrid time instant (t, j) in dom φ (hybrid time domain [160]), the envisioned lifespan corresponds to such an envisioned remaining lifespan plus the already elapsed time t. As an additional property, note that along any solution, the function ψ is non-decreasing. Indeed, its derivative along the flows is trivially zero, while the first condition in (3.31) implies that it is non-decreasing at jumps. Moreover, it is straightforward to conclude from (3.29) that:

$$\frac{d}{dt}\psi^+(t,j) = 1 + \delta J(\varphi(t,j)), \tag{3.33}$$

which can never be zero because of the assumption that the two components in the matrices B_i are not equal. We now split the proof in two cases comparing the optimal solution to our solution and establishing that they have the same lifespan.

Case 1: If the optimal solution φ_{opt} performs no jumps, then the envisioned lifespan from the initial condition is already T, namely $\psi(0,0) = J(\varphi(0,0)) = J(\varphi_{opt}(0,0)) = T$. As a consequence, our solution will flow for at least (therefore exactly, from optimality) T ordinary time because the function ψ is non-decreasing.

Case 2: If the optimal solution φ_{opt} performs one jump, denote by $t^* \in [0, T)$ the time of that jump and note that it must satisfy:

$$t^* + J^+(\varphi_{opt}(t^*, 0)) = t^* + J(\varphi_{opt}(t^*, 1)) = T. \tag{3.34}$$

Denote also $\psi_{opt}(t, j) = t + J(\varphi_{opt}(t, j))$. Since the solution flows for all $(t, j) \in [0, t^*) \times \{0\}$, and since $\frac{d}{dt}\psi^+_{opt}$ can never be zero as emphasized after (3.33), it must necessarily be that $\frac{d}{dt}\psi^+_{opt}(t, 0) > 0$ for all $t \in [0, t^*)$. Otherwise there would be another solution jumping before t^* and lasting longer than φ_{opt} which is impossible due to optimality. As a consequence, since our solution φ starts from the same initial condition as φ_{opt}, it must hold, also based on (3.33), that $1 + \delta J(\varphi(0,0)) = \frac{d}{dt}\psi^+(0,0) = \frac{d}{dt}\psi^+_{opt}(0,0) > 0$, meaning, according to the second condition in (3.31), that our solution does not jump at the initial time. Due to the uniqueness of solution of our flow dynamics, we also conclude that $1+\delta J(\varphi(t,0)) > 0$ for all $t \in [0, t^*]$, thus implying that our solution behaves optimally until t^*. The interval is now closed due to continuity of the solution along the flows. Then, we have from (3.34) that $\psi^+(t^*, 0) = t^*+J^+(\varphi(t^*, 0)) = T$, which completes the proof by the non-decreasing property of ψ established above. □

3.3.2 General case

The approach of Section 3.3.1 is now extended to the case with $n > 2$ node, $m \geq 2$ modes, and the harvesting systems. Therefore, at time kT_c the measurement of

the remaining energy is equal to $y_i(k) = \alpha_i(k)x_i(k) = \alpha_i(k)(x_i(k-1) - B_iu_i(k-1) + E_iw_i(k-1))$. The hybrid dynamics in (3.12), (3.23) essentially concentrate on pairwise relations between each possible pair of reachable nodes. In particular, the $J_{il}(\mathbf{x},\mathbf{u})$, $J_{il}^{+}(\mathbf{x},\mathbf{u})$, k_{il}^{*}, $\delta J_{il}(\mathbf{x},\mathbf{u})$ quantities can be defined as those in (3.26), (3.27), (3.28), and (3.29) respectively, when focusing on nodes S_i, S_l satisfying $(i,l) \in \mathcal{E}$ as follows:

$$J_{il}(\mathbf{x},\ \mathbf{u}) := \min_{k=i,l;\ k:u_k \neq \mathbf{0}^m} \frac{y_k - X_k^{min}}{B_k u_k} \tag{3.35}$$

$$J_{il}^{+}(\mathbf{x},\ \mathbf{u}) := J_{il}(g_x^{il}(\mathbf{x},\mathbf{u}), g_u^{il}(\mathbf{x},\mathbf{u})) \tag{3.36}$$

$$k_{il}^{*} = \arg \min_{k=i,l;\ k:u_k \neq \mathbf{0}^m} \frac{y_k - u_{k^c}^T \Delta_k u_k}{B_k u_{k^c}} \tag{3.37}$$

$$\delta J_{il}(\mathbf{x},\ \mathbf{u}) = \min_{k \in k_{il}^{*}} \frac{-B_k u_k}{B_k u_{k^c}}, \tag{3.38}$$

where k^c represents the "second" node in a pair, namely $k^c = i$ if $k = l$, and *vice-versa*. In (3.38), we do not consider the part E_kw_k of the harvesting systems to avoid many possible switches when the energy is harvested periodically, i.e. the behavior of the harvesting system is reflected as the oscillations.

With the above definitions, the jump sets $\mathcal{D}_{il}$ can now be defined by suitably generalizing the expression in (3.31):

$$\begin{aligned} \mathcal{D}_{il} = \{(\mathbf{x},\ \mathbf{u}, \alpha, \mathbf{w}) :\ & J_{il}(\mathbf{x}^+, \mathbf{u}^+) \geq J_{il}(\mathbf{x},\ \mathbf{u}) \\ & \text{and } \delta J(\mathbf{x},\ \mathbf{u}) \leq -1 \text{ and } \alpha_i = \alpha_l = 1\}, \end{aligned} \tag{3.39}$$

where the swap is now inhibited if any of the two nodes in a pair is in the *Unreachable* condition.

With the selection (3.39), we may now express the following result, which is a straightforward consequence of the jump rules in (3.19) - (3.22) and of the fact that the jumps in (3.15) have no effect on the mission because they merely correspond to swapping the states of nodes S_i and S_l.

Proposition 1. *Given any solution to* (3.12), (3.23), (3.39), *if at least* d_1 *nodes are not in the* Unreachable *condition at each time in its domain, then the mission* (3.11) *is always satisfied during the flows of the solution.*

The proof of the proposition is omitted because it is a straightforward consequence of the definition of jumps rules and of the fact that flowing is forbidden unless the mission is accomplished.

3.4 Simulation results with the HDS strategy

To validate the theoretical results presented above, some simulation are performed to illustrate the features of the proposed control law. A scenario with two different cases (without and with harvesting system) will be discussed.

3.4.1 System description and control configuration

The proposed approach is now evaluated in simulation with $n = 6$ nodes. The benchmark is similar to the one from Section 2.5.1. Each node S_i may be placed in two functioning modes, namely the *Active* mode M_1 and the *Standby* mode M_2, already described in Section 2.5.1. The sensor nodes may also drop in the *Unreachable* condition. The sensor nodes S_3 and S_5 also possess harvesting capabilities as described in Section 2.5.2.

Note that, in simulation, we consider that the remaining energy measurement is continuous, i.e. $\mathbf{y}(t) = \mathbf{x}(t)$. The remaining energy for the sensor node S_i in the *Unreachable* condition is supposed equal to X_i^{min}. However, this assumption is not always true in real-life conditions. For instance, when the node falls in the *Unreachable* condition because the energy in the battery is too low, i.e. $x_i = X_i^{min}$, this assumption is true. However, at the time when the node falls in the *Unreachable* condition because of communication perturbations, the remaining energy is not known.

Table 3.1: Average power consumption b_{ij} of node S_i in mode M_h

Sensor node	Average power consumption in mode M_1, $[mAV]$	Average power consumption in mode M_2, $[mAV]$
S_1	34.854	5.846
S_2	36.482	6.031
S_3	36.593	6.105
S_4	36.482	6.105
S_5	36.556	6.105
S_6	33.041	5.735

The nodes are supposed to be deployed over a limited space (e.g. an office room) to control the environmental conditions (temperature, humidity). The mission is defined with $d_1 = 3$, which ensures that enough information is provided to control the room environmental conditions. For each node S_i, the components of vector B_i are given in Table 3.1. The first (resp. second) component of B_i corresponds to the mean power consumed in mode M_1 (resp. M_2). The initial state value of $\mathbf{x}$ is recalled in Table 3.2: for each S_i, it is equal to $\xi_i \cdot X_i^{max}$, where ξ_i is the energy coefficient and X_i^{max} represents the nominal battery capacity. The initial state value of $\mathbf{u}$ is chosen, taking into account the mission, to penalize the nodes with smaller lifespan. It is equal to $\mathbf{u}(0,0) = [1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0]^T$ so that $d_1 = 3$ nodes are placed in M_1. The same initialization choice is performed in Scenario 2.4 from Section 2.5.3 and in Scenario 2.6 from Section 2.6.

The switching energy consumption matrices are supposed equal for all nodes.

Table 3.2: Node battery and harvesting system characteristics

Sensor node	Battery capacity X_i^{max}, $[mAV \cdot hour]$	Harvesting availability E_i, $[mAV]$	Energy coef. ξ_i, [1]	Harvesting period, *per 24 hours*
S_1	3885	missing	1	–
S_2	3885	missing	0.8	–
S_3	3885	77.7	0.9	7h-12h
S_4	3515	missing	0.7	–
S_5	3515	99.9	1	13h-18h
S_6	8510	missing	1	–

They are given by:

$$\Delta_i = \begin{bmatrix} 0 & 0.05 \\ 0.01 & 0 \end{bmatrix} [mWh]. \tag{3.40}$$

Note that these numerical values are rough estimation as they are very difficulty to be measured.

3.4.2 Simulation results

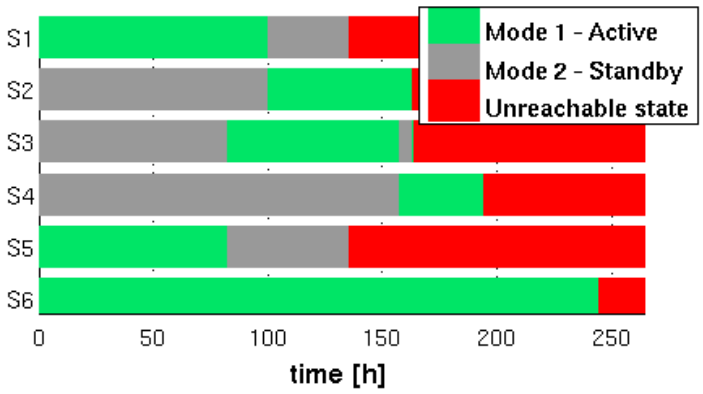
The control strategy based on the HDS approach is evaluated in the MATLAB/Simulink environment using the HyEQ Toolbox [164]. Two situations are considered, namely without and with harvesting capability for nodes S_3 and S_5.

Scenario 3.1: power management strategy based on the HDS approach

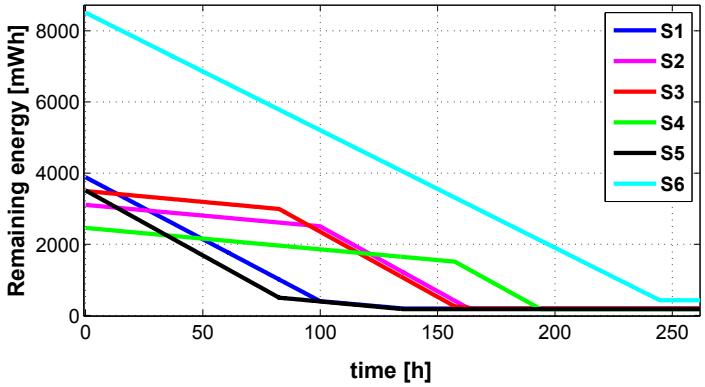
Figure 3.2(a) shows the evolution of the functioning mode for each node when the HDS control is applied. The harvesting capability of nodes being not considered here, $\mathbf{w}$ in (3.13) is equal to $\mathbf{0}_n$. The energy in the battery of each node is given in Figure 3.2(b). We can observe that the remaining energy of all nodes is never equal to 0 (as in the Scenarios 2.1-2.6 discussed in Section 2.5.3). Basically, this is related to the constraints (3.7) on the state value x_i, that prevents to fully drain the battery. As can be seen, the mission (3 nodes in mode M_1) is fulfilled approximately until 164 *hours*, that represents the WSN lifespan.

In this case, the nodes jump between the mode M_1 and mode M_2 following the rule (3.15). When a sensor node has more energy, it falls in the *Unreachable* condition following the rule (3.19).

Figure 3.3(a) and Figure 3.3(b) show the same scenario, but with harvesting systems. The WSN lifespan is now equal to $\approx$ 266 *hours*. Nodes S_3 and S_5 possess the harvesting systems with harvesting profile given in Figure 2.7. Thus, the evolution of their remaining energy depicted in Figure 3.3(b), presents oscil-



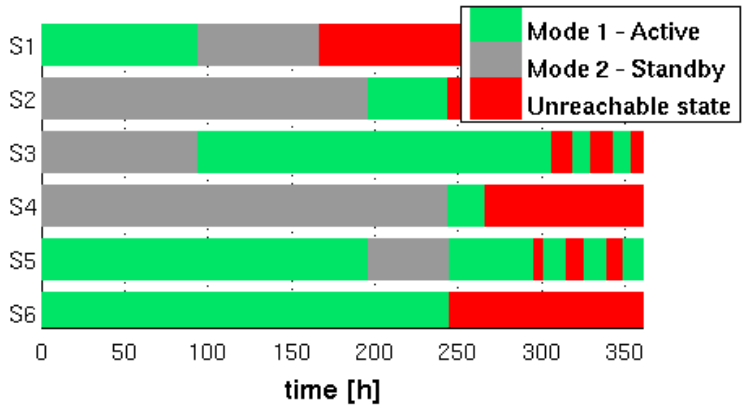
(a) Evolution of the functioning modes of the nodes



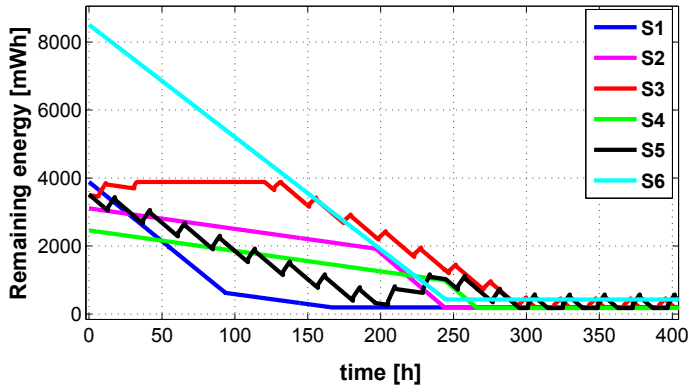
(b) Evolution of the remaining energy in each node battery

Figure 3.2: Scenario 3.1: power management strategy based on HDS, without harvesting systems

lations. Here, the nodes jump following the same rules that in a previous case for the same situation. To jump from the *Unreachable* condition to *Standby*, a node follows the rule (3.21). To jump from *Unreachable* condition to *Active* mode, when the mission is not fulfilled, a node follows the rules (3.21) and (3.20).



