Energy-aware control and communication co-design in wireless net-worked control systems

Nicolas Cardoso de Castro

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Automatique et Productique**

Arrêté ministériel : 7 août 2006

Présentée par

Nicolas CARDOSO DE CASTRO

Thèse dirigée par **Carlos CANUDAS DE WIT**
et codirigée par **Federica GARIN**

préparée au sein du laboratoire **GIPSA-Lab, Département Automatique**,
du centre de recherche **INRIA Grenoble Rhône-Alpes**
et de l'école doctorale **Électronique, Électrotechnique, Automatique et Traitement du Signal**

Energy-aware control and communication co-design in wireless Networked Control Systems

Co-conception contrôle/communication pour économiser l'énergie dans les systèmes commandés en réseau sans-fil

Thèse soutenue publiquement le **4 Octobre 2012**,
devant le jury composé de :

Mazen ALAMIR, Président
Directeur de Recherche CNRS, GIPSA-Lab (Grenoble, France)
Maurice HEEMELS, Rapporteur
Full Professor, Eindhoven University of Technology (Eindhoven, Pays-Bas)
Luca SCHENATO, Rapporteur
Associate Professor, University of Padova (Padoue, Italie)
Antonio BICCHI, Examinateur
Full Professor, University of Pisa (Pise, Italie)
Antoine CHAILLET, Examinateur
Maître de Conférences, Université Paris Sud - L2S (Paris, France)
Suzanne LESECQ, Examinateur
Ingénieur-Chercheur, CEA-LETI MINATEC (Grenoble, France)
Daniel E. QUEVEDO, Examinateur
Associate Professor, University of Newcastle (Newcastle, Australia)

To Jessica, having you around really changed my life.

Abstract

Energy is a key resource in Networked Control Systems, in particular in applications concerning wireless networks. This thesis investigates how to save energy in wireless sensor nodes with control and communication co-design. First, this thesis reviews existing techniques and approaches that are used to save energy from a communication and a control point of view. This review is organized according to the layered communication architecture covering from bottom to top the Physical, Data Link, Network, and Application layers. Then, from the conclusion that the radio chip is an important energy consumer, a joint radio-mode management and feedback law policy is derived. The radio-mode management exploits the capabilities of the radio chip to switch to low consuming radio-modes to save energy, and to adapt the transmission power to the channel conditions. This results in an event-based control scheme where the system runs open loop at certain time. A natural trade-off appears between energy savings and control performance. The joint policy is derived in the framework of Optimal Control with the use of Dynamic Programming. This thesis solves the optimal problem in both infinite and finite horizon cases. Stability of the closed-loop system is investigated with Input-to-State Stability framework. The main conclusion of this thesis, also shown in simulation, is that cross-layer design in Networked Control System is essential to save energy in the wireless nodes.

Résumé

L'énergie est une ressource majeure dans les systèmes commandés en réseau, en particulier dans les applications concernant les réseaux sans fil. Cette thèse étudie comment économiser l'énergie dans les capteurs sans fil avec une co-conception contrôle et communication. Cette thèse examine les techniques et les approches existantes qui sont utilisées pour économiser l'énergie d'un point de vue de la communication et du contrôle. Cette étude est organisée selon une architecture de communication par couches couvrant de bas en haut les couches Physique, Liaison, Réseau, et Application. Puis, à partir de la conclusion que la puce radio est un important consommateur d'énergie, une loi conjointe de gestion des modes radio et de contrôle en boucle fermée est établie. La gestion des modes radio exploite les capacités de la puce radio à commuter dans des modes de basse consommation pour économiser l'énergie, et à adapter la puissance de transmission aux conditions du canal. Il en résulte un système de contrôle basé sur des événements où le système fonctionne en boucle ouverte à certains moments. Un compromis naturel apparaît entre l'économie d'énergie et les performances de contrôle. La loi conjointe est établie avec une formulation de contrôle optimal utilisant la Programmation Dynamique. Cette thèse résout le problème optimal dans les deux cas d'horizon infini et fini. La stabilité du système en boucle fermée est étudiée avec la formulation *Input-to-State Stability* (ISS). La principale conclusion de cette thèse, également illustrée en simulation, est que la conception à travers différentes couches dans les systèmes commandés en réseau est essentielle pour économiser l'énergie dans les noeuds d'un réseau sans fil.

Acknowledgment

First and foremost, I would like to thank my advisor, Carlos Canudas de Wit, for his trust and advice. His knowledge and his intuition greatly contributed to the realisation of this thesis. I also warmly thank my co-advisor, Federica Garin, for her support and friendship. While Carlos headed me to the correct direction to dig, she was an unvaluable help in the process of digging.
I am grateful to Carlos for the opportunity he gave me to participate in the work of the NeCS team, and especially in the European project FeedNetBack. This gave me the chance to meet great people at various places, including Venezia, Stockholm, Eindhoven, Sevilla, Trento, and Montréal.

A special thank goes to Daniel E. Quevedo for hosting me at Newcastle University, Australia. During the three months I spent there, he gave me a precious help in the progression of this work. I acknowledge the people there that made my stay enjoyable, especially my roommates, Mauricio and Carmen. I am grateful to Inria for the financial support that made this collaboration possible. I also thank Daniel for accepting to review this thesis.

I am very grateful to Karl Henrik Johansson for hosting me one week at KTH University, Sweden. My stay there and the numerous discussions I had were enriching.

I also thank the members of the jury for taking the time to read this document and for the useful discussions and feedback.

I am very grateful to João Miranda Lemos, who gave me my first insight to the research world and the will to continue on that direction.

I am especially grateful to the people that shared my life during these three years. Thank to Lara, for her friendship, for bringing the joy to the office, for sharing her experience, but also for keeping me updated with events occuring in Spain. Thank to Gabriel for sharing the doubts, the tricks, and the jokes. Thank to all other members of the NeCS team with whom I shared scientific discussions, technical and administrative assistance, meals, and for many, friendship.
A special thank goes to Elodie, Florence and Caroline, who made administrative technicalities almost enjoyable.
Working at Inria was pleasant, and I thank all the people that contributed to the nice environment, inluding people from the Library, IT, Human Resources and Communications departments.

Finally, I acknowledge the support of my friends who were nice enough to occupy my mind with everything but scientific materials during weekends and vacations. I am also grateful to my parents and my brother for the cheering in difficult moments and for their love and trust during all these years.