(a) Evolution of the functioning modes of the nodes



(b) Evolution of the remaining energy in each node battery

Figure 3.3: Scenario 3.1: power management strategy based on HDS, with harvesting systems

Comparison of the results and discussion

A summary of Scenario 3.1 with the "dummy" and basic scenarios from Section 2.5.3 is show in Table 3.3.

The simulation results of the control strategy based on the HDS approach are promising. The WSN lifespan obtained using the strategy based on a HDS approach is 1.28 longer without harvesting systems and 1.39 larger with harvesting systems that the on of the basic scenario. The comparison with the "dummy" scenario shows an increase by 1.80 without harvesting systems and by 1.39 with harvesting systems of the WSN lifespan with the HDS approach.

The number of switches in Scenario 3.1 is not large than 7 switches without

Table 3.3: Comparison of scenarios

Scenarios	WSN lifespan without harv. sys., [*hours*]	WSN lifespan with harv. sys., [*hours*]	Number of switches, [1]
Scenario 2.1: dummy	91	192	0
Scenario 2.2: basic	128	192	3
Scenario 3.1: HDS	164	266	7 (5)

harvesting systems and 5 switches with harvesting systems. In some conditions and with the corresponding nodes (the nodes that can be awakened by the request of supervisor, see Section 1.3), that allows further reduce the energy consumption and not to increase the disturbance of radio channel, because the supervisor rarely sends the control to sensor nodes.

In simulation, the conditions to switch are verified continuously using the measurements of the remaining energy in the battery of each node. Actually, we consider that the remaining energy in the batteries is measured continuously, and known on the supervisor side. This hypothesis is obviously not true in real-life conditions. Therefore, we will have to discretize the control strategy and impose that switching conditions are verified at each decision time T_c to be able to implement the HDS control strategy in real life.

TheHDS approach is not complex from a computational point of view. Thus, running this control algorithm is not consuming. As a consequence, it can simply be embedded in a node used as supervisor with relatively small computational capabilities.

3.5 Conclusion

The conception of an energy management strategy for a WSN based on a HDS approach has been proposed in this chapter. The WSN system, similar to the one in Chapter 2, is modeled as a hybrid dynamic system with a set of jumps rules.

This scheduling law fixes the mode of each node at time t so as to fulfill the so-called mission. The scheduling law is first designed in the simple case with 2 nodes, 2 modes (and no *Unreachable* condition) and without harvesting systems. In this situation, the solution is proved to be unique with maximum lifespan for the WSN. Then, the scheduling law is extended to the case with more than 2 nodes and more than 2 modes, taking into account the unreachability of some nodes and the harvested energy.

Simulation results of the proposed control strategy have been compared with

the "dummy"and basic scenarios presented in Chapter 2. The comparison shows that this HDS strategy is promising in terms of the WSN lifespan increase. Indeed, for a particular scenario, the WSN lifespan can by extended by a factor of 1.28 without harvesting systems and by a factor of 1.39 with harvesting systems compared to the basic situation from Scenario 2.2.

Moreover, the benefit of the strategy based on a HDS approach is its low computational complexity, because it only requires the evaluation of few algebraic equations. As a consequence, running the HDS control strategy itself will consume a very small amount of energy, especially when compared to the MPC approach from Chapter 2. This HDS strategy naturally takes into account the switching cost which allows to have a more accurate system model when compared to the model we used in Chapter 2.

The HDS control objective is to extend the WSN lifespan while fulfilling the mission. The control law chooses the nodes with the larger lifespan and places them in *Active* mode. Its main interest is that it does not enforce nodes to swap their role if it is not absolutely mandatory. As a consequence, the number of switches can be minimized, which in turn, saves the switching energy cost. However, this HDS strategy must be adapted for an implementation in real life. Here, it is supposed that the remaining energy in the battery of each node is continuously monitored. Therefore, the supervisor continuously knows how much energy is available in each reachable node and it can decide at any time when nodes have to swap their role. This aspect will be discussed in Chapter 4 where the HDS strategy is implemented on a real benchmark.

Chapter 4

Comparison of Both Approaches and Validation on a Test-Bench

Control strategies based on MPC and HDS approaches have been proposed to improve the WSN lifespan of a set of sensor nodes while meeting requirements imposed by the application, these requirements being expressed in terms of the number of nodes that must provide measurements to a supervisor. In the previous chapters, we have validated both approaches in simulation. We will now compare these strategies thanks to the results of scenarios 2.1-2.6 and 3.1.

The results led us to consider an implementation of the proposed strategies. For that purpose, we first need to analyze the simulation results of all scenarios presented in Chapters 2 and 3 to choose the more promising ones. The assumptions for both control techniques must be discussed to guess if the implementation on a real life can be achieved and that what is needed for that.

The test-bench hardware is discussed especially, we try to see if the sensor nodes at bond are representative of today sensor networks available from the industry. The implementation results drive to the final conclusions and validations. The results presented in this chapter have been published in [118, 165].

4.1 Comparison of MPC and HDS approaches in simulation

The proposed energy management strategies based on MPC and HDS approaches are validated in simulation in Chapters 2 and 3. Remember that the scenarios presented in these chapters are:

- Scenario 2.1: situation without control at application level ("dummy" situation);
- Scenario 2.2: management of the node activity to fulfill the mission based on a finite-state automaton, i.e. there are any d_1 sensor nodes that are active (basic situation);
- Scenario 2.3: energy management strategy based on the MPC approach and solved with MIQP, $Q = \mathbf{0}_{6\times 6}$ and $R = B_{cons}^T \times B_{cons}$;
- Scenario 2.4: energy management strategy based on the MPC approach and solved with MIQP, $Q = diag(\frac{1}{X_1^{max}}, ..., \frac{1}{X_6^{max}})$ and $R = B_{cons}^T \times B_{cons}$;
- Scenario 2.5: energy management strategy based on the MPC approach when a MILP problem is solved, $c = [b_{11}\ b_{12}\ \cdots\ b_{62}]^T$;
- Scenario 2.6: energy management strategy based on the MPC approach when a MILP problem is solved, with $c = [\frac{b_{11}}{x_1+E_1w_1-X_1^{min}}\ \frac{b_{12}}{x_1+E_1w_1-X_1^{min}}\ \cdots\ \frac{b_{62}}{x_6+E_6w_6-X_6^{min}}]^T$
- Scenario 3.1: power management strategy based on the HDS approach.

A summary of these scenarios is shown in Table 4.1.

Table 4.1: Comparison of scenarios in terms of WSN lifespan and number of switches

Scenarios	WSN lifespan without harv. sys., [*hours*]	WSN lifespan with harv. sys., [*hours*]	Number of switches, [1]
Scenario 2.1: dummy	91	192	0
Scenario 2.2: basic	128	192	3
Scenario 2.3: MIQP	178	244	3
Scenario 2.4: MIQP	175	287	$\approx 10^2$
Scenario 2.5: MILP	178	244	3
Scenario 2.6: MILP	171	284	$\approx 10^2$
Scenario 3.1: HDS	164	266	7 (5)

The simulation results of the control strategies based on MPC and on HDS approaches are promising when the WSN lifespan is considered. The WSN lifespan obtained

- using the MPC approach is 39% longer without harvesting systems and 50% longer with harvesting systems;
- using the HDS approach is 28% longer without harvesting systems and 39% longer with harvesting systems

compared to the basic situation from Scenario 2.2. For a specific benchmark case, the WSN lifespans obtained using the MPC approach can be longer by $4.1\% - 8.5\%$ without harvesting systems compared to the HDS one. The lifespan of WSN with harvesting systems is

- smaller by 8.3% in Scenario 2.3 (MPC approach) compared with Scenario 3.1 (HDS approach);
- longer by 7.9% in Scenario 2.4 (MPC approach) compared with Scenario 3.1 (HDS approach).

This phenomenon can be explained by the characteristics of the problems that are resolved in different scenarios. For instance, in scenarios 2.4, 2.6 and 3.1 the remaining energy is taken into account in the problem cost unlike other ones, that better manages the set of nodes with harvesting systems.

The number of switches depends on the control strategy and on the considered scenario. The strategy based on MPC does not try to limit the number of switches compared to the strategy based on HDS, where a switch is enforced only if it significantly extended the WSN lifespan. As a consequence, we observe a large (in the range of 10^2) number of switches in Scenario 2.4 and Scenario 2.6. Note that the switching cost is not explicitly taken into account in our MPC strategy. If we consider that the energy cost of the switch from mode M_1 to mode M_2 (resp. from mode M_2 to mode M_1) is equal to $\delta_i^{1\to 2} = 0.01\ mWh$ (resp. $\delta_i^{2\to 1} = 0.05\ mWh$), the energy cost of 10^2 switches is approximately equivalent to 1 *hour* of the WSN lifespan. In real-life conditions, the values $\delta_i^{1\to 2}$ and $\delta_i^{2\to 1}$ depend on the quality of the radio link (related to the QoS) between the supervisor and the node S_i. The number of switches in Scenario 3.1 is much less important than in Scenarios 2.4 and 2.6. Indeed, we have 7 switches without harvesting systems and 5 with harvesting systems. The reduction in the number of switches allows further to reduce the energy consumption (especially the one due to the switches) and not to increase the overload of radio channel, because the number of communications also decreases. In fact, if we need to swap roles between nodes, extra message must be sent from the supervisor to the node.

The problems of the energy consumption reducing, which are tackled by control strategies, have different levels of complexity. Table 4.2 shows the simulation time to solve the control problem for one control period $T_c = 1\ hour$ (one control step) for both strategies. Note that the simulated system with the MPC strategy is discrete and a difference equation is computed. However, the simulated system with the control strategy based on HDS is continuous. Thus, differential equation must be numerically integrated, and control algorithm must be discretized with sampling time $T_c = 1\ hour$. Here, the values are obtained in the Matlab environment using the *tic-toc* function. Matlab turns in one core of Intel Xeon Processor E5620 with 12 MB of cache, 2.40 GHz and 5.86 GT/s [150]. From these values, even if the evaluation is rough, it is evident that the control strategy solved with MIQP is more "demanding" compared to the problem solved with MILP. The difference between MIQP and MILP is in the range of one order of magnitude. The comparison between the HDS approach and the MPC shows that the HDS algorithm is in the range of one (resp. two) order of magnitude easier to compute compared with MILP (resp. compared with MIQP).

Table 4.2: Comparison of scenarios in terms of computational time (simulation cases)

Scenarios	Simulation duration, [$hours$]	Simulation time for 1 step, [sec]
Scenario 2.3: MIQP	178, without harv. sys.	0.369
	244, with harv. sys.	0.485
Scenario 2.4: MIQP	175, without harv. sys.	0.255
	287, with harv. sys.	0.318
Scenario 2.5: MILP	178, without harv. sys.	0.044
	244, with harv. sys.	0.044
Scenario 2.6: MILP	171, without harv. sys.	0.044
	284, with harv. sys.	0.044
Scenario 3.1: HDS	164, without harv. sys.	0.0038
	266, with harv. sys.	0.0038

Thus, analysis in terms of scalability should be conducted. These results (number of switches and computational time) seem to advocate for strong drawbacks for MPC solved with MIQP when compared to the other strategies. Indeed, for the optimization costs chosen for MIQP and MILP, it was quite evident that MILP would exhibit a smaller computational time. However, this analysis should be extended in order to have realistic guesses on the implementation constraints of the control strategies. Indeed, the simulation scenarios only consider 6 nodes[1], which is a very small number of nodes, compared to today WSN deployments.

[1]This number comes from the number of nodes at hand in the real benchmark.

However, from Table 4.2, it can advocate that the MPC approach solved with MIQP is effective for few sensor nodes employed in one geographically defined space, because the solution of the optimization problem seems "heavier" than the solution of other control approaches. The control strategy solved with a MILP problem is less "heavy". Also, the control strategy solved with MPC/MILP is "heavier" as the one based on HDS approach. Note that for implementation in the real test-bench, the control strategies will not be run in Matlab, and optimization routines from well known optimization packages will be used.

4.2 Simulation results in the presence of unreachable nodes

Now, we analyze the behavior of the control strategies in simulation without harvesting systems when communication disturbances are considered. In this section we only consider the strategies based on MPC solved with MILP, and based on HDS. These choices follow from the analysis before. Note that the communication disturbances are coarsely modeled. Basically, the (un)availability of data at the supervisor side is considered.

4.2.1 Simulation of the control strategies with unreachable nodes

During the simulation with unreachable nodes, we consider that the unreachable nodes consume the same energy than if they were in the mode M_1. This corresponds basically to the wast situation. This consumption choice is justified by the fact that, when a sensor node is considered by the supervisor as unreachable, this sensor node can already be in mode M_1 and it tries to connect to the supervisor during a time period.

The profiles of unreachability are described in Table 4.3 and shown in Figure 4.1. In the presence of unreachable nodes, and when extra nodes are in mode M_2, the control strategies place nodes in mode M_1 such that the mission is fulfilled.

Table 4.3: Description of the unreachability profiles

Sensor node	Start of unreachability, $[hours]$	End of unreachability $[hours]$
S_1	45	64
S_2	31	63
S_3	0	20
S_4	108	127
S_5	–	–
S_6	96	128

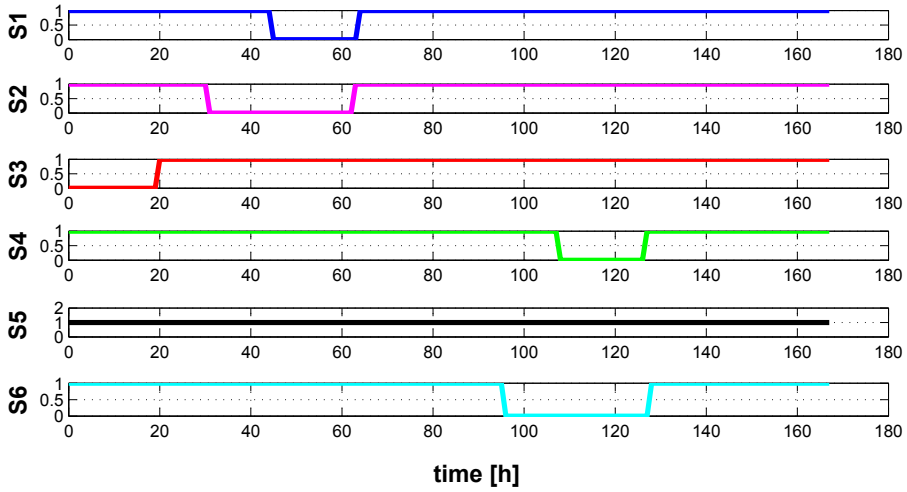


Figure 4.1: Profiles of unreachability for each sensor node. "1" stands for the node is reachable while "0" means the node is unreachable

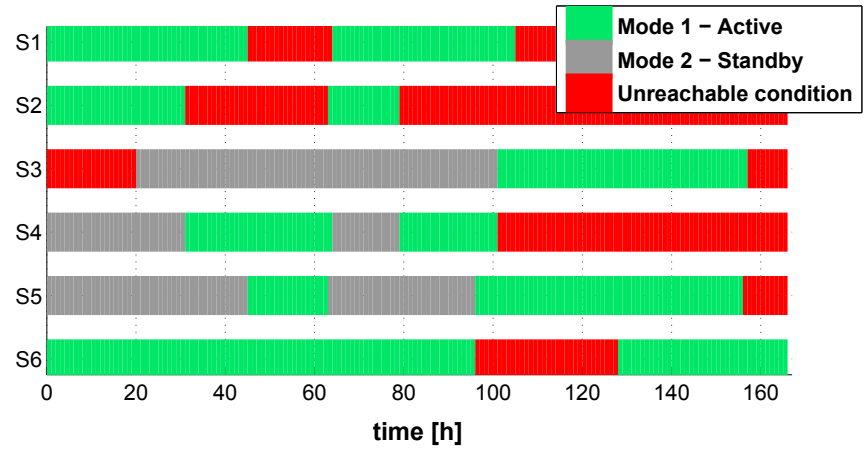
In order to evaluate the behavior of the control strategies under unreachable nodes, scenarios 2.5 and 2.6 (see Section 2.6.2) and Scenario 3.1 (see Section 3.4) are now simulated. The values of the energy consumption, battery characteristics, initial energy for each node are defined in Chapter 2 for the MPC approach when solved with MILP, and in Chapter 3 for the HDS approach.

Scenario 2.7: energy management strategy when a MILP problem is solved with $c = [b_{11}\ b_{12}\ \cdots\ b_{62}]^T$, in the presence of unreachable nodes

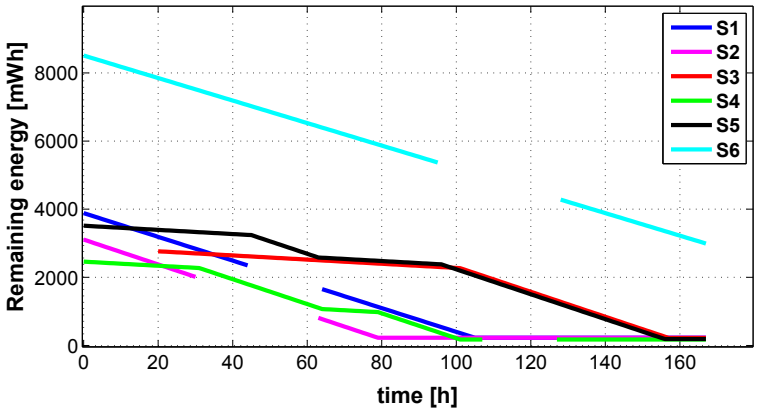
This scenario is similar to Scenario 2.5 considered in Section 2.6.2, where the MPC approach is applied and the problem is solved using MILP. Figure 4.2 shows the evolution of functioning modes and the remaining energy in the battery of each node when the unreachability profiles of Figure 4.1 are considered. As can be seen, the WSN lifespan is equal to 105 *hours* until the first time the mission is no more fulfilled. Then, during the next 23 *hours* (until 128 *hours*) the mission is no more fulfilled because only two nodes are reachable. From 128 *hours* to 156 *hours*, the mission is filled again thanks to node S_6 that becomes reachable again.

In the presence of unreachable nodes, the control strategy is able to place some nodes in mode M_1, provided that extra nodes are in mode M_2, in order to replace active nodes that have fallen in the *Unreachable* condition. In this way, the mission can still be fulfilled.

Figure 4.2(b) presents the remaining energy in each node. Note that the gaps correspond to the situation when a sensor node becomes unreachable. In this situation, we do not know how much energy is still available in the unreachable



(a) Evolution of the functioning modes of the nodes

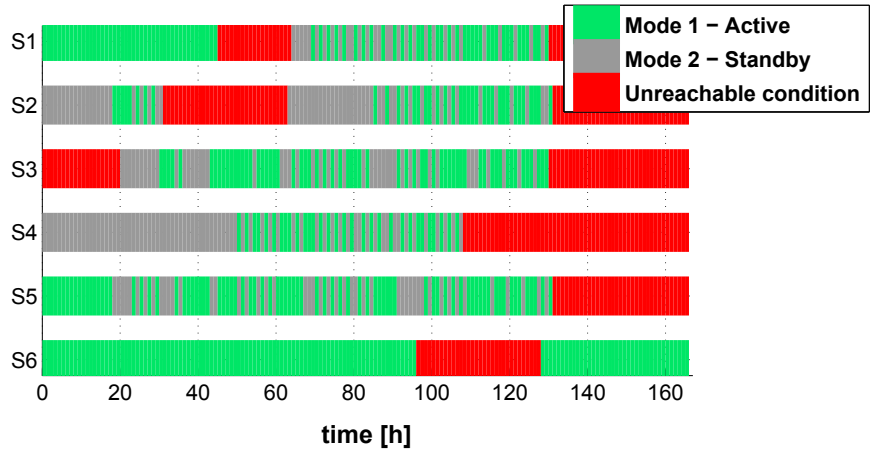


(b) Evolution of the remaining energy in each node battery

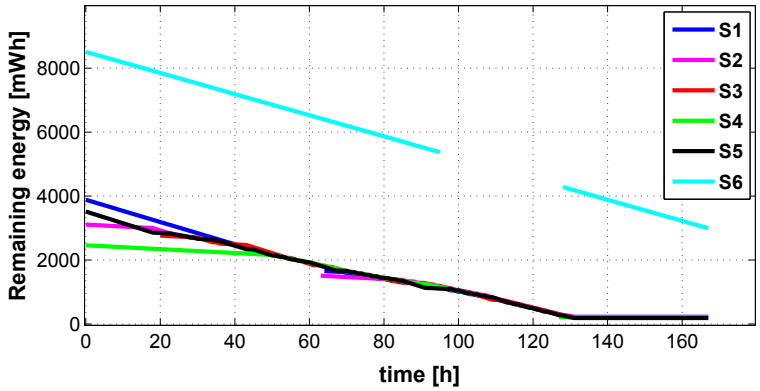
Figure 4.2: Scenario 2.7: energy management strategy based on MPC/MILP, $c = [b_{11}\ b_{12}\ \cdots\ b_{62}]^T$, with unreachable nodes

nodes. This, indeed, is fully consistent with real-life conditions. When a node comes back, i.e. it is again reachable by the controller, it sends its remaining energy level to the supervisor so that the control strategy will be able to consider this node as a potential node to be placed in the *Active* or *Standby* mode.

Scenario 2.8: energy management strategy when a MILP problem is solved with $c = [\frac{b_{11}}{x_1+E_1w_1-X_1^{min}} \ \frac{b_{12}}{x_1+E_1w_1-X_1^{min}} \ \cdots \ \frac{b_{62}}{x_6+E_6w_6-X_6^{min}}]^T$, in the presence of unreachable nodes



(a) Evolution of the functioning modes of the nodes



(b) Evolution of the remaining energy in each node battery

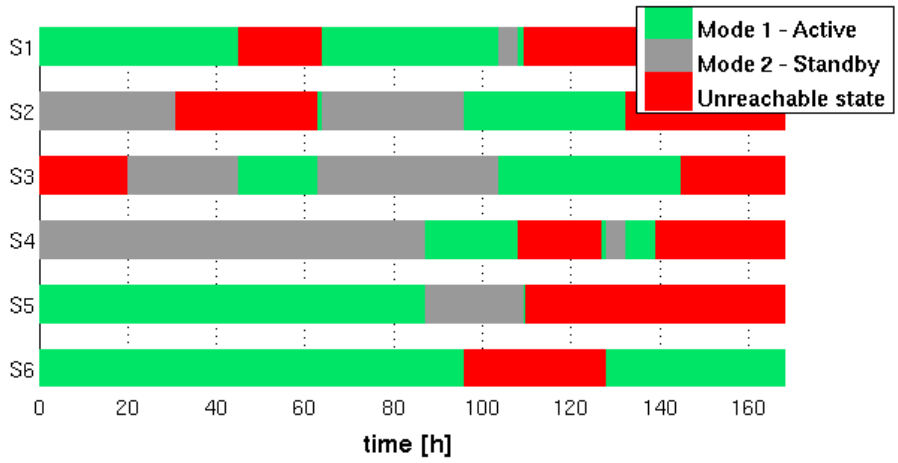
Figure 4.3: Scenario 2.8: energy management strategy based on MPC, $c = [\frac{b_{11}}{x_1+E_1w_1-X_1^{min}} \ \frac{b_{12}}{x_1+E_1w_1-X_1^{min}} \ \cdots \ \frac{b_{62}}{x_6+E_6w_6-X_6^{min}}]^T$, with unreachable nodes

This scenario is similar to Scenario 2.6 considered in Section 2.6.2 but with unteachable nodes. Figure 4.3 shows the evolution of functioning modes and the remaining energy of all nodes in the presence of unteachable nodes. It can be seen that the WSN lifespan is equal to 131 *hours*. After time $k = 131$ *hours*, the mission is never filled again compared to Scenario 2.7. This phenomenon is

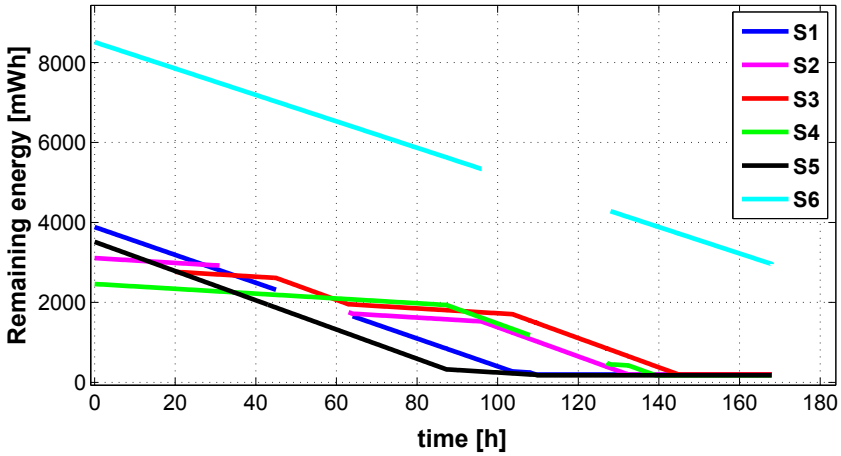
related to the choice of the problem cost for the control strategy which tries to approximately keep the same lifespan for all nodes. We can see the same gaps as in Figure 4.2(b) in the remaining energy in the battery of the nodes, see Figure 4.3(b). These gaps are related to the unreachability profiles defined in Figure 4.1.

Scenario 3.2: power management strategy based on the HDS approach in presence of unreachable nodes

Figure 4.4 shows the evolution of the functioning mode and the remaining energy



(a) Evolution of the functioning modes of the nodes



(b) Evolution of the remaining energy in each node battery

Figure 4.4: Scenario 3.2: power management strategy based on HDS, with unreachable nodes

in the battery for each node when the HDS control is applied. This scenario is similar to Scenario 3.1, except that some nodes become unreachable with the unreachability profiles given in Figure 4.1. Therefore, nodes can "disappear" and/or "appear" during the system evolution. As a consequence, nodes jump following the rules described in (3.19)-(3.21). It can be seen that the WSN lifespan is equal to 109 $hours$. Then, from time $t = 109$ $hours$ to $t = 127$ $hours$, the mission is no more fulfilled, because there is not enough reachable nodes to fulfill the mission. Indeed only 2 nodes are reachable over $t \in [109, 127]$ $hours$. Then, the mission is again satisfied during the next 12 $hours$, until time $t = 139$ $hours$. The same gaps as in Figure 4.2(b) can be seen on the remaining energy curves presented in Figure 4.4(b). These gaps correspond to the periods of time when nodes are unreachable.

4.2.2 Comparison of the results with unreachable nodes

The results for the scenarios with unreachable nodes are summarized in Table 4.4. The strategy based on MPC considered in Scenario 2.8 exhibits the larger lifespan (131 $hours$) compared to Scenario 2.7 (105 $hours$) and Scenario 3.2 (109 $hours$), without interruption of the mission. However, Scenario 2.8 exhibit a large number of switches, which can induce extra energy consumption. This also leads to extra communication overload because the supervisor has to send a new control vector much more often than in Scenario 2.7, for instance. Recall that Scenario 2.7 differs from Scenario 2.8 in the coefficient that appear in the cost function to be minimized. The strategy based on HDS, considered in Scenario 3.2 is appealing, because the WSN lifespan is similar to the one of Scenario 2.7, and the number of switches is smaller than the one of Scenario 2.8. These results also depend on the unreachability profiles. Actually, for scenarios discussed now, after a period of time when the mission cannot be fulfilled, some nodes come back to the reachable state. Then they can be placed in mode M_1 and the mission can be fulfilled again. This aspect appears in column "Extension of WSN lifespan" in Table 4.4.

Table 4.4: Comparison of scenarios with unreachable nodes

Scenarios	WSN lifespan, [$hours$]	Extension of WSN lifespan, [$hours$]	Number of switches, [1]
Scenario 2.7: MILP	105	28	7
Scenario 2.8: MILP	131	–	$\approx 10^2$
Scenario 3.2: HDS	109	12	11

4.3 Implementation on a real test-bench

The energy management strategies described in Chapters 2 and 3 are now implemented in a real test-bench in order to be compared and analyzed. The hardware and software aspects of the test-bench are first shortly described. Then, implementation issues of both control strategies are discussed. Lastly, the experimental results are provided and studied.

4.3.1 Discussion about MPC and HDS strategies assumptions

Before the implementation of the MPC and HDS control strategies for the energy management of a set of nodes, we must discuss on common and specific assumptions for each strategy. This discussion will help to properly implement both approaches on a real test-bench. It will also allow to provide a fainer comparison of the implementation results of the strategies.

Common assumptions

We have similar system models for the control strategies based on MPC (see Section 2.2) on the one hand, and on HDS (see Section 3.2) on the other hand. These system models have common assumptions:

- a given mission (service) must be fulfilled *via* a network deployed in a given geographical area. All nodes in this WSN must have the same functionality i.e. all nodes have the similar role in the system. They are interchangeable, which allows to change the active nodes by others at time k;
- the remaining energy in the battery of each node must be known at time k to implement the energy management strategies. Namely, the measure of the remaining energy y_i of the node S_i must be received otherwise, S_i is said reachable by the supervisor such that node S_i is "in the game".

These assumptions need be taken into account during real-life experiments. Namely, we will deploy a WSN with interchangeable sensor nodes in one given geographical area (i.e. a working office). A capacity estimation method proposed hereafter. Information on the remaining energy in the battery of each sensor node y_i will be obtained *via*.