Contents

List of Figures

List of Tables

List of Acronyms

ACK Acknowledgment 71

CSMA Carrier Sense Multiple Access 33

EBC Event-Based Control 108

DMS Dynamic Modulation Scaling 115

GISpS Globally Input-to-State practically Stable 131

ISS Input-to-State Stability 113

ISpS Input-to-State practical Stability 94

LQ Linear Quadratic 98

LQG Linear Quadratic Gaussian 30

MAC Media Access Control 115

MPC Model Predictive Control 113

NCS Networked Control System 112

QoS Quality of Service 35

TDMA Time Division Multiple Access 30

WSN Wireless Sensor Network 116

ZOH Zero Order Holder 48

List of Publications

8. **Joint Radio chip management and Event-Based Control for wireless Networked Control Systems**
   Nicolas Cardoso de Castro, Daniel E. Quevedo, Federica Garin and Carlos Canudas de Wit. *Journal Paper under preparation*.

7. **Optimal Radio-Mode Switching for wireless Networked Control using Dynamic Programming**
   Nicolas Cardoso de Castro, Federica Garin and Carlos Canudas de Wit. *Book Chapter accepted in Lecture Notes in Control and Information Sciences, Springer*.

6. **Smart Energy-Aware Sensors for Event-Based Control**
   Nicolas Cardoso de Castro, Daniel E. Quevedo, Federica Garin and Carlos Canudas de Wit. *In 51st IEEE Conference on Decision and Control (CDC 2012), pp.7224-7229, 10-13 December, 2012.*
   http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6426482

5. **Energy-Aware wireless Networked Control Using Radio-Mode Management**
   Nicolas Cardoso de Castro, Carlos Canudas de Wit, Federica Garin. *In American Control Conference (ACC 2012), pp.2836-2841, 27-29 June, 2012.*
   http://hal.inria.fr/hal-00679683

4. **On Energy-Aware Communication and Control Co-design in wireless Networked Control Systems**
   Nicolas Cardoso de Castro, Carlos Canudas de Wit, Karl Henrik Johansson. *In 2nd IFAC Workshop on Estimation and Control of Networked Systems (NecSys 2010), pp.49-54, 13-14 September, 2010.*
   http://www.ifac-papersonline.net/Detailed/44641.html

3. **Design methodologies for event-based control systems**
   Nicolas Cardoso de Castro, Federica Garin, Carlos Canudas de Wit, Peter Hokayem, John Lygeros, Carlos Vivas, Pablo Millán, Luis Orihuela, Francisco R. Rubio, Ubaldo Tiberi, Carlo Fischione, Karl Henrik Johansson. *Technical report, Deliverable D05.04 FeedNetBack project, 31 August, 2011.*
   http://www.feednetback.eu/public-deliverables

2. **Source and channel coding algorithms for energy-limited networks**
   Nicolas Cardoso de Castro, Carlos Canudas de Wit and Karl Henrik Johansson. *Technical report, Deliverable D05.02 FeedNetBack project, 20 July, 2010.*
   http://www.feednetback.eu/public-deliverables

1. **Model Predictive control of Depth of Anaesthesia: Guidelines for controller configuration**
   Nicolas Cardoso de Castro and João Miranda Lemos. *In 30st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2008) pp.5822-5825, 20-25 August, 2008*
   http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4650538

Introduction

Background and motivation

Networked Control Systems (NCSs) are control systems where the communication between the sensors, the controller and the actuators occurs through a network, see Figure I and [Hespanha et al., 2007, Zhang et al., 2001, Heemels and van de Wouw, 2010]. The network can be wired or wireless. This kind of configuration arises naturally from the development of digital architecture with remote control, and is mainly intended to describe either a control system where the loop is closed over a communication network, or a system composed of several sensors and actuators sharing a network. Some illustrative scenarios are given in Figure II. NCSs can be encountered in a wide range of domains, such as automated highway systems (traffic control), autonomous underwater vehicles, remote surgery, industrial automation, swarm robotics, communication networks, Wireless Sensor Networks (WSNs), smart grids, water distribution, smart buildings.

While automatic control originally considers continuous systems, the rise of digital computation brought a large interest in discrete-time systems. However, although discrete-time systems are better suited for realistic digital control problems, the field of discrete-time control is based on the paradigm that the exact information is instantaneously available from the sensors to the controller and from the controller to the actuators. The effects of various practical imperfections, such as delay, perturbations, additive noise or input saturation, have been studied to increase the quality of the closed loop in real systems. However the imperfections caused by a network are generally much more restrictive than the ones under consideration in the case of dedicated-wire systems. When considering a network, not only the information may suffer from variable delays or dropouts, but also it is quantised and generally divided into finite-size packets which necessitate a reconstruction on the receiver side. These stiff restrictions may not be tackled by the existing discrete-time approaches. One has to step into the study of network related communication problems which makes NCS the fusion of Communication/Information Theory with Control Theory, and brings new challenging problems.

Indeed, NCSs are not just an extension of discrete-time systems. From a control view point, including the network imperfections in the model prevents the use of well-known control analysis and design tools. On the other hand, communication view point considers the transmission of data generated without being concerned with the content of the information. A separated use of both fields would result in very conservative behaviour. Indeed, it is possible to design a control application that guarantees some performance on top of a network designed to provide some given Quality of Service (QoS). However in this approach, the control design is based on the worst case scenario of the underneath communication channel and then provides poor control performance. This motivates the use of co-design control strategies, taking into account jointly the communication channel and the control application, which is addressed in this thesis.

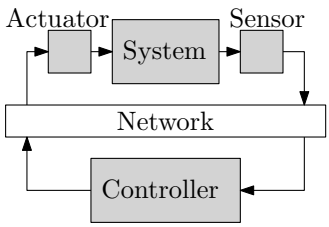
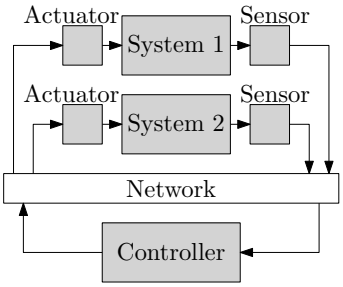
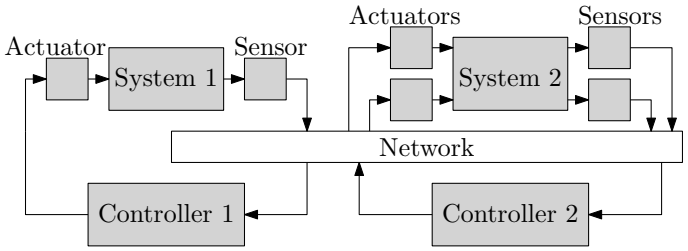


***Figure I** – Block diagram of a typical Networked Control System: the sensor sends its measurement to the controller over the network, which sends the control input to the actuator over the same network.*



(a) Example of several systems sharing the same network and a single controller.