Specific assumptions for each strategy

Each energy management strategy has extra assumptions:

- the energy consumed when switching from one mode to another one is naturally taken into account in $\delta_i^{k\to l}$ in the strategy based on the HDS

approach. For the MPC approach it is supposed to be integrated in b_{il}. Note that the energy consumed for a switch is very hard to measure. It depends on the sensor node itself and on the environment conditions, especially the communication disturbances;

- the strategy based on HDS consists of continuous physical processes of the energy charge/discharge of the node batteries (described by x_i and y_i), and of finite-states for the functioning modes (described by u_i). In simulation ,we have considered that the measurement of the remaining energy $y_i = x_i$ is done continuously. However, in real-life conditions (and thus in the test-bench implementation), the remaining energy in the node battery is supposed to be delivered by each node to the supervisor at each sampling time k. In the MPC strategy, this information related to the remaining energy is delivered at each sampling time k.

4.3.2 Estimation of the remaining energy in the battery node

While sensors accurately measure the gasoline level in a tank, there is no simple sensor available to measure the remaining energy in a battery. Instead, the battery State of Charge (SOC) is estimated from other measurements. Different SOC estimation methods are reported in the literature, e.g. ampere-hour counting, Open Circuit Voltage (OCV)-based estimation, model-based estimation *via* Kalman filtering [166], and others [167, 168]. Note that these approaches deal with relatively "large" battery packs for laptops and electrical vehicles. Their implementation in sensor nodes that possess limited computational capability is not appropriated. In this section, a low computational cost capacity estimation method of a node battery will be proposed.

The battery capacity represents the amount of energy that can be extracted from the battery under certain specified conditions. Battery manufacturers use the SOC to specify the battery performance. The $SOC_i \in [0, 100]\%$ (expressed in percent) describes the ratio of the remaining energy x_i to the nominal capacity $X_i^{max} \in \mathbb{R}_{>0}$ of the battery of node S_i [169]:

$$x_i = SOC_i \cdot X_i^{max}. \tag{4.1}$$

This motion is shown on Figure 4.5. Note that a new battery should have a SOC of 100% which corresponds to the nominal battery capacity, and $X_i^{min} = SOC_i^{min} \cdot X_i^{max} \neq 0$, where SOC_i^{min} is specified by the battery manufacturer or defined by the user.

The estimation of the SOC for a battery may be a more or less complex problem, depending on the battery type, the chosen estimation method, the requested estimation precision and the application in which the battery is used

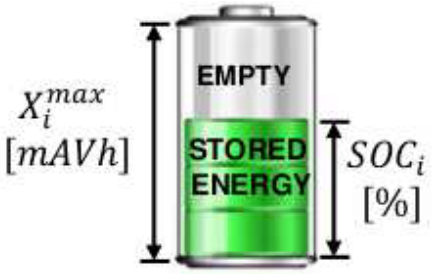


Figure 4.5: Battery SOC

[167]. According to the analysis of existing SOC estimation methods, here the ampere-hour counting method has been chosen here because:

- low-cost sensors for battery calibration are available in laboratories, for instance ammeters and voltmeters;
- the computational cost to estimate the SOC is very low;
- the estimation approach can be embedded in the computing unit of the sensor node.

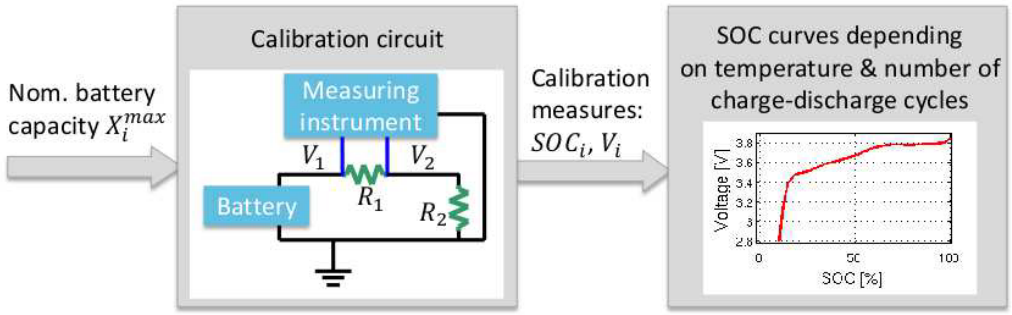
We propose an estimation method for the batteries used in our test-bench, which are Lithium-ion (Li-ion) ones. However, it can be applied to batteries with another chemistries. This estimation of the remaining energy in the battery of a sensor node is proposed to be performed in two steps as depicted in Figure 4.6:

- a battery calibration step, see Figure 4.6(a);
- an on-line estimation step, see Figure 4.6(b).

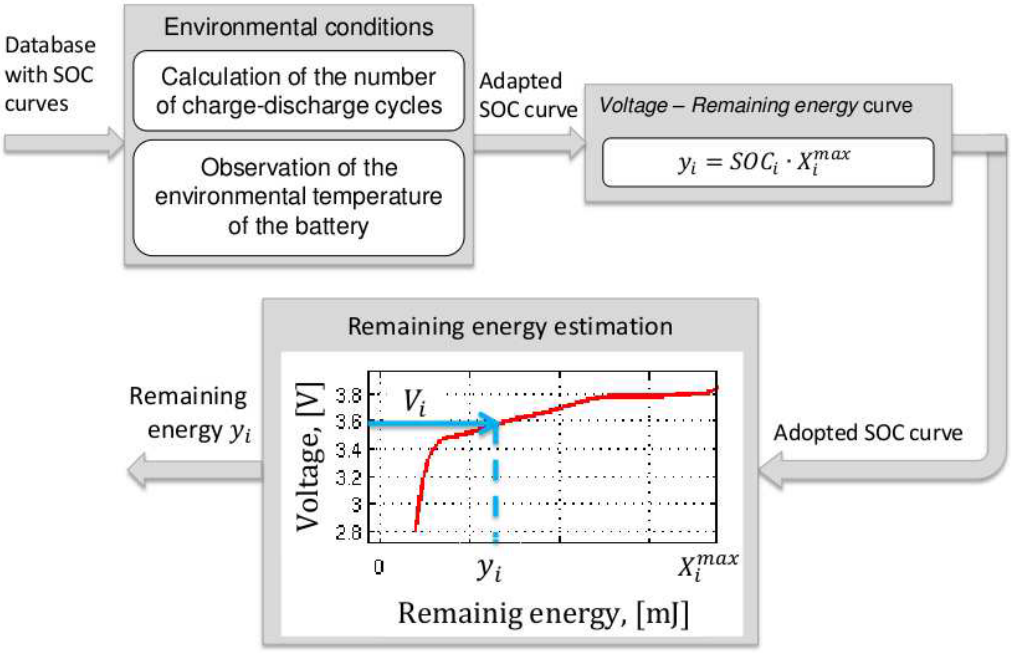
Both steps are now summarized.

Battery calibration step

The battery calibration is performed off-line during laboratory experiments on a new battery for which the SOC is considered equal to 100% (i.e. nominal capacity, taken from the data-sheet). When the battery ages, the parameters used to describe the voltage relaxation process become increasingly less accurate. The result is a decrease in the accuracy of the remaining energy estimation. To compensate the aging effect, the number of charge-discharge cycles and other environmental conditions, especially the battery environmental temperature, can be taken into account [170]. As a consequence, the estimation accuracy for aging batteries is almost as precise as for new ones. After each battery charge-discharge cycle, the battery needs to rest for at least four hours to attain its equilibrium and get accurate measurements [171].



(a) Battery calibration



(b) On-line estimation

Figure 4.6: Estimation of the remaining energy in a battery - 2-steps approach

When the nominal battery capacity is known and the current $i(t)$ extracted from the battery can be measured, an accurate calculation of SOC changes can be performed. Here, $i(t)$ is given by (see Fig. 4.6(a)):

$$i(t) = \frac{V_1 - V_2}{R_1}, \tag{4.2}$$

where R_1 is a shunt resistor. This approach can be used for Li-ion batteries because there are no significant side reactions during normal operation [168].

However, for the SOC estimation, the initial SOC value $SOC(0)$ must be known:

$$SOC_i(t) = SOC_i(0) - \int_0^t \frac{\eta \cdot i(t)}{X_i^{max}} \mathrm{d}t. \tag{4.3}$$

$i(t)$ is the instantaneous current, assumed positive for discharge, negative for charge, delivered by the battery. X_i^{max} is the nominal battery capacity. The Coulombic efficiency is $\eta = 1$ for discharge, and $\eta \leqslant 1$ for charge.

Using a rectangular approximation for the integration and a sampling period $\Delta t \equiv [k,\ k+1]$, a discrete-time approximate recurrence can be derived:

$$(SOC_i)_{k+1} = (SOC_i)_k - \frac{\eta \cdot \Delta t}{X_i^{max}} i_k. \tag{4.4}$$

The measures conducted during the battery calibration phase provide a database with the voltage vs. SOC curves (see Fig. 4.6(a)). These lattes curves depend on the temperature and the battery aging (number of charge/discharge cycles).

On-line estimation step

The on-line estimation step consists of two sub-steps. The first one selects from the database built during the calibration step, one SOC curve adapted to the environmental temperature and to the number of charge/discharge cycles. The second step estimates the remaining energy y_i at time k in the battery of sensor node S_i, using the appropriate SOC curve and the voltage measurement at the battery terminals at time k. Due to its low computational burden, this estimation phase can be implemented within the sensor node used for real-life experiments. It also can be implemented at the supervisor side, provided this after has information regarding the SOC curves for each node, the number of charge/discharge cycles, and the battery ambient temperature.

4.3.3 Test-bench description

The hardware test-bench considered here consists of one supervisor, one router and $n = 6$ sensor nodes. The supervisor is a laptop where the control strategy is implemented. It is equipped for communication with a Wi-Fi card. The Wi-Fi router is used to increase the range of the network and to form the infrastructure topology. The sensor nodes are Flyport WiFi 802.11g modules developed by OpenPicus [112] connected to a sensing element. Each node has a battery, as show in Figure 4.7. The sensing element contains a temperature & humidity sensor DHT-11 [172]. The Flyport WiFi 802.11g module is a programmable system-on-module with integrated WiFi 802.11g connectivity. The current consumption of the Flyport WiFi 802.11g module for different modes are shown in Table 4.5. For a programed sensor node, two functioning modes are, namely the *Active*

mode M_1 and the *Standby* mode M_2, that have been described for simulation purposes in Chapter 2 and in Chapter 3. Based on the technical data-sheet of the Flyport module and laboratory measurements, the numerical values of the energy consumption of the nodes have been used in simulations. They are and recalled in Table 4.6.

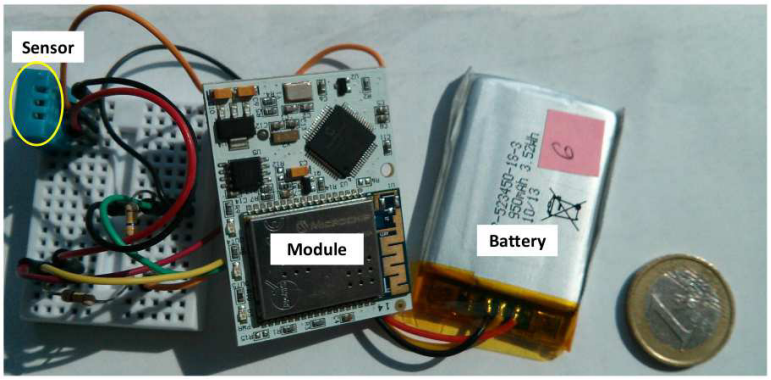


Figure 4.7: Used sensor node

Table 4.5: Current consumption of different components of the Flyport WiFi module [112]

State	**Current**	**Remarks**
Wi-Fi not connected	$39,75\ mA$	MCU *on* and Wi-Fi on but not connected
Wi-Fi connected	$162,70\ mA$	MCU *on* and Wi-Fi infrastructure mode connected to an access point
Wi-Fi burst	$282,50\ mA$	RF Burst on data TX
Hibernate mode	$28,21\ mA$	MCU *on* and Wi-Fi transceiver *off*
Sleep mode	$1,44\ mA$	MCU *off* and Wi-Fi transceiver *off*

Table 4.6: Average energy consumption b_{ij} (mWh) of node S_i in the functioning modes M_1 and M_2

Sensor node	Average current consumption in mode M_1, $[mA \cdot hour]$	Average current consumption in mode M_2, $[mA \cdot hour]$
S_1	9.42	1.58
S_2	9.86	1.63
S_3	9.89	1.65
S_4	9.86	1.63
S_5	9.88	1.65
S_6	8.93	1.55

Table 4.7: Characteristics of the node battery

Sensor node	Battery Type	Nominal Voltage V_i, $[V]$	Battery capacity P_i, $[mA \cdot h]$
S_1	Type 1	3.7	1050
S_2	Type 1	3.7	1050
S_3	Type 1	3.7	1050
S_4	Type 2	3.7	950
S_5	Type 2	3.7	950
S_6	Type 2	3.7	950

Each node has one among two types of Li-polymer rechargeable batteries [139] as show in Table 4.7. Note that harvesting systems are not available. The used Li-ion batteries have nominal capacities $X_i^{max} = 3885mWh$ for type 1, and $X_i^{max} = 3515mWh$ for type 2. These batteries embed an electronic protection circuit. This latter limits the minimum SOC value to 10% of the nominal battery capacity for type 1 and to 16% of the nominal battery capacity for type 2. As a consequence, the minimal energy for the sensor nodes using type 1 battery is equal to $X_i^{min} = 3885mWh \cdot 0.1 = 388,5mWh$ for $i = 1, 2, 3$. For type 2 battery, $X_i^{min} = 3515mWh \cdot 0.16 = 562.4mWh$ for $i = 4, 5, 6$.

After the battery calibration step described in Section 4.3.2, an accurate experimental model of the battery $Voltage - SOC$ curves is built. Figure 4.8 depicts an example of such SOC profiles for both types of new batteries, at 23°C, which corresponds to the ambient temperature in the office where the sensor nodes will be deployed. This calibration step together with the protection circuit allow to safely (i.e. without damaging the battery) and efficiently exploit the battery capabilities. The estimation of the remaining energy in the battery of all nodes is implemented, together with the control law.

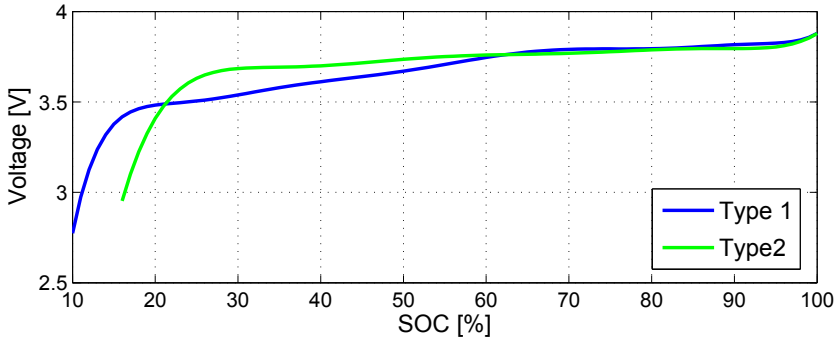


Figure 4.8: SOC profiles for both battery types

The coordination between the sensor nodes and the supervisor is realized *via* the router. It uses and used the LINC coordination environment [173]. LINC is a resource-based middleware with a particular emphasis on the sensor/actuator network field. The LINC middleware tackles issues raised by applications considering a large number of sensor nodes (i.e. several hundreds), distributed over a wide field (i.e. a building or a set of buildings) and connected via heterogeneous and unreliable communication protocols (e.g., various wireless networks) [174].

4.3.4 Definition of the mission

For this test-bench, WSN with 6 sensor nodes has been deployed in a working office as shows in Figure 4.9. In order to control the air conditioning system, temperature and humidity are sensed through the sensor nodes. During the day, when the office is in use, the control of the air conditioning system requires measurements from three sensor nodes. During the night, measurements from only one sensor node are enough to ensure the appropriate control of the air conditioning unit. Therefore, the mission is split in two phases corresponding respectively to day and night periods of time. Therefore, the constraints that define the mission have to be dynamically changed, depending on the time schedule, leading to a dynamic mission, as summarized in Table 4.8.

Table 4.8: Definition of the dynamic mission

Time period		d_1	**Objectives**
Day	8am−6pm	3	3 nodes in M_1
Night	6pm−8am	1	1 nodes in M_1

At time instant $t = 0$, all the sensor nodes of the system are active (in mode M_1). Basically, they must transmit their initial remaining energy in the battery. Then, the supervisor checks whether the node batteries have enough energy so that any node S_i can fulfill the mission (i.e. being in mode M_1). If this is true, during the day period, 3 nodes are placed in mode M_1 while the others nodes that are reachable are placed in M_2. During the night period, one node is placed in mode M_1 and the others are placed in mode M_2.

When a sensor node S_i is not seen by the supervisor, it is considered in the *Unreachable* condition. This situation occurs for instance when the remaining energy of a sensor node is lower or equal to X_i^{min}, or because of any other faulty conditions. Then, the control law assigns new modes to the remaining nodes in order to meet the dynamic mission while minimizing the energy consumption of the sensor network. When the supervisor receives again information from a node that was beforehand in the *Unreachable* condition, it places this node in mode M_1 or in mode M_2 depending on the mission fulfilling.

Figure 4.9: WSN deployed in a working office

4.3.5 Experimental results for the MPC strategy

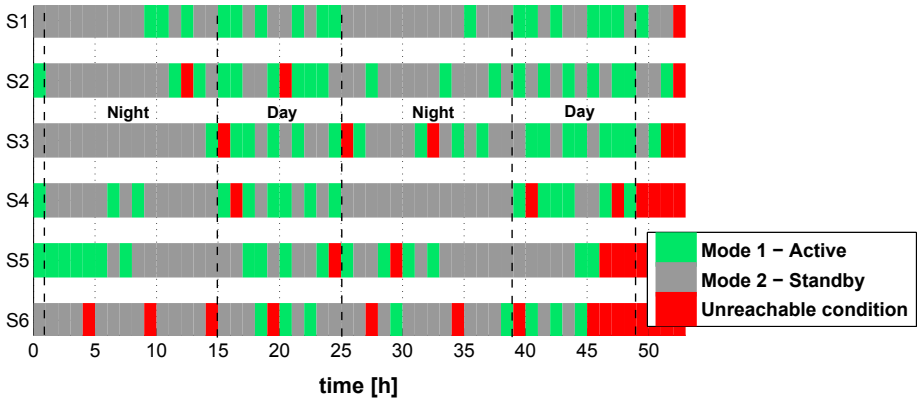
Now, the energy management strategy based on the MPC approach is implemented on the real test-bench shortly presented above. The control strategy is rewritten in Python and integrated in the LINC middleware. The MILP problem is solved with PICOS [175] using the Mosek solver [142].

Scenario 2.6 is considered (see Section 2.5.3) with the coefficients of the cost function $c = [\frac{b_{11}}{x_1+E_1w_1-X_1^{min}} \ \frac{b_{12}}{x_1+E_1w_1-X_1^{min}} \ \cdots \ \frac{b_{62}}{x_6+E_6w_6-X_6^{min}}]^T$. With this choice, the control strategy changes of the functioning mode of the nodes in order to ensure the mission. It also tries to keep the same lifespan for all sensor nodes. The experiment starts at 5 $p.m.$ and lasts 54 $hours$. The WSN lifespan is equal to 53 $hours$.

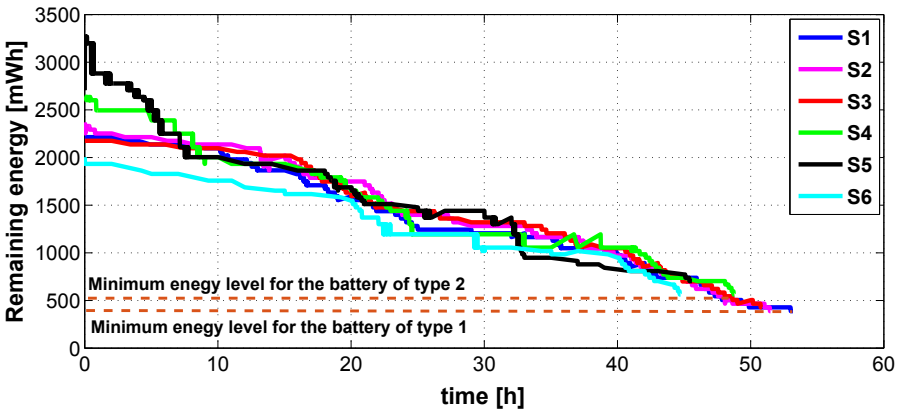
The experimental results are provided in Figure 4.10(a) that shows the functioning modes imposed by the supervisor for each sensor node. The mission during the working hours (resp. the night period) can be fulfilled until at least 3 (resp. 1) nodes do not have their batteries drained and are reachable. The estimated remaining battery capacities are given in Figure 4.10(b).

During the WSN lifespan, the nodes S_2 - S_6 fall in the *Unreachable* condition at the different time instances that may be related to different radio channel perturbations. When a node fall in the *Unreachable* condition, it is no longer taken into account by the supervisor when solving the control problem. When the supervisor receives again information related to the remaining energy of the sensor node, that was unreachable beforehand, the supervisor adds that node again in the set of nodes to be controlled.

The perturbation of the radio channel can also influence on inaccurate esti-



(a) Evolution of the functioning modes of the nodes



(b) Estimated remaining energy of each sensor node

Figure 4.10: Scenario 2.4 evaluation in real-life conditions

mation of the remaining energy in the battery. This may explain the oscillations observed in the evolution of the estimated remaining energy in the nodes. Recall that no harvester is used.

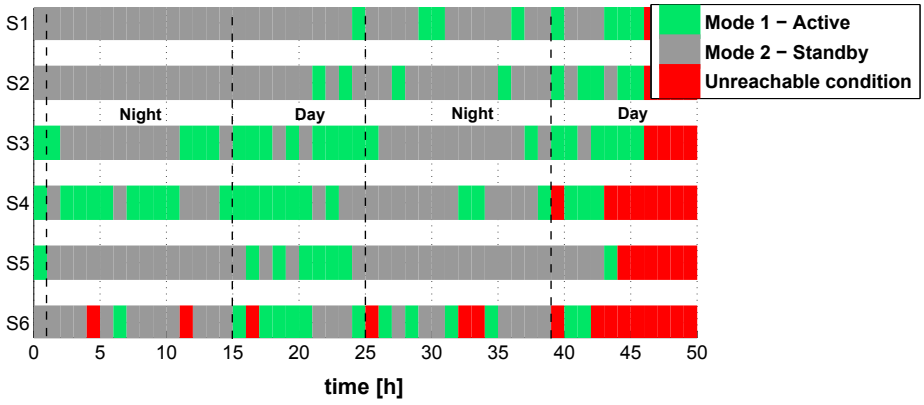
The remaining energy in the node batteries does not go down lower than the minimum energy level provided by the manufacturer. For the nodes $S_1 - S_3$ with the battery of type 1, the minimum energy level is equal to 388.5 mWh. For nodes $S_4 - S_6$ with the battery of type 2, the minimum energy level is equal to 562.4 mWh. At the end of the experiment, the sensor nodes have the following remaining energy in their batteries: S_1 – 388.5 mWh; S_2 – 388.5 mWh; S_3 – 427.35 mWh; S_4 – 562.4 mWh; S_5 – 703 mWh; S_6 – 562.4 mWh.

We can see in Figure 4.10(b) that the slope of the curves of the estimated

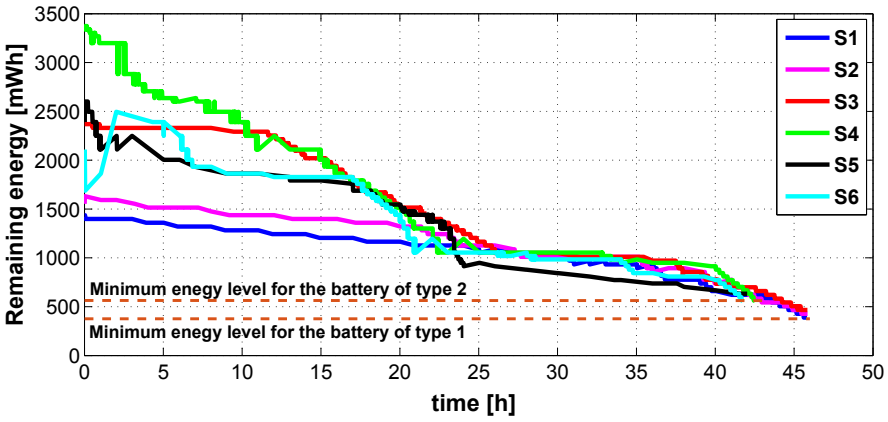
remaining energy is proportional to the energy consumption in the different functioning modes of the nodes. The number of switches is also large than in the simulated scenario (see for instance Section 4.2). Here we have $\approx 10^2$ switches.

4.3.6 Experimental results for the HDS strategy

The energy management strategy based on the HDS approach is now implemented on the same test-bench. For that purpose, the control strategy is also written in Python and integrated in the LINC middleware.



(a) Evolution of the functioning modes of the nodes



(b) Estimated remaining energy of each sensor node

Figure 4.11: Scenario 3.1 evaluation in real-life conditions

Figure 4.11 shows the evaluation of Scenario 3.1 from Section 3.4 in the real-life conditions. The implementation starts at 5 *p.m.* and lasts 50 *hours*. The

evolution of the functioning modes and the estimated remaining energy of all nodes are presented in Figure 4.11(a) and Figure 4.11(b) respectively. It can be observed that the WSN lifespan is equal to 46 *hours* where the dynamic mission is fulfilled periodically for day and night periods of time. During the WSN lifespan, sensor nodes S_4 and S_6 fall in the *Unreachable* condition, which may be caused by perturbations of radio channel. At the end of experiments, the nodes still possess the following remaining energy in their batteries: S_1 – 388.5 mWh; S_2 – 427.35 mWh; S_3 – 466.20 mWh; S_4 – 562.4 mWh; S_5 – 632.7 mWh; S_6 – 562.4 mWh.

We can see in Figure 4.11(b) that the slope of the curves of the estimated remaining energy is also proportional to the energy consumption in the different functioning modes of the nodes. The number of switches is more important in the real-life conditions compared with the simulation from Section 3.4. This can reflect large radio channel perturbations and the dynamic mission fulfilling that impose extra switches. Moreover, the switching cost evaluation may be inaccurate, leading to to a change of the estimated node remaining lifespan.

4.3.7 Comparison of implementation results of both strategies

A summary of both control strategies implemented in a real test-bench is shown in Table 4.9. The WSN lifespan with the strategy based on MPC is equal to 53 *hours* with the initial energy of whole WSN equal to $14.521 * 10^3$ mWh. The WSN lifespan with the strategy based on HDS is equal to 46 *hours* with the initial energy of whole WSN equal to $13.242 * 10^3$ mWh. Considering that the initial energy of the node batteries is 8.81% larger in the case of MPC, we got a lifespan 13.21% larger by using this MPC strategy compared to the HDS one. We can see that the MPC control provides a better result then HDS one. However, the comparison of both strategies in implementation is quite, since the initial conditions are different. In addition, the perturbation due to the environmental conditions and the profiles of the unreadability of nodes may modify the WSN lifespan and, as consequence, the comparison of the implementation results can differ.

Table 4.9: Comparison of the experiments scenarios in terms of WSN lifespan

Experiment	WSN lifespan, [*hours*]	Initial energy of whole WSN, [mWh]	Final energy of whole WSN, [mWh]
Scenario 2.6: MILP	53	$14.521 * 10^3$	$3.032 * 10^3$
Scenario 3.1: HDS	46	$13.242 * 10^3$	$3.039 * 10^3$

Figure 4.12 shows the evolution for the estimated remaining energy of all

nodes obtained with both control strategies during the experiments. We can see that the behavior of the both curves is similar. The slope of the curves is larger (i.e. the energy consumption is higher) when it is necessary fulfill the day mission (we need 3 active nodes), and the slope is smaller when the night mission (1 active node) is fulfilled.

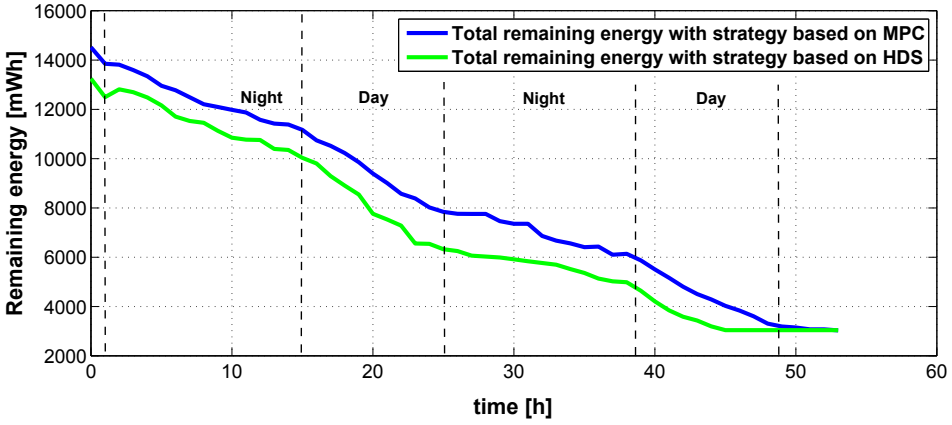


Figure 4.12: Evaluation of the total remaining energy for the whole WSN

4.4 Conclusion

In this chapter we have compared two control strategies based on a MPC and a HDS approaches. From the comparison of simulated scenarios, we have chosen three scenarios 2.5, 2.6 and 3.1 that seen promising in terms of WSN lifespan extend and implementation cost. The behavior of the control strategies for these scenarios have been evaluated in simulation even when unreachable nodes are considered. This leads to more realistic simulation conditions when compared to real-life conditions. Consequently, based on the simulation results with unreachable nodes the scenarios 2.6 and 3.1 are considered to be implemented.