(b) Example of several systems sharing the same network; system 1 has a direct link between controller 1 and the actuator; system 2 has several sensors and actuators.

***Figure II** – Illustrative scenarios of NCS architectures.*

Another important motivation of this thesis is to save energy in a NCS. Here again, both communication and control fields have addressed this topic separately whereas a control and communication co-design may save more energy, as it will be discussed later.

Networked Control Systems have gained in interest over the last ten years for the following reasons:

Lower costs On the one hand, the sensor nodes are typically simple embedded pieces of hardware. They are all similar and produced in large quantities to decrease the costs. On the other hand, the use of a network to link the sensors and the actuators to the controller reduces the amount of wires from the case where dedicated wires are used. This is especially true in the case of wireless network and still holds when comparing, for instance, serial bus to dedicated wires. The use of a network also decreases the installation costs.

Ease of installation, reconfiguration and maintenance In the case of a wired configuration, the use of a network eases the installation and reconfiguration because the wires are no longer specific to a sensor or an actuator, and can be placed and moved anywhere in the network. Of course, the possibility to change the placement of the nodes depends on the topology of the network. In the case of a wireless configuration, the nodes are in general battery-driven and can be placed anywhere. Especially they can be deployed in harsh environments where accessibility is limited or difficult. Maintenance is eased by the capacity of the network to diagnose the problems on links, connection devices (*e.g.* routers) or nodes.

Large flexibility The use of a network increases the flexibility of a system, the number and the position of the nodes can be changed with less effort than in the dedicated wires case. In case of failure of a node or a link, the network can dynamically reconfigure itself to maintain the communication with the other nodes. Moreover, a network also brings flexibility for the final application. In a NCS, the control can be distributed, or computed remotely, for instance over Internet. A network can mix wired and wireless nodes and include easily any type of devices to tackle a wide range of applications such as forest fire monitoring or highway traffic control.

However, the use of a network in a control system also introduces issues generally omitted in Control Theory:

Varying sampling interval Whereas data is supposed to be available at given sampling instant in discrete control, possibly with some modeled delays, the wireless channel causes varying transmission intervals due to the channel condition (*e.g.* fading, interferences) and to the limited bandwidth which has to be shared among several nodes, causing *e.g.* congestion or failures.

Packet loss The transmission may not succeed, especially in wireless channel, resulting in data loss.

Quantisation The use of computer-based control schemes makes digital data very common. The precision of the quantisation depends on the final application, the available channel bandwidth and the energy consumption that one can allow to transmit the data. The latter is mostly true with wireless channels.

Energy efficiency The Control Community gives a limited look on the energy consumption, mostly related to the energy consumed by the actuator. In wireless configurations, nodes are generally

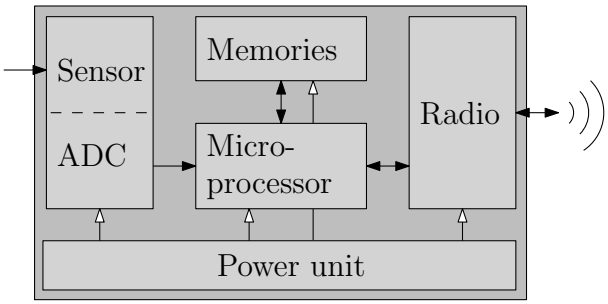


***Figure III** – Node architecture is composed of: a sensing unit (sensor and Analogue to Digital Converter), a computational unit (Micro-Processor and Memories), a Radio and a Power unit (which contains in some cases a energy-harvesting device), taken from [Akyildiz et al., 2002].*

powered on battery. The amount of energy is then limited for all the tasks (sensing, computing, communicating), not only for the actuating task. Saving energy becomes a major challenge to enlarge the life-time of the system.

In this thesis, we focus on the issue of energy-efficiency in wireless NCSs. Although saving energy might be of great interest in the wired case, it is not critical. Indeed, it is common that nodes connected to a wired network also have access to an unlimited source of energy. On the opposite, wireless nodes are battery-driven, and it is sometimes economically inefficient, or time-costly to change the batteries. Even in the presence of energy-harvesting devices, the utilisation of energy must be optimised in order to maximise the life-time of the nodes and the network.

The goal of this thesis is to identify how to save energy in a wireless node and to propose energy-aware control schemes. It is then important to know how a wireless sensor node (called hereafter *smart node*) is composed. A smart sensor node is typically packaged with a sensing unit, a processing unit, a radio and a power unit, as depicted in Figure III. The sensing unit consists in a sensor chip and an analog-to-digital converter that provides the measurement as a digital signal for the processing unit. The processing unit gathers a micro-processor, a memory, and storage capabilities; it can process the measurement received from the sensor and control the radio to send or receive information from the network. The radio is used to connect the node to the rest of the network. All the components are powered by the power unit, which includes a battery and sometimes a harvesting device, such as a solar cell unit, and a power management system. This configuration allows the sensor node to execute some process on its measurement before sending it to the controller.
It is admitted that the energy consumed by the radio represents 50% to 80% of the overall consumption of a smart sensor node, see *e.g.* [Joshi et al., 2007, Kimura and Latifi, 2005, Wen et al., 2007]. An often-quoted rule of thumb is that executing 3 million instructions is equivalent to transmitting 1000 bits at a distance of 100 meters in terms of expended energy [Mann et al., 2007]. The general idea to keep from these figures is that it is interesting to optimise the use of the radio, even if a small amount of computation effort needs to be added. The goal of the first part of this thesis is to understand in details how energy is consumed for the transmission in a radio, in order to propose schemes to reduce the consumption. Turning off the radio is not the only way to save energy: as it will be seen hereafter, one can change the radio modulation, the transmission power, among other parameters. But these techniques influence the closed-loop system and the management of the radio must therefore be taken into account at the control level by doing co-design, as proposed in this thesis.