We have detailed the common and specific assumptions for each strategy. To validate the assumption of the known remaining energy in the battery of each node at time k, a method for estimating the remaining energy has been proposed. It has a low computational cost that allows its implementation at the supervisor side or at the node side.

The real test-bench is considered for the experiments. The hardware taken in the test-bench is the industrial material, the some of them (sensors, router, batteries) are already used in building automation. The energy consumption for all of these devices has been measured to define the energy consumption in each functioning mode of a whole node. The mission can change dynamically, which allows to take into account the environment changes with time.

The energy management strategies have be successfully implemented on the real test-bench. The experimental results show that the WSN lifespan is larger with the MPC strategy compared with the HDS strategy by 13.21% with the initial energy larger by 8.81%. This comparison is quite difficult because profiles of the unreachability of nodes and the environmental conditions, that increase the energy consumption, are not same in the both experiments. However, we can say that the both control strategies can be used in the real WSN to manage the energy consumption of the set of nodes.

Chapter 5

Conclusion and Perspectives

Wireless Sensor Networks (WSNs) have undergone in the last few years a tremendous growth, both in industry and in academia. This is mainly due to the different potentials of this technology such as:

- the wire cost is reduced with the use of wireless sensor nodes. The wire cost is actually becoming more and more prominent in the deployment of sensor networks. Moreover, wires imply maintenance costs;
- application domains that were inaccessible to wired sensor nodes are now at reach thanks to miniaturized wireless measurement devices.

However, WSNs also need to face significant design challenges because of their limited computing and storage capabilities and also their dependence on limited energy as the sensor nodes are usually supplied by a battery. The energy is a critical resource and it often constitutes a major obstacle to the deployment of sensor networks that will be used everywhere in the world of tomorrow.

This PhD thesis focused on energy/power management for a set of sensor nodes, with a particular emphasis on energy conservation at the application level.

5.1 Synthesis

In Chapters 2 and 3, we proposed two different strategies based of Control Theory to manage the energy in a WSN. The strategies are located at the application level in order to provide a given service, named mission in the present work. The objective is to increase the lifespan of the whole WSN in which the sensor nodes and network protocols are already optimized for energy conservation while fulfilling the so-called mission.

The proposed control strategies are based on two different approaches from the Automatic Control Theory community, namely, Model Predictive Control (MPC) and Hybrid Dynamical System (HDS) approach.

The first strategy, based on a MPC approach, is described in Chapter 2. For this MPC strategy, the model of the energy consumption for the whole set of sensors is proposed. The system model takes into account various constraints such as a limitation of the battery capacity of a node and constraints related to the mission. Recall that the mission quantifies the service as a given (minimum) number of measurements that must be provided by the WSN to the application level over a defined period of time. All nodes in this WSN must have the same functionality, i.e. they are interchangeable, which allows to change the active nodes by others at any time. We considered that some nodes are powered by a battery together with harvesting capability.

The strategy places a reduced number of nodes in the *Active* mode, while other nodes are either in the *Standby* mode or in the *Unreachable* condition. The choice of the active nodes is guided by the minimization of the energy consumption, or equivalently, by the extension of the network lifespan. The control problem is tacked with the MPC approach. It is solved using MIQP or MILP, depending on the cost function to be minimized. For a particular scenario, the simulation results show that the WSN lifespan can be extended by a factor of 1.4 without harvesting systems and by a factor of 1.5 with harvesting systems, when compared to the "basic" situation where sensor nodes are activated only to fulfill the mission.

The simulation results on a realistic bench-mark of the MPC strategy show the effectiveness of this approach in terms of increasing the WSN lifespan. However, this strategy presents some drawbacks. The switching cost is roughly taken into account in the matrix that models the energy consumption of each node in a given functioning mode. This model cannot take into account the link quality between a node and the supervisor, i.e. the change of the radio link environment is neglected. Beside, the MPC strategy, especially when MIQP is used, is complex in terms of computation, i.e. it has a significant energy cost and time cost, which can make it difficult to be embedded in a computing node.

The results of the research activities presented in Chapter 2 have been published in IECON'14 [117] and ECC'15 [118].

Given these drawbacks, the second strategy based on a HDS approach is proposed for the energy/power management of the WSN. This is presented in Chapter 3. Actually, the use of a hybrid approach is natural because the WSN system is *per se* hybrid. Moreover, this approach also naturally takes into account the switching cost between two functioning modes of a node. The scheduling law is designed based on the cost function definition and on the jump rules.

This scheduling law fixes the mode of each node at time t so as to fulfill the so-called mission. The scheduling law is first designed in the simple case with 2 nodes, 2 modes (and no *Unreachable* condition) and without harvesting systems. In this situation, the solution is proved to be unique with maximum lifespan for the WSN. Then, the scheduling law is extended to the case with more than 2 nodes and more than 2 modes, taking into account the unreachability of some nodes and the harvested energy.

Using the HDS strategy, it can be shown in simulation that the WSN lifespan can be extended by a factor of 1.28 without harvesting systems and by a factor of 1.39 with harvesting systems compared to the "basic" situation. Moreover, the benefit of this HDS strategy is its low computational complexity, because it only requires the evaluation of a few algebraic equations. As a consequence, running the HDS control strategy itself will consume a small amount of energy, especially when compared to the MPC approach that requires solving an optimization problem.

The research activities presented in Chapter 3 have been accepted for publication in CDC'15 [152].

The comparison of both control strategies for different scenarios on a realistic bench-mark is first conducted in simulation, in the Matlab environment. The results are presented in Chapter 4. We can see that the MPC strategy seems to extend a bit more the network lifespan when compared to the HDS approach. However, the HDS strategy is simpler and it takes into account the radio environment changes, which is a clear benefit of this strategy.

This comparison in simulation allowed to choose the more promising scenarios for both strategies for a hardware implementation.

The second part of Chapter 4 is devoted to the implementation of the selected scenarios on a real test-case. In order to realize the experiments, the common and specific assumptions for both strategies are recalled and discussed. One of this assumption is related to the knowledge of the remaining energy in the battery of each node. An estimation method of the remaining energy has been proposed and implemented in order to fulfill this assumption. The estimation method presents a low computational cost because it is based on a simple Ampere-counting approach. Thus it can be implemented within the computational part of a node.

We choose to implement a dynamic mission. Beside the fact that we want to prove that our strategies can cope with adaptive context, this dynamic mission fits with the changes in the test-case environment needs. Moreover, this allows even more energy conservation in the WSN.

Both energy management strategies have been successfully implemented, and the WSN lifespan can be extended (the mission being fulfilled) when compared to a basic configuration. The experiments are realized in a real working office.

The application consists in monitoring the environment parameters, namely, the temperature and humidity level. During the experiments, the radio link quality is not monitored (nor controlled!). It is known that this radio link quality strongly influences the WSN lifespan results and, consequently, the comparison of both control strategies must be conducted carefully, due to the different radio perturbation profiles that are in practice not known. However, it is important to note that the implementation results are very promising for both strategies.

The research activities related to the estimation method of the remaining energy presented in Chapter 4 has been published in NEWCAS'15 [165]. Moreover, activities related to the implementation of the MPC approach was partially presented at ECC'15 [118].

5.2 Perspectives

The work presented in this document opens many research perspectives. We structure our future works in two categories, namely, theoretical and practical ones.

The theoretical future works may consist in:

- the way the switching cost in the MPC strategy is taken into account must be revisited. Actually, the switch of a node from one node to another one has a cost that must be properly taken into account. Moreover, this switch (energy) cost will decrease the WSN lifespan. As a consequence, it should be interesting to minimize as well the number of switches between two different functioning modes for a node. This could be taken into account in the MPC strategy *via* the introduction of $\mathbf{u}_k^T \Delta \mathbf{u}_{k-1}$ in the cost function, where $\Delta = \text{diag}(\Delta_1, ..., \Delta_n)$ and

$$\Delta_i = \begin{bmatrix} 0 & \delta_i^{2\to 1} & \cdots & \delta_i^{h\to 1} & \cdots & \delta_i^{m\to 1} \\ \vdots & & \ddots & & & \vdots \\ \delta_i^{1\to m} & \delta_i^{2\to m} & \cdots & \delta_i^{h\to m} & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m\times m}. \tag{5.1}$$

  This problem becomes even more complex: the control $\mathbf{u}_k$ should be calculated to minimize the cost function (see (2.14)) and at the same time $\mathbf{u}_k^T \Delta \mathbf{u}_{k-1}$ must be minimal;

- both control strategies should be tested on other missions. An interesting situation is related to the case where several missions have to be fulfilled at the same time with the same set of sensor nodes, leading to possibly contradictory decisions regarding the mode of each sensor node. Solving these conflicts will lead to a compromise between the missions fulfillment and the energy consumption. This problem could be tackled *via* the solution of a multi-objective optimization problem;

- future works should also target a more accurate analysis of the computational complexity of the proposed control strategies. Indeed, this complexity might prevent the control strategies scalability. Moreover, it might be interesting to implement in a sensor node (with limited computational capabilities) the control strategy, this node becoming the supervisor. Then, depending on the remaining energy in this "supervisor - sensor node", one could choose another sensor node as supervisor;

- after the complexity analysis of the proposed control strategies, the distribution of the control should be considered. Currently, the sensor nodes are spatially distributed, but the control is centralized (in a supervisor). For the distribution of the control strategy, the communication architecture must be re-thought. The energy consumption model will be modified, depending on the communication architecture (unicast, multicast, broadcast, geocast, etc.). The need for such a distribution of the control will depend on the system architecture. Indeed, if the system is highly hierarchical, a centralized approach might be well adapted.

The future practical works should provide a deeper analysis of the proposed strategies when implemented on real test-cases. For instance, the test-case studied during this PhD thesis should be extended to a larger number of nodes.

Sensor nodes with extra functionalities are regularly delivered by the Industry (as well as the academic world). At the moment, the sensor nodes at hand can neither send their power/energy consumption in a given functioning mode, nor the energy cost of switches between two modes, nor the remaining energy in the battery. This information is of great interest for our strategies because the choice of the functioning mode for each sensor node depends on its actual power consumption in the various functioning modes and on the remaining energy in its battery. At the moment, these values are estimated from data-sheets and laboratory measurements. However, it is well known that, depending on the radio link quality, the consumption of the node will change, but at the moment, we cannot cope with this! The best we can do is a "best effort approach". Actually we consider the worst situation for the power consumption in each mode. Also, it should be of great interest to be able to wake-up a node at any time. For instance, consider that an *Active* node becomes unreachable. If we want to fulfill the mission all the time, we have to activate a sensor node in order to replace this unreachable node. Unfortunately, this functionality is seldom implemented in industrial sensor nodes. Thus, the present PhD thesis suggests new functionalities to be embedded in sensor nodes.

Last, but not least, from a more personal perspective, it should be interesting to implement one of the control strategies in a large-scale test-case located in the Rostan school (my school during 10 years) in Ukraine (see Figure 5.1). The

main goal of this test-case would be to monitor the energy consumption, air quality, luminosity and occupancy of the whole school. Depending on the time period (day/night), the number of measurements required for the monitoring will change, and some nodes have to be placed in the *Standby* mode. Then, from this monitoring, the ultimate goal will to reduce the energy consumption of the school. This objective is clearly larger than the one studied in this PhD thesis as we will have to deal with sensor nodes, but also actuators with several functioning modes.

Figure 5.1: Smart school at Rostan, Ukraine

Bibliography

[1] B. Krishnamachari. *Networking wireless sensors*. Cambridge University Press, 2005. [cited at p. xi]

[2] R. Kacimi. *Techniques de conservation d'énergie pour les réseaux de capteurs sans fil*. PhD thesis, Institut National Polytechnique de Toulouse, 2009. [cited at p. xii]

[3] C.-Y. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003. [cited at p. 2]

[4] R. T. Lacoss. Distributed mixed sensor aircraft tracking. In *American Control Conference*, pages 1827–1830, 1987. [cited at p. 2]

[5] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "smart dust". In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278, 1999. [cited at p. 2]

[6] J.M. Rabaey, M.J. Ammer, Jr. da Silva, J.L., D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, 2000. [cited at p. 2]

[7] L. Lessig. *The future of ideas: The fate of the commons in a connected world*. Vintage, 2002. [cited at p. 2]

[8] `www.libelium.com`, 2015. [cited at p. 2]

[9] M. Billinghurst and T. Starner. Wearable devices: new ways to manage information. *Computer*, 32(1):57–64, 1999. [cited at p. 2]

[10] J. Farringdon, A. J. Moore, N. Tilbury, J. Church, and P. D. Biemond. Wearable sensor badge and sensor jacket for context awareness. In *The Third International Symposium on Wearable Computers*, pages 107–113, 1999. [cited at p. 2]

[11] M. Migliardi, M. Gaudina, and A. Brogni. Enhancing personal efficiency with pervasive services and wearable devices. In *International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 152–158, 2011. [cited at p. 3]

[12] P. Castillejo, J.-F. Martinez, J. Rodriguez-Molina, and A. Cuerva. Integration of wearable devices in a wireless sensor network for an e-health application. *IEEE Wireless Communications*, 20(4):38–49, 2013. [cited at p. 3]

[13] Definition of connected car - what is the connected car? defined. AUTO Connected Car, 2014. [cited at p. 3]

[14] A. Sharma, R. Chaki, and U. Bhattacharya. Applications of wireless sensor network in intelligent traffic system: A review. In *3rd International Conference on Electronics Computer Technology*, volume 5, pages 53–57, 2011. [cited at p. 3]

[15] J. Sun, Z. Wu, and G. Pan. Context-aware smart car: from model to prototype. *Journal of Zhejiang University SCIENCE*, 10(7):1049–1059, 2009. [cited at p. 3]

[16] De la maison communicante au bâtiment intelligent, 2015. [cited at p. 4]

[17] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007. [cited at p. 4, 10]

[18] M. Beaudin and H. Zareipour. Home energy management systems: A review of modelling and complexity. *Renewable and Sustainable Energy Reviews*, 45(0):318–335, 2015. [cited at p. 4]

[19] N.K. Suryadevara, S.C. Mukhopadhyay, S.D.T. Kelly, and S.P.S. Gill. Wsn-based smart sensors and actuator for power management in intelligent buildings. *IEEE/ASME Transactions on Mechatronics*, 20(2):564–571, 2015. [cited at p. 4]

[20] H. A. A-zkan. A new real time home power management system. *Energy and Buildings*, 97(0):56–64, 2015. [cited at p. 4]

[21] C. Gomez and J. Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6):92–101, 2010. [cited at p. 4]

[22] Home automation superstore, `www.smarthome.com`, 2015. [cited at p. 4]

[23] R. G. Hollands. Will the real smart city please stand up? intelligent, progressive or entrepreneurial? *City*, 12(3):303–320, 2008. [cited at p. 4]

[24] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci. Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors. In *Fourth International Conference on Sensor Technologies and Applications*, pages 281–286, 2010. [cited at p. 4]

[25] K. Su, J. Li, and H. Fu. Smart city and the applications. In *International Conference on Electronics, Communications and Control*, pages 1028–1031, 2011. [cited at p. 4]

[26] D. Washburn, U. Sindhu, S. Balaouras, R. A. Dines, N. Hayes, and L. E. Nelson. Helping cios understand "smart city" initiatives. *Growth*, 17, 2009. [cited at p. 4]

[27] T. Nam and T. A. Pardo. Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, pages 282–291, 2011. [cited at p. 4]

[28] A. Caragliu, C. Del Bo, and P. Nijkamp. Smart cities in europe. *Journal of urban technology*, 18(2):65–82, 2011. [cited at p. 4]

[29] A. Bielsa. Wireless applications the smart city project in santander, 2013. http://www.sensorsmag.com/wireless-applications/smart-city-project-santander-11152. [cited at p. 4]

[30] Industrial networking and iot, `www.cisco.com`, 2015. [cited at p. 5]

[31] R. Conant. Wireless sensor networks: Driving the new industrial revolution. *Industrial Embedded Systems*, pages 8–11, 2006. [cited at p. 5]

[32] A. Tiwari, F. L. Lewis, and S. S. Ge. Wireless sensor network for machine condition based maintenance. In *Control, Automation, Robotics and Vision Conference*, volume 1, pages 461–467, 2004. [cited at p. 5]

[33] M. Becker, B.-L. Wenning, C. Görg, R. Jedermann, and A. Timm-Giel. Logistic applications with wireless sensor networks. In *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*, 2010. [cited at p. 5]

[34] S. Spieker and C. Rohrig. Localization of pallets in warehouses using wireless sensor networks. In *16th Mediterranean Conference on Control and Automation*, pages 1833–1838, 2008. [cited at p. 5]

[35] H. Ren, M. Q.-H. Meng, and X. Chen. Physiological information acquisition through wireless biomedical sensor networks. In *2005 IEEE International Conference on Information Acquisition*, 2005. [cited at p. 5]

[36] P. S. Pandian, K. Mohanavelu, K. P. Safeer, T. M. Kotresh, D. T. Shakunthala, P. Gopal, and V. C. Padaki. Smart vest: Wearable multi-parameter remote physiological monitoring system. *Medical engineering & physics*, 30(4):466–477, 2008. [cited at p. 5]

[37] U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. Amon: a wearable multiparameter medical monitoring and alert system. *IEEE Transactions on Information Technology in Biomedicine*, 8(4):415–427, 2004. [cited at p. 5]

[38] R. Paradiso, G. Loriga, and N. Taccini. A wearable health care system based on knitted integrated sensors. *IEEE Transactions on Information Technology in Biomedicine*, 9(3):337–344, 2005. [cited at p. 5]

[39] S. Khan, A.-S. K. Pathan, and N. A. Alrajeh. *Wireless Sensor Networks: Current Status and Future Trends*. CRC Press, 2012. [cited at p. 6]

[40] J. Porter, P. Arzberger, H.-W. Braun, P. Bryant, S. Gage, T. Hansen, P. Hanson, C.-C. Lin, F.-P. Lin, T. Kratz, et al. Wireless sensor networks for ecology. *BioScience*, 55(7):561–572, 2005. [cited at p. 6]

[41] I. F Akyildiz and E. P. Stuntebeck. Wireless underground sensor networks: Research challenges. *Ad Hoc Networks*, 4(6):669–686, 2006. [cited at p. 6]

[42] M. Hefeeda and M. Bagheri. Wireless sensor networks for early detection of forest fires. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, 2007. [cited at p. 6]

[43] K. K. Khedo, R. Perseedoss, A. Mungur, et al. A wireless sensor network air pollution monitoring system. *arXiv preprint arXiv:1005.1737*, 2010. [cited at p. 6]

[44] M. P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic. A survey of military applications of wireless sensor networks. In *Mediterranean Conference on Embedded Computing*, pages 196–199, 2012. [cited at p. 6]

[45] R. Rahul. Swarm robotics: a developing field, ieeesbcet.org/swarm-robotics-a-developing-field, 2013. [cited at p. 6]

[46] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996. [cited at p. 6]

[47] A. Durresi, M. Durresi, and L. Barolli. Security of mobile and heterogeneous wireless networks in battlefields. In *International Conference on Parallel Processing - Workshops*, pages 167–172, 2008. [cited at p. 6]

[48] A.-B. García-Hernando, J.-F. Martínez-Ortega, J.-M. López-Navarro, A. Prayati, and L. Redondo-López. *Problem Solving for Wireless Sensor Networks*. Springer, 2008. [cited at p. 6]

[49] J. A. Stankovic. Research challenges for wireless sensor networks. *ACM SIGBED Review*, 1(2):9–12, 2004. [cited at p. 6]

[50] J. L. Hill. *System architecture for wireless sensor networks*. Diss. University of California, Berkeley, 2003. [cited at p. 6]

[51] J.F. Morais and G.L. Siqueira. Wireless technologies environmental impacts. In *SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference*, pages 523–527, 2009. [cited at p. 7]

[52] A. Didioui. *Energy-aware transceiver for energy harvesting wireless sensor networks*. PhD thesis, Université Rennes 1, 2014. [cited at p. 7, 16]

[53] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002. [cited at p. 8, 10, 14]

[54] V. Mhatre and C. Rosenberg. Homogeneous vs heterogeneous clustered sensor networks: a comparative study. In *IEEE International Conference on Communications*, volume 6, pages 3646–3651, 2004. [cited at p. 8]

[55] F. L. Lewis et al. Wireless sensor networks. *Smart Environments: Technologies, Protocols, and Applications*, pages 11–46, 2004. [cited at p. 8]

[56] M. Umadevi and M. Devapriya. Supervising the employees of clusters in wsn–a novel perusal. *International Journal Of Scientific Research And Education*, 2(9), 2014. [cited at p. 8]

[57] A. N. Das, D. O. Popa, P. M. Ballal, and F. L. Lewis. Data-logging and supervisory control in wireless sensor networks. *International Journal of Sensor Networks*, 6(1):13–27, 2009. [cited at p. 8]

[58] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004. [cited at p. 8]

[59] B. Romdhani. *Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage*, diss. insa de lyon edition, 2012. [cited at p. 8]

[60] Ijinus, `www.ijinus.com/fr/applications`, 2015. [cited at p. 8]

[61] G. De Sousa, J. P. Chanet, A. Jacquot, D. Boffety, G. André, and K. M. Hou. Data collection and management solution for wireless sensor networks. In *AgEng 2010: International Conference on Agricultural Engineering*, 2010. [cited at p. 8]

[62] M2ocity, `www.m2ocity.com`, 2015. [cited at p. 8]

[63] N. C. De Castro, C. C. De Wit, and K. H. Johansson. On energy-aware communication and control co-design in wireless networked control systems. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 49–54, 2010. [cited at p. 10]

[64] J. D. Day and H. Zimmermann. The osi reference model. *Proceedings of the IEEE*, 71(12):1334–1340, 1983. [cited at p. 10]

[65] A. Boukerche. *Algorithms and protocols for wireless, mobile Ad Hoc networks*, volume 77. John Wiley & Sons, 2008. [cited at p. 10]

[66] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008. [cited at p. 10]

[67] O. Berder and O Sentieys. Powwow: Energy-efficient hw/sw techniques for wireless sensor networks. *Proceedings of the Workshop on Ultra-Low Power Sensor Networks*, pages 229– 233, 2010. [cited at p. 11]

[68] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004. [cited at p. 12]

[69] I. Ahmedy, M. A. Ngadi, S. N. Omar, and J. Chaudhry. A review on wireless sensor networks routing protocol: Challenge in energy perspective. *Scientific Research and Essays*, 6(26):5628–5649, 2011. [cited at p. 12]

[70] A. Alkhatib and G. S. Baicher. Wireless sensor network architecture. In *International conference on computer networks and communication systems IPCSIT*, volume 35, pages 11–15, 2012. [cited at p. 12]

[71] N. H. Mak and W.K.G. Seah. How long is the lifetime of a wireless sensor network? In *International Conference on Advanced Information Networking and Applications*, pages 763–770, 2009. [cited at p. 12]

[72] D. Dessales. *Conception d'un réseau de capteurs sans fil, faible consommation, dédié au diagnostic in-situ des performances des bâtiments en exploitation.* PhD thesis, Poitiers, 2011. [cited at p. 13]

[73] T. B. Reddy, I. Karthigeyan, B.S. Manoj, and C. S. R. Murthy. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad Hoc Networks*, 4(1):83–124, 2006. [cited at p. 13]

[74] J. Lee and N. Jindal. Energy-efficient scheduling of delay constrained traffic over fading channels. *IEEE Transactions on Wireless Communications,*, 8(4):1866–1875, 2009. [cited at p. 13]

[75] C. Schurgers, O. Aberthorne, and M. Srivastava. Modulation scaling for energy aware communication systems. In *Proceedings of the International symposium on Low power electronics and design*, pages 96–99. ACM, 2001. [cited at p. 13]

[76] M. Nagahara, D. E. Quevedo, and J. Ostergaard. Sparse packetized predictive control for networked control over erasure channels. *IEEE Transactions on Automatic Control*, 59(7):1899–1905, 2014. [cited at p. 13]

[77] D.E. Quevedo and A. Ahlen. A predictive power control scheme for energy efficient state estimation via wireless sensor networks. In *47th IEEE Conference on Decision and Control*, pages 1103–1108, 2008. [cited at p. 13]

[78] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002. [cited at p. 14]

[79] H. A. Nguyen, A. Forster, D. Puccinelli, and S. Giordano. Sensor node lifetime: An experimental study. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 202–207, 2011. [cited at p. 14, 17]

[80] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanovic. Energy-aware design of compressed sensing systems for wireless sensors under performance and reliability constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(3):650–661, 2013. [cited at p. 15, 18]

[81] F. Berthier, E. Beigne, P. Vivet, and O. Sentieys. Power gain estimation of an event-driven wake-up controller dedicated to wsns microcontroller. *13th IEEE International NEW Circuits And Systems conference*, 2015. [cited at p. 16]

[82] Y. Zatout. Using wireless technologies for healthcare monitoring at home: A survey. In *IEEE 14th International Conference on e-Health Networking, Applications and Services*, pages 383–386, 2012. [cited at p. 16]

[83] A. Nechibvute, A. Chawanda, and P. Luhanga. Piezoelectric energy harvesting devices: an alternative energy source for wireless sensors. *Smart Materials Research*, 2012, 2012. [cited at p. 17]

[84] H. Jeong, J. Kim, and Y. Yoo. Adaptive broadcasting method using neighbor type information in wireless sensor networks. *Sensors*, 11(6):5952–5967, 2011. [cited at p. 17]

[85] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu. An estimation of sensor energy consumption. *Progress In Electromagnetics Research B*, 2009. [cited at p. 17]

[86] Q. Wang, M. Hempstead, and W. Yang. A realistic power consumption model for wireless sensor network devices. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 1, pages 286–295, 2006. [cited at p. 17]

[87] V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 2002. [cited at p. 17]

[88] C. Alippi, G. Anastasi, and M. Di Francesco, M.and Roveri. Energy management in wireless sensor networks with energy-hungry sensors. *IEEE Instrumentation & Measurement Magazine*, 12(2):16–23, 2009. [cited at p. 17]

[89] M.D. Scott, B.E. Boser, and K.S.J. Pister. An ultralow-energy adc for smart dust. *IEEE Journal of Solid-State Circuits*, 38(7):1123–1129, 2003. [cited at p. 17]

[90] N. Verma and A.P. Chandrakasan. An ultra low energy 12-bit rate-resolution scalable sar adc for wireless sensor nodes. *IEEE Journal of Solid-State Circuits*, 42(6):1196–1205, 2007. [cited at p. 17]

[91] M. Kohvakka, M. Hannikainen, and T.D. Hamalainen. Ultra low energy wireless temperature sensor network implementation. In *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 801–805, 2005. [cited at p. 17]

[92] B. Lo and G.-Z. Yang. Architecture for body sensor networks. In *Perspectives in Pervasive Computing*, pages 23–28, 2005. [cited at p. 17]

[93] K. Mikhaylov and J. Tervonen. Optimization of microcontroller hardware parameters for wireless sensor network node power consumption and lifetime improvement. In *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, pages 1150–1156, 2010. [cited at p. 17]

[94] D. Singh, S. Sai Prashanth, S. Kundu, and A. Pal. Low-power microcontroller for wireless sensor networks. In *TENCON 2009 - 2009 IEEE Region 10 Conference*, pages 1–6, 2009. [cited at p. 17]

[95] A. De La Piedra, A. Braeken, and A. Touhafi. Sensor systems based on fpgas and their applications: a survey. *Sensors*, 12(9):12235–12264, 2012. [cited at p. 17]

[96] G. Mathur, P. Desnoyers, P. Chukiu, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. *ACM Transactions on Sensor Networks*, 5(4), 2009. [cited at p. 17]

[97] M.A.M. Vieira, C.N. Coelho, Jr. da Silva, D.C., and J.M. da Mata. Survey on wireless sensor network devices. In *IEEE Conference Emerging Technologies and Factory Automation*, volume 1, pages 537–544, 2003. [cited at p. 17]

[98] A. Didioui, C. Bernier, L. Q. V. Tran, and O. Sentieys. Envadapt: An energy-aware simulation framework for power-scalable transceivers for wireless sensor networks. In *Proceedings of 20th European Wireless Conference*, pages 1–6, 2014. [cited at p. 18]

[99] J. Crols and M. Steyaert. *CMOS wireless transceiver design*, volume 411. Springer Science & Business Media, 2013. [cited at p. 18]

[100] D. Feng, C. Jiang, G. Lim, Jr. Cimini, L.J., G. Feng, and G.Y. Li. A survey of energy-efficient wireless communications. *IEEE Communications Surveys & Tutorials*, 15(1):167–178, 2013. [cited at p. 18]

[101] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000. [cited at p. 18]

[102] S. Roundy, P. K. Wright, and J. Rabaey. A study of low level vibrations as a power source for wireless sensor nodes. *Computer Communications*, 26(11):1131–1144, 2003. [cited at p. 18]

[103] J. Kymissis, C. Kendall, J. Paradiso, and N. Gershenfeld. Parasitic power harvesting in shoes. In *Second International Symposium on Wearable Computers Digest of Papers*, pages 132–139, 1998. [cited at p. 18]

[104] G. Pistoia. *Battery operated devices and systems: From portable electronics to industrial products*. Elsevier, 2008. [cited at p. 18]