New approaches for networked control

At first, the new challenges risen by NCSs have been addressed mainly under the existing frameworks of stochastic, robust or adaptive control. The deep knowledge of design and analysis tools available in these fields allows to extend these tools to take into account the specificities of NCSs.

Stochastic control is mostly used to establish the stability of the closed-loop system in spite of the presence of a network [Antunes et al., 2012b, Donkers et al., 2012, Antunes et al., 2013].

Robust control aims to design feedback controllers that ensure closed-loop performance and stability in spite of the presence of uncertainties [Wu et al., 2010]. In this approach the influence of the network is considered as a bounded modeling error leading to very conservative designs.

Other works also consider the network imperfections bounded in order to derive closed-loop stability ([Chaillet and Bicchi, 2008], or subject to input/output constraints ([Bemporad, 1998]).
A different approach is considered in [Quevedo and Nešić, 2012] where a controller is derived using the framework of Model Predictive Control (MPC). An optimisation problem to minimize the state deviation and the control energy is solved over a finite horizon but only the first element of the solution is applied to the system and the whole procedure is repeated again on the following sampling instant in a receding manner. Robustness is brought to the scheme by transmitting all the control inputs computed over the finite horizon rather than only the one to be applied, so that the actuator can use the buffered control input when transmission fails. Assuming that the dropouts follow a Markovian process with given constraints on the loss probabilities, authors in [Quevedo and Nešić, 2012] prove the stability of their scheme. Even though the dynamic behaviour of the network depends on exogenous variables that are stochastic or difficult to model, robust control does not benefit from the knowledge one has about the influence of the network which some effects can be measured.

Adaptive control uses the fact that one can model the network with uncertainties, to adapt the controller to the network state. For instance, asynchronous packet arrivals can be tackled by considering adaptive sampling interval. With this approach, the control input is updated on the base of a varying period which is adapted in order to meet the network constraints. Another approach to deal with asynchronous packet arrivals consists in modeling the arrivals as varying delays. This modulation is used to prove the stability of the loop closed with a static feedback ([Fridman et al., 2004]) or the design of adaptive controllers ([Colandairaj et al., 2007, Raimo and Vantaski, 2007]).

Although the use of stochastic, robust and adaptive control for NCSs is an efficient approach to tackle new challenges brought by the presence of a network, it may be difficult to extend these techniques with the benefit that can be brought by the Communication Theory. An illustrative example of the limitation that these techniques maintain is the classical synchronous control view point. Synchronous control may prevent to model certain details about the communication channel. Thus, part of the Community of NCS has shown a real interest to relax the synchronous paradigm that is not necessarily a realistic assumption when several nodes are distributed over a network. Deep focus has been devoted to intermittent, or event-based, control and estimation, *i.e.* setups where the communications between the nodes are triggered by local decisions or events rather than by clock ticks. This causes the measurements or the control inputs to not be available at some undetermined time (see *e.g.* [Hu et al., 2007, Cogill, 2009, Gupta et al., 2006, Quevedo et al., 2010, Cogill et al., 2007, Imer and Basar, 2010, Sinopoli et al., 2004, Xu and Hespanha, 2005]). Author in [Cogill, 2009] derives a policy to decide when to send measurements to the controller (or control inputs to the system in the case this link is more costly) and the optimal associated state-feedback. The event-based policy tested on an example is less efficient than a periodic policy sending more information, but more efficient than a policy sending the same amount of samples at random time instants. Authors in [Imer and Basar, 2006] consider a different setup where the controller has to choose between either measuring the system or controlling it. They derive an offline policy using an optimal control framework. Under quadratic cost function, the optimal choice to minimize the state deviation appears to be a linear threshold between controlling and measuring. An Linear Quadratic Gaussian (LQG) controller taking into account quantisation and packet losses is derived in [Trivellato and Benvenuto, 2009]. The controller also decides for the number of bits to be sent (the packet length), which can be related to the transmission duration, *i.e.* the time spent in transmitting mode. In the case of a stable system with small bandwidth, results are close to optimal, despite the quantisation and packet loss. But for unstable system, a larger bandwidth is needed. Authors in [Xu and Hespanha, 2004] use Dynamic Programming to derive optimal communication in NCS. In the case of discrete Linear Time Invariant (LTI) system with fixed delay, the optimal policy turns out to be deterministic. The setup is however different from what we are interested in, since [Xu and Hespanha, 2004] focuses on the estimation problem (since control performances are related to the quality of the estimates) assuming that every node is in charge of broadcasting its state to the other nodes in the network.
Different setups consider several sensor nodes where only one node can transmit at a time, see *e.g.* [Gupta et al., 2006], or setups where the sensor nodes can adapt the transmission power of the radio chip to face bad channel conditions, or to save energy, see *e.g.* [Quevedo et al., 2010]. A drawback about Event-Based Control (EBC) is that the nodes must monitor the event in order to trigger a communication when it is needed. This implies that, even if the radio can be turned off, a part of the smart node still needs to be working. More energy can be saved by relaxing the monitoring of the events. This is done in self-triggered control. Before switching off, the node decides when will be the next sensing instant to update the control input. The whole node can be turned off, except a timer that

generates an interrupt request to turn on the micro-processor and the rest of the node. This approach is different in the fact that the system is running open loop until the next triggering time instant, without possibility to react to unpredictable perturbations (contrary to EBC which keeps monitoring the system). A natural trade-off appears between robustness and energy efficiency.

On the other hand, the communication literature investigates different techniques to handle efficient networking communications. Among all challenges that the Communication Community is interested in, we mainly focus on energy saving. It is important to notice that the communication literature is only interested in open loop configurations.
Media Access Control (MAC) and Network protocols are topics that draw attention of the Communication Community. At the node level, many works consider the regulation or the scheduling of the traffic on the network as a function of the packet load. Namely, the energy used for transmission can be lowered when the amount of data to be transmitted is small. At the network level, relay nodes can be switched off when the traffic in the rest of the network is not congested. Another technique to save energy consists in switching off only some parts of the node, introducing the notion of energy mode management (explained in details in Chapter 2). While some contributions address setups where entire features of the node (computation, communication, sensing) are turned off (see *e.g.* [Sinha and Chandrakasan, 2001]), we restrict our attention to the case where only the radio chip is switched to low consuming modes, turning off some of its components (see [Brownfield et al., 2010, Jurdak et al., 2008]). The formers can be related to self-triggered control while the latter can be related to EBC. Indeed, in the case of radio-mode management, the node is still able to detect and react to important perturbations in the system under control. The advantage of intermediate consuming radio-mode is that they take less time to wake up than the deepest sleepy mode, allowing to save energy during narrow time windows.

The main drawback of all these techniques, in the control point of view, is that the energy is saved regardless of the nature of the data and that they do not expose parameters that the control application can adapt to meet the control performance. This implies a separation between the communication layers and the application layer (the control scheme itself).

An *ad hoc* approach would consist in designing the control application and the network configuration separately. A reasonable sampling period is chosen on arbitrary criteria. Standard discrete-time control tools are used to derive the controller. Finally the network is configured so that the QoS (typically the maximal allowable delay and the bandwidth utilisation) meets the constraints imposed by the sampling period. This leads to a very conservative design where the control performance are poor and energy savings are limited. A more relevant approach than the *ad hoc* approach to save energy while guaranteeing control performance is co-design of communication and control. Moreover, it has to be noticed that many schemes for NCS can be improved because, although they limit the amount of communications, they are not energy-aware since they require all the nodes of the network to be active. Then, we are not only interested in networked control schemes, but in the energy-efficient ones.

Contribution of this thesis

This document is divided into two main parts. The goal of the first part (Chapter 1) is to understand how energy is consumed in a NCS and to point out energy efficient control schemes in the view of communication characteristics. The review of the literature is organised according to the layered communication architecture covering from bottom to top the Physical, Data Link, Network, and Application layers. The focus is specifically given on advances that concern energy-aware management in wireless communication and control co-design. It is argued that existing work is limited to single layer approaches, with a lack of design methods taking into account several layers. Finally this part motivates a relevant technique that is addressed in the rest of this thesis as a promising method for efficient energy savings and networked control co-design.

The second part (Chapters 2-4) investigates energy-aware EBC control with radio-mode management. Indeed, the Control Community has already shown large interest in the topics of intermittent control and EBC, allowing to turn off the radio of the nodes, which is the main energy consumer, on longer time intervals than in the periodic case. While the existing literature only addresses policies using two radio-modes (`Tx`-Transmitting, and `Sleep`), this thesis considers several transmitting radio-modes and intermediate low consuming radio-modes. Indeed, to save energy, the radio chips of the wireless nodes have several radio-modes to vary the transmission power and the depth of sleep.

Increasing the transmission power, *i.e.* consuming more energy, increases the success probability of the transmission over an erasure channel. Switching the radio to low consuming modes, which prevent transmission, saves energy. However, the more components are turned off in the radio chip, the more time and energy are needed to switch back to a transmitting mode. A trade-off appears between control performance and energy savings. We derive a switched linear model that accounts for both the closed-loop system and the radio-modes. We define a cost criterion that takes into account both the control performance and the energy consumption. Then, we derive joint event-based closed-loop controller and radio-mode management policy to efficiently control the system and save energy at the smart node side.

Chapter 2 introduces the system that will be used in the rest of the thesis. It gives general comments on the approach that will be investigated in the following chapters.

Chapter 3 focuses on the infinite time horizon problem. The joint controller and mode management policy are derived under the framework of optimal control, using Dynamic Programming.

Chapter 4 considers a similar problem with finite horizon. The solution is derived using the framework of MPC with the use, again, of Dynamic Programming.

In both infinite and finite cases, stability is discussed (but only proved in the finite case using Input-to-State Stability (ISS) framework) and the schemes are illustrated with simulations.

Notation and definitions

- $\mathbb{Z}_{\geq 0}$ is the set of non-negative integers, including zero.
- $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers, including zero.
- $\mathbb{R}^n$ is the real vector space of dimension n.
- $\mathbb{R}^{n\times p}$ is the set of real-valued matrices with dimension $n \times p$.
- $\mathbb{R}^n_{\succ 0}$ is the set of real-valued symmetric positive definite matrices with dimension $n \times n$.
- $\mathbb{R}^n_{\succeq 0}$ is the set of real-valued symmetric positive semidefinite matrices with dimension $n \times n$.
- $\mathbb{M}_1 \triangleq \{1, 2, \cdots, N_1\}$ is the set of transmitting radio-modes (N_1 being the number of transmitting radio-modes).
- $\mathbb{M}_2 \triangleq \{N_1+1, N_1+2, \cdots, N_1+N_2\}$ is the set of low-consuming modes (N_2 being the number of low-consuming radio-modes).
- $\mathbb{M} \triangleq \mathbb{M}_1 \cup \mathbb{M}_2$ is the set of all radio-modes (transmitting and low-consuming ones).
- $N \triangleq N_1 + N_2$ is the number of radio-modes (transmitting and low-consuming ones).
- $\mathbb{X} \triangleq \mathbb{R}_{n_x+n_u} \times \mathbb{M}$ is the augmented state space of the system we consider in this thesis, where n_x is the dimension of the state space and n_u the dimension of the control space.
- $\mathbf{0}$ denotes a vector or a matrix filled with 0's, of appropriate dimension.
- $\mathbf{1}$ denotes a vector or a matrix filled with 1's, of appropriate dimension.
- $\mathbf{I}$ denotes the identity matrix of appropriate dimension.
- Given a random variable w taking values in $\mathbb{R}^n$, $w \sim \mathcal{N}(\nu, W)$ means that the distribution of w is Gaussian, with mean $\nu \in \mathbb{R}^n$ and variance $W \in \mathbb{R}^{n\times n}$.
- For a matrix M, $M^\top$ designates the transpose of M.
- $\min\{a; b\}$ gives the minimum between the scalar values a and b.
- $\arg\min_{a\in\mathbb{R}}\{f(a)\}$ gives the value a for which $f(a)$ is minimum. When several a give the same minimum, one of them is returned arbitrarily.

Chapter 1

Multi-layer architecture in wireless Networked Control Systems

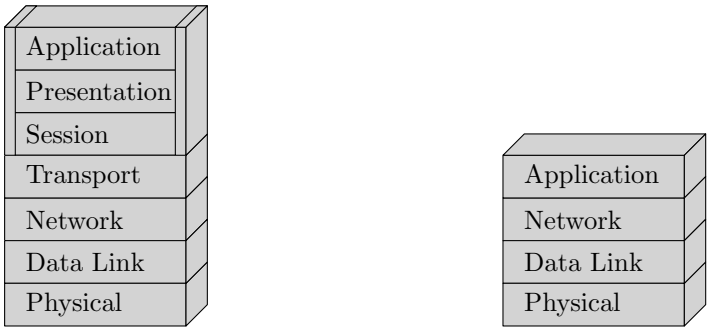
Contents

This chapter gives an insight to the effect of the communication on the control of networked systems with the focus on energy efficiency. A four layers architecture is introduced to describe how energy is used to manage data in a Networked Control System (NCS). Relevant techniques to save energy at each layer are reviewed. Finally the more promising directions to save more energy, which will be exploited in the next chapters, are presented.