[105] M.K. Stojcev, M.R. Kosanovic, and L.R. Golubovic. Power management and energy harvesting techniques for wireless sensor nodes. In *9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*, pages 65–72, 2009. [cited at p. 18]

[106] M. Healy, T. Newe, and E. Lewis. Wireless sensor node hardware: A review. In *IEEE Sensors*, pages 621–624, 2008. [cited at p. 18]

[107] A. Sinha and A. Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design & Test of Computers*, 18(2):62–74, 2001. [cited at p. 18]

[108] `www.arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth`, 2015. [cited at p. 19]

[109] `www.ti.com/lit/ug/slau227f/slau227f.pdf`, 2015. [cited at p. 19]

[110] `wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf`, 2015. [cited at p. 19]

[111] `www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf`, 2015. [cited at p. 19]

[112] `www.openpicus.com`, 2015. [cited at p. 19, 35, 89, 90]

[113] `pt.slideshare.net/josejaimeariza/sun-spot-wireless-sensors-networks`, 2015. [cited at p. 19]

[114] `www4.ncsu.edu/~kkolla/CSC714/datasheet.pdf`, 2015. [cited at p. 19]

[115] `www.aragosystems.com/images/stories/WiSMote/Doc/wismote_en.pdf`, 2015. [cited at p. 19]

[116] `www.aragosystems.com/en/wisnet-item/wismotemini.html`, 2015. [cited at p. 19]

[117] O. Mokrenko, S. Lesecq, W. Lombardi, D. Puschini, C. Albea, and O. Debicki. Dynamic power management in a wireless sensor network using predictive control. In *40th Annual Conference of the IEEE Industrial Electronics Society*, pages 4756–4761, 2014. [cited at p. 21, 100]

[118] O. Mokrenko, M. I. Vergara-Gallego, L. Lombardi, L. Lesecq, D. Puschini, and C. Albea. Design and implementation of a predictive control strategy for power management of a wireless sensor network. *The 14th annual European Control Conference*, 2015. [cited at p. 21, 75, 100, 102]

[119] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice-a survey. *Automatica*, 25(3):335–348, 1989. [cited at p. 22, 25]

[120] J. M. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002. [cited at p. 22, 23, 31]

[121] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. [cited at p. 22]

[122] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009. [cited at p. 22]

[123] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *AIChE Symposium Series*, volume 93, pages 232–256, 1997. [cited at p. 22]

[124] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003. [cited at p. 22]

[125] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999. [cited at p. 22]

[126] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013. [cited at p. 22]

[127] F. Allgöwer and A. Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser Base, 2000. [cited at p. 22]

[128] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems*, 20(3):38–52, 2000. [cited at p. 22]

[129] A. Bemporad and M. Morari. Predictive control of constrained hybrid systems. In *Nonlinear model predictive control*, pages 71–98. Springer, 2000. [cited at p. 22]

[130] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999. [cited at p. 22, 29]

[131] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999. [cited at p. 23, 25]

[132] L. A.F. Santos, M. de O. Marques, and I. S. Bonatti. Power-aware model for wireless sensor nodes. *University of Campinas*, 2008. [cited at p. 26]

[133] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides. Model-based design exploration of wireless sensor node lifetimes. In *Wireless Sensor Networks*, pages 277–292. Springer, 2007. [cited at p. 26]

[134] R. C. Cope and Y. Podrazhansky. The art of battery charging. In *The Fourteenth Annual Battery Conference on Applications and Advances*, pages 233–235, 1999. [cited at p. 26]

[135] A. Del Pia, S. S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in np. *arXiv preprint arXiv:1407.4798*, 2014. [cited at p. 29]

[136] P. S. Sausen, J. R. de Brito Sousa, M. A. Spohn, A. Perkusich, and A. M. N. Lima. Dynamic power management with scheduled switching modes. *Computer Communications*, 31(15):3625–3637, 2008. [cited at p. 32]

[137] S. Du, A. K. Saha, and D. B. Johnson. Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In *26th IEEE International Conference on Computer Communications*, pages 1478–1486, 2007. [cited at p. 33]

[138] B. Stelte. An event management communication system for wireless sensor networks. In *IFIP / IEEE International Symposium on Integrated Network Management*, pages 630–633, 2011. [cited at p. 33]

[139] `www.farnell.com/datasheets/1666650.pdfand1666648.pdf`, 2015. [cited at p. 35, 91]

[140] `hackspark.fr/fr/1-5w-solar-panel-81x137.html`, 2015. [cited at p. 35]

[141] R. L. Williams, D. A. Lawrence, et al. *Linear state-space control systems*. John Wiley & Sons, 2007. [cited at p. 37]

[142] `www.mosek.com`, 2015. [cited at p. 38, 93]

[143] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, 2004. [cited at p. 38]

[144] H. Chen, L. Cui, and S. Lu. An experimental study of the multiple channels and channel switching in wireless sensor networks. In *Proceedings of the 4th international symposium on innovations and real-time applications of distributed sensor networks (IRADSN)*, pages 54–61, 2009. [cited at p. 46]

[145] J. Saraswat and P. P. Bhattacharya. Effect of duty cycle on energy consumption in wireless sensor networks. *International Journal of Computer Networks and Communications*, 5(1), 2013. [cited at p. 46]

[146] J. P. Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015. [cited at p. 47]

[147] E. Castillo, A. J. Conejo, P. Pedregal, R. Garcia, and N. Alguacil. *Building and solving mathematical programming models in engineering and science*, volume 62. John Wiley & Sons, 2011. [cited at p. 47]

[148] A. Bemporad, F. Borrelli, M. Morari, et al. Model predictive control based on linear programming˜ the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002. [cited at p. 47]

[149] C. H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981. [cited at p. 51]

[150] Intel xeon processor e5620, `ark.intel.com`, 2015. [cited at p. 51, 78]

[151] J. Lunze and F. Lamnabhi-Lagarrigue. *Handbook of hybrid systems control: theory, tools, applications*. Cambridge University Press, 2009. [cited at p. 53, 54]

[152] O. Mokrenko, C. Albea Sanchez, L. Zaccarian, and S. Lesecq. Feedback scheduling of sensor network activity using a hybrid dynamical systems approach. *54th IEEE Conference on Decision and Control*, 2015. [cited at p. 53, 101]

[153] H. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2):161–167, 1966. [cited at p. 54]

[154] F. W. Vaandrager and J. H. Van Schuppen. *Hybrid Systems: Computation and Control: Second International Workshop, HSCC'99, Berg en Dal,*. Springer Science & Business Media, 1999. [cited at p. 54]

[155] P. J. Antsaklis and A. Nerode. Special issue on hybrid control systems. *IEEE Transactions on Automatic Control*, 43, 1998. [cited at p. 54]

[156] A. S. Morse, S. S. Pantelides, S. Sastry, and J. M. Schumacher. A special issue on hybrid systems. *Automatica*, 35(3), 1999. [cited at p. 54]

[157] W. P. M. H. Heemels, B. De Schutter, J. Lunze, and M. Lazar. Stability analysis and controller synthesis for hybrid dynamical systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1930):4937–4960, 2010. [cited at p. 54]

[158] R. Postoyan, N. van de Wouw, D. Nesic, and W.P.M.H. Heemels. Tracking control for nonlinear networked control systems. *IEEE Transactions on Automatic Control*, 59(6):1539–1554, 2014. [cited at p. 54]

[159] K. Aihara and H. Suzuki. Theory of hybrid dynamical systems and its applications to biological and medical systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1930):4893–4914, 2010. [cited at p. 54]

[160] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012. [cited at p. 54, 67]

[161] A. J. Van Der Schaft, J. M. Schumacher, A. J. van der Schaft, and A. J. van der Schaft. *An introduction to hybrid dynamical systems*, volume 251. Springer London, 2000. [cited at p. 55]

[162] R. Goebel, R.G. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009. [cited at p. 55]

[163] R. Alur, C. Courcoubetis, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. In *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*, pages 329–351, 1994. [cited at p. 55]

[164] R. Sanfelice, D. Copp, and P. Nanez. A toolbox for simulation of hybrid systems in matlab/simulink: Hybrid equations (hyeq) toolbox. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 101–106, 2013. [cited at p. 70]

[165] O. Mokrenko, M.-I. Vergara-Gallego, W. Lombardi, S. Lesecq, and C. Albea. Wsn power management with battery capacity estimation. In *13th IEEE International NEW Circuits And Systems conference*, 2015. [cited at p. 75, 102]

[166] D. Di Domenico, G. Fiengo, and A. Stefanopoulou. Lithium-ion battery state of charge estimation with a kalman filter based on a electrochemical model. In *IEEE International Conference on Control Applications*, pages 702–707, 2008. [cited at p. 86]

[167] S. Piller, M. Perrin, and A. Jossen. Methods for state-of-charge determination and their applications. *Journal of Power Sources*, 96(1):113–120, 2001. [cited at p. 86, 87]

[168] W. Waag, C. Fleischer, and D. U. Sauer. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources*, 258:321–339, 2014. [cited at p. 86, 88]

[169] W. Junping, G. Jingang, and D. Lei. An adaptive kalman filtering based state of charge combined estimator for electric vehicle battery pack. *Energy Conversion and Management*, 50(12):3182–3186, 2009. [cited at p. 86]

[170] R. Rao, S. Vrudhula, and D. N. Rakhmatov. Battery modeling for energy aware system design. *Computer*, 36(12):77–87, 2003. [cited at p. 87]

[171] Learn about batteries, `batteryuniversity.com`, 2015. [cited at p. 87]

[172] `www.aosong.com/en/products/details.asp?id=109`, 2015. [cited at p. 89]

[173] M. Louvel and F. Pacull. Linc: A compact yet powerful coordination environment. In *Coordination Models and Languages*, pages 83–98, 2014. [cited at p. 92]

[174] L.-F. Ducreux, C. Guyon-Gardeux, S. Lesecq, F. Pacull, and S. R. Thior. Resource-based middleware in the context of heterogeneous building automation systems. In *38th Annual Conference on IEEE Industrial Electronics Society*, pages 4847–4852, 2012. [cited at p. 92]

[175] Picos: A python interface for conic optimization solvers, `picos.zib.de`, 2012. [cited at p. 93]

List of Publications

International Journals

1. Mokrenko O., Vergara-Gallego M.I., Albea Sanchez C., Zaccarian L., Lesecq S. Hybrid dynamic modeling and control for a WSN with harvesting energy. In preparation for *Journal in preparation for IEEE Transactions on Control Systems Technology.*

International Conferences

1. Vergara-Gallego M.I., Mokrenko O., Louvel M., Lesecq S. Implementation of an Energy Management Control Strategy for WSNs using the LINC Middleware. Accepted in *EWSN'16: International Conference on Embedded Wireless Systems and Networks*, Graz, Austria, 2016.

2. Mokrenko O., Albea Sanchez C., Zaccarian L., Lesecq S. Automatic assignment of sensor node activity using Hybrid Dynamic System approach. Accepted in *CDC'15: 54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015.

3. Mokrenko O., Vergara-Gallego M.I., Lombardi W., Lesecq S., Puschini D., Albea C. Design and Implementation of a Predictive Control Strategy for Power Management of a Wireless Sensor Network. In *ECC'15: 14th annual European Control Conference*, Linz, Austria, 2015.

4. Mokrenko O., Vergara-Gallego M.I., Lombardi W., Lesecq S., Albea C. WSN Power Management with Battery Capacity Estimation. In *NEWCAS'15: 13th IEEE International NEW Circuits And Systems conference*, Grenoble, France, 2015.

5. Mokrenko O., Lesecq S., Lombardi W., Puschini D., Albea C., Debicki O. Dynamic Power Management in a Wireless Sensor Network using Predictive

Control. In *IECON'14: 40th Annual Conference of the IEEE Industrial Electronics Society*, Dallas, TX.-USA, 2014.

Titre : Gestion de l'énergie d'un réseau de capteurs au niveau application

Résumé

L'énergie est une ressource clé dans les réseaux de capteurs sans fil (WSNs), en particulier lorsque les nœuds capteurs sont alimentés par des batteries. Cette thèse s'inscrit dans le contexte de la réduction de la consommation de l'énergie d'un réseau de capteurs au niveau application construite au-dessus de ce réseau, grâce à des stratégies de contrôle, en temps réel et de façon dynamique. La première stratégie de gestion de l'énergie considérée s'appuie sur le contrôle prédictif (MPC). Le choix de MPC est motivé par les objectifs globaux qui sont de réduire la consommation d'énergie de l'ensemble des nœuds capteurs tout en assurant un service donné, nommé *mission*, pour le réseau de capteurs. En outre, un ensemble de contraintes sur les variables de contrôle binaires et sur les nœuds capteur doit être rempli. La deuxième stratégie de gestion de l'énergie au niveau de l'application utilise une approche de contrôle hybride (HDS). Ce choix est motivé par la nature inhérente du système WSN qui est par essence hybride, en particulier lorsque l'on s'intéresse à la gestion de l'énergie. La nature hybride vient essentiellement de la combinaison de processus physiques continus tels la charge et décharge des batteries des nœuds; tandis que la partie discrète est liée à la modification des modes de fonctionnement et l'état *Inaccessible* des nœuds. Les stratégies proposées sont évaluées et comparées en simulation sur des différents scenarios réalistes. Elles ont aussi été mises en œuvre sur un banc d'essai réel et les résultats obtenus ont été discutés.

Mots-Clés : Réseaux de capteurs sans fil, Gestion de l'énergie, Contrôle prédictif, Contrôle hybride

Title: Energy management of a Wireless Sensor Network at application level

Abstract

Energy is a key resource in Wireless Sensor Networks (WSNs), especially when sensor nodes are powered by batteries. This thesis is investigates how to save energy of the whole WSN, at the application level, thanks to control strategies, in real time and in a dynamic way. The first energy management strategy investigated is based on Model Predictive Control (MPC). The choice of MPC is motivated by the global objectives that are to reduce the energy consumption of the set of sensor nodes while ensuring a given service, named *mission*, for the sensor network. Moreover, a set of constraints on the binary control variables and on the sensor modes must be fulfilled. The second energy management strategy at the application level is based on a Hybrid Dynamical System (HDS) approach. This choice is motivated by the hybrid inherent nature of the WSN system when energy management is considered. The hybrid nature basically comes from the combination of continuous physical processes, namely, the charge / discharge of the node batteries; while the discrete part is related to the change in the functioning modes and the *Unreachable* condition of the nodes. The proposed strategies are evaluated and compared in simulation on a realistic test-case. Lastly, they have been implemented on a real test-bench and the results obtained have been discussed.

Key-Words: Wireless Sensor Network, Energy Management, Model Predictive Control, Hybrid Dynamic System

Laboratoires : CEA-Leti, MINATEC - 17 rue des Martyrs, 38054 Grenoble Cedex 9, France
LAAS - 7, avenue du Colonel Roche, 31031 Toulouse Cedex 4, France