1.1 A four layers architecture

We focus our attention on wireless NCSs where the nodes have computational resources, as described in Figure III. This kind of smart nodes are widely considered in Wireless Sensor Networks (WSNs) [Akyildiz et al., 2002, Correia et al., 2007], which does not necessarily include a feedback control and is mostly devoted to tracking and monitoring. Whereas reliability (*i.e.* successful packet delivery) is the main matter in WSNs, latency (*i.e.* delayed packet delivery) is also fundamental in the case of NCSs. Thus, instead of maximising the reliability, as in WSNs, communication devices in NCSs should balance reliability and latency to save energy while meeting the control requirements.

The energy consumption can be considered at different levels of the network, as it is detailed throughout this chapter. At the node level, the parameters of communication can be tuned to save the energy of the node. At the network level, the scheduling techniques maximise the life-time of the whole network. The tuning of the nodes and the network has major implications at the control level, where some constraints must be met to ensure stability and given control performance. A layered approach is presented hereafter to identify how the energy is used in a wireless network and the influence of existing energy saving techniques on the control loop.



(a) The (OSI) stack is the most complete stack, including seven layers. It is used for networked communication between computers with the support of session and encryption.

(b) The Networked Control System stack is limited to four layers, and it is better suited for the communication of the nodes in a NCS.

***Figure 1.1** – The different steps involved in a networked communication are commonly abstracted with a layer approach, grouped in a stack, where each layer has a dedicated function. The number of layers in a stack depends on the type of communication to be described.*

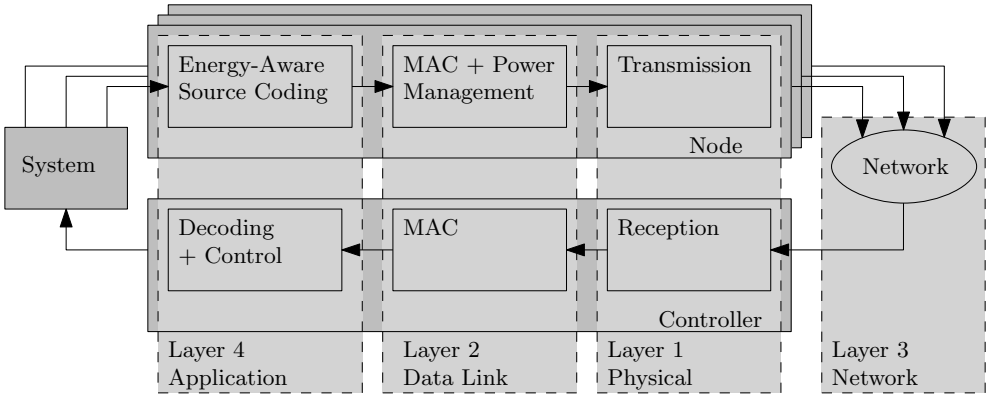


***Figure 1.2** – Illustration of the four layers of the NCS stack (Physical, Data Link (MAC), Network and Application) on a block diagram of a NCS comprising three sensor nodes taking measurements from the system and communicating via a network to the controller.*

1.1.1 The four layers stack

A layered model is commonly used in Communication Theory to describe the processes taking place in a networked communication. The standard Open System Interconnection (OSI) model is composed of seven layers, as depicted in Figure 1.1(a), even though the *Presentation* and *Session* layers are often included in the *Application* one in other communication models. The description of this layer stack is out of the scope of this section. In the context of NCS a stack comprising only four layers is relevant, named here the NCS stack; it is given in Figure 1.1(b) and described hereafter. It comprises the Physical, Data Link (Media Access Control (MAC)), Network, and Application layers. Figure 1.2 gives a block diagram representation which depicts how the four layers are related to the elements of the system.

The purpose of this chapter is to review the multi-layer architecture of NCSs in the light of the energy-use in each layer, and to point out major contributions in the area of energy-management policies, layer per layer. A specific focus is given on advances that concern energy-aware management in communication and control co-design, but we argue also that there is a lack of integrated design methods that take into account the various layers in an overall systematic way suitable for energy-aware networked control. The review of the existing works reported in the literature is organised into a bottom-to-top four layers architecture [Liu and Goldsmith, 2004].

1. **Physical layer.** This layer is in charge of the radio modulation of the digital data[1]. It concerns

[1]We recall that we focus on wireless networks.

the emission and reception processes at the radio level. In the case of an emission, certain transmission power and bit-rate are used for the transmission. It is related to the success probability of the transmission, and it is increased to face varying channel conditions or to decrease the erasure probability of critical data in steady channel conditions. The bit-rate is related to transmission delay. The bit-rate used by the source should be known at the reception. Both power and bit-rate can be controlled to save energy.

2. **Data Link (MAC) layer.** In wireless communications, the transmission medium is shared among all the nodes, and there is no dedicated path between a source and a destination. This implies that several nodes transmitting in the same time interfere with each other. The data link layer defines how to use and share the transmission medium. The MAC protocols define a set of rules to allow successful transmissions. This layer also considers the activity modes of the node, which consist of turning off some parts of the node to save energy.

3. **Network layer.** In a network with several nodes, there may be several possibilities to reach a destination from a source. The main role of the Network layer is to route the data to the destination in an energy-efficient manner in order to maximise the life-time of the whole network. The routing protocol has to adapt its strategy in order to recover in case of link failure.

4. **Application layer.** This layer concerns the source encoding and decoding, and the final application, *i.e.* the computation of the control law. Encoding is used to reduce the amount of data to be sent on the network and it consumes time and computation resources.

1.2 How to save energy?

This section reviews the four layers in order to understand how the energy is consumed and what techniques can be employed to save energy. We focus on schemes that offer controllable parameters that could be used by the control application to adapt the energy consumption.

1.2.1 Physical layer

The Physical layer consists of the hardware, *i.e.* the radio chip, used to transmit a bit stream over the network. It specifies the low-level parameters of the transmission, such as the modulation scheme, the frequency, the period and the phase of the signal, or the transmission power. There is no consideration about packet, error correction or reliability at this layer. Two major parameters influence the energy consumption: the transmission power and the bit-rate, used to modulate the data into radio waves; both parameters can be controlled.

Power control

The transmission power is generally related to the communication reliability. Increasing the power level decreases the erasure probability, up to a certain level where interference must be taken into account. Changing the transmission power has two main purposes; the first is to face varying channel conditions; the second is to increase the probability of successful transmission of critical data. In the former, more energy is consumed in order to maintain the same quality of the link while the channel throughput weakens (because of moving obstacles or perturbations or moving nodes). In the latter, more energy is consumed in order to ensure the transmission of critical data in the same transmission conditions. However, arbitrarily large power will induce interferences with other communication links and will have unmoderated energy consumption. The energy has to be balanced to save the battery life while ensuring correct transmissions. To this aim, power control can rely on the receiver observations which are fed back to the transmitter, or on the channel conditions probed by the transmitter.

Power control based on the receiver does not use channel model, it consists in either increasing the transmission power at the transmitter side when packets are not well received, or decreasing the power in case of success. The feedback loop is achieved through a predictive controller in [Quevedo and Ahlen, 2008], or simple algorithms in the MIAD Power Control in [Zurita Ares et al., 2007] (see Figure 1.3) and in the Hybrid method in [Correia et al., 2007]. In both [Quevedo and Ahlen, 2008] and [Zurita Ares et al., 2007], the bit error rate is computed at the receiver side thanks to control data embedded in the transmitted packets (such as a checksum). The receiver sends back that error measure to the transmitter which adjusts its power, whereas in

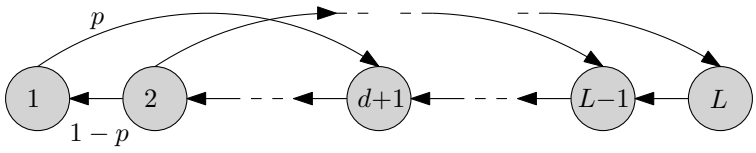


***Figure 1.3** – Markov chain model of the MIAD power control algorithm from [Zurita Ares et al., 2007]. The radio chip can be used in one of the L amplitude levels, 1 being the lowest amplitude. A packet is lost with probability p. The MIAD control scheme consists in increasing the power by d levels when a packet is dropped and decreasing the power by 1 level when the packet is successfully transmitted, as depicted by this Markov chain model.*

[Correia et al., 2007] the decision is based on Acknowledgments (ACKs) from the receiver to the transmitter.

The other approach is based on a channel model, each node controls its transmission power based on channel measurements such as the interference power, the packet error rate [Zurita Ares et al., 2007], the noise power and the signal attenuation [Correia et al., 2007]. Authors in [Zurita Ares et al., 2007, Correia et al., 2007] propose algorithms to compute the minimum transmission power ensuring correct reception, given a channel model and channel measurements. Although, in this approach, the knowledge of the receiver is also fed back to the transmitter to determine the channel state, it is different from power control based on the receiver which does not rely on channel model.

Power control needs extra communication between the receiver and the transmitter. It is shown in [Johansson et al., 2007] that in a WSN where nodes are not too far from each other (up to several hundreds of meters, depending on the setup, the scheme and the frequency), a scheme with a power control mechanism consumes more energy than a scheme with a fixed power amplitude plus a given margin to ensure correct bit error rate when the channel deteriorates. The short distances between nodes make the consumption dominated by radio circuitry instead of the transmission energy. The energy saved by decreasing the transmission power under good channel conditions is wasted by the extra communication used to decide the power amplitude, as explained in Johansson et al. [2007]. Power control should then only be considered in the case of distant nodes or energy-efficient radio units where the transmission power dominates the unit consumption.

Bit-rate control

Power control is mainly used to regulate transmission quality under packet loss and signal interference. Bit-rate control, sometimes referred to as Dynamic Modulation Scaling (DMS), is an alternative to regulate transmission reliability without increasing the transmission power, at the price of increasing the transmission latency.

DMS is a method that consists in switching the modulation characteristics (under constant power), to improve the signal reception. Figure 1.4 shows two modulation strategies: the 8-Phase Shift Keying (8-PSK) codes 8 words by changing the signal phase. Each word corresponds to a signal with evenly spaced phase and constant amplitude. The 16-Quadrature Amplitude Modulation (16-QAM), codes 16 words by changing the phase and the amplitude as shown in the figure. Switching between these two types of modulation changes the number of bits sent in a period of time (the size of the constellation). Switching is in general performed via a scheduler [Schurgers et al., 2003] that selects the best modulation scheme as a function of the packet-rate load. When the demand is low, the scheduler uses a low-bit modulation scheme that uses less energy per period, but increases the data delivering time.

In terms of control, adapting the bit-rate can be viewed at the controller side as receiving measurements with variable delay or with variable quantisation. The latter is used in [Bao et al., 2009] to derive joint rate allocation and feedback control. The bit-rate is controlled to minimise a standard Linear Quadratic (LQ) cost $J = \Sigma_{t=1}^{T}(x_t^2 + \rho u_{t-1}^2)$, where x is the system state and u the control input, over a finite time T using a finite number of bits. In the case of a noisy channel, the authors show that the optimal bit-rate is not the highest one.

Another application of Linear Quadratic Gaussian (LQG) in a NCS with quantisation and subject to packet losses is addressed in [Trivellato and Benvenuto, 2009]. It compares two communication techniques using Time Division Multiple Access (TDMA) (explained in details in the next subsection); the first one is using a fixed modulation and the second one a modulation with rate adaptation. The LQG cost to be minimised considers the TDMA slot durations of each node, and also the number of bits

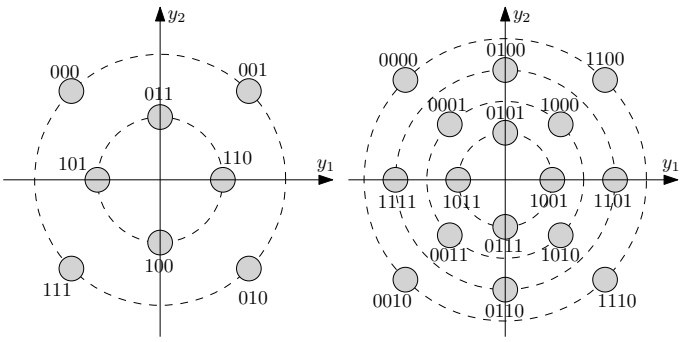


***Figure 1.4** – Modulation constellations in the complex plan for 8-QAM (left) and 16-QAM (right) modulation schemes. By switching from the 16-QAM to the 8-QAM constellation, the receiver can more easily distinguish the received symbols in presence of bad channel conditions. One direct implication of the switching is the change of data rate, and then of the latency.*

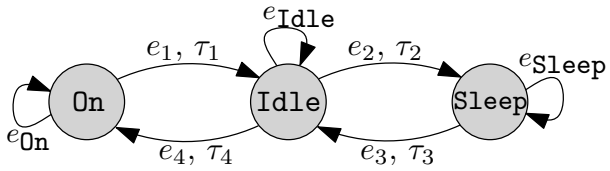


***Figure 1.5** – Illustration of a three activity modes switching automaton. Energy costs are consumed to switch between modes (e_i, $i \in \{1,2,3,4\}$) or to stay in a given mode ($e_{\texttt{ON}}$, $e_{\texttt{Idle}}$, $e_{\texttt{Sleep}}$). Time-transition costs (τ_i, $i \in \{1,2,3,4\}$) are only needed to switch between modes.*

to be used by each node in the second modulation case. On the example simulated in the paper, the fixed modulation turns out to give a performance close to the optimal control in the perfect case (*i.e.* no quantisation nor packet loss).

One major difficulty of DMS is that the receiver must be aware of a modulation change, which implies communications between transmitter and receiver before a switching.

Another difficulty, studied in [Joshi et al., 2007], concerns the bounds on the constellation size. Energy savings are only achieved if the adapted constellation size is above a minimum value which each node can decide and below a maximum value which can only be computed with the network information (topology, node density and location).

1.2.2 Data link (MAC) layer

The Data Link (MAC) layer defines how to access the medium, shared among all the nodes. It regulates the communication by providing a way to avoid simultaneous transmissions or to recover from transmission collisions. It is in charge of the physical addressing and it is able to detect and correct errors, and to send ACKs, among other features often provided by a MAC protocol. This layer is also in charge of controlling the state of the radio chip.

Radio units typically save energy by switching between different activity modes (`Idle`, `Sleep`, in addition to receiving (`Rx`), transmitting (`Tx`)) when no communication is needed. This energy management implies that some of the node components are switched off at some moments. This concerns the frequency synthesiser, the crystal oscillator, and the voltage regulator in the radio unit [Brownfield et al., 2010], but also the sensing unit, the processor, and the memories [Lin et al., 2006]. Also, switching between modes is both time and energy consuming (see Figure 1.5). A trade-off has to be made between node awareness and energy consumption. An important amount of energy is wasted by nodes in the idle-listening state (the node is in the receiving mode without data to receive).

An important drawback of having the radio in low-consuming mode is that the node is no longer connected to the network. The node is not able to route packets in multi-hop paths, or to perform a sensing operation on the controller's demand. However, if the sensing and the computational units are still active, the node is able to detect important events and to turn on the radio to send the corresponding data.

Two basic idea to save energy at this layer is to use a scheduler to manage the activity modes, or to control the packet delivery service via MAC protocol parameters tuning (frame or slot size, amount of random access, sleep and wake time, backoff characteristic).

Activity modes management

Activity modes are the states of a node (or of the radio chip in the node). While the two basic modes are `On` and `Off`, one can consider intermediate modes where only a part of the node (or of the radio chip) is active. From a control point of view, there is a difference between considering that only the radio chip can be turned to a low-consuming mode or that the whole node can be turned off. In the first case, energy is only saved at the radio level, but the node is still able to sense the system and wake up the radio to send an update to the controller in case of need, whereas when the whole node is switched off, the system cannot react to unpredictable perturbations.

The Communication Community uses scheduling policies that are used to determine when, how long, and how deep a node should sleep. As it will be presented in Section 1.2.4 mode management can profit from Event-Based Control (EBC), even though the Control Community only considers the two basic modes [Cogill, 2009].

When the mode management is considered at the Data link layer, the mode switching decision is based on the amount of packets that should be sent over the network or the probability that an event (*e.g.* a perturbation) occurs rather than the requirements of the control scheme.
At this layer, a simple approach to introduce mode management is to consider protocols based on TDMA, where the network sampling time is divided into a number of slots equal to the number of nodes and a unique slot is assigned to each node, in order to avoid collisions. This kind of protocols is periodic and it allows to schedule sleep time as shown in Figure 1.6. The sleeping interval can be enlarged by considering adaptive communication slots where the radio is switched off before the end of the slot if no packet is left to transmit. This is achieved in [Brownfield et al., 2010] where the activity modes concern only the radio-modes, *i.e.* the other components of the node such as the processor or the sensor are never turned off. The authors in [Brownfield et al., 2010] consider several low consuming modes: `Idle`, `Power Down` and `Power Off`, in addition to the `Transmit` and `Receive` modes, as detailed in Table 1.1. This permits to balance energy savings and reactivity when the sleep periods are too short to turn off the node completely. The major drawback of TDMA-based scheduling is the scalability since a slot must be assigned to each node. The latency increases with the number of nodes. A different approach is presented in [Sinha and Chandrakasan, 2001] where the activity modes take into account the processor, the memory, the sensor and its analog-digital converter, and the radio, as detailed in Table 1.2. Based on the occurrences of events and their probabilities, and on the energy and time needed to switch between the different modes, authors in [Sinha and Chandrakasan, 2001] derive a scheme to choose the activity mode offering the better trade-off between energy consumption and the probability to miss an event.

The activity modes management is also addressed in [Passos et al., 2006], even though not in a control framework, using the Hybrid Automata framework. It is addressed in the case of analog environment, using discrete programing to model the activity within the node. Energy is saved by adapting the duty cycle according to the communication reliability required by the application. Although simulation results show efficient energy savings, the model used in [Passos et al., 2006] does not consider the extra time and energy needed to switch between modes.

	Frequency Synthesiser	**Crystal oscillator**	**Voltage regulator**
Transmit	On	On	On
Receive	On	On	On
Idle	Off	On	On
Power Down	Off	Off	On
Power Off	Off	Off	Off

***Table 1.1** – Details of the radio-modes from [Brownfield et al., 2010]. Radio-modes concern only the radio chip in the node.*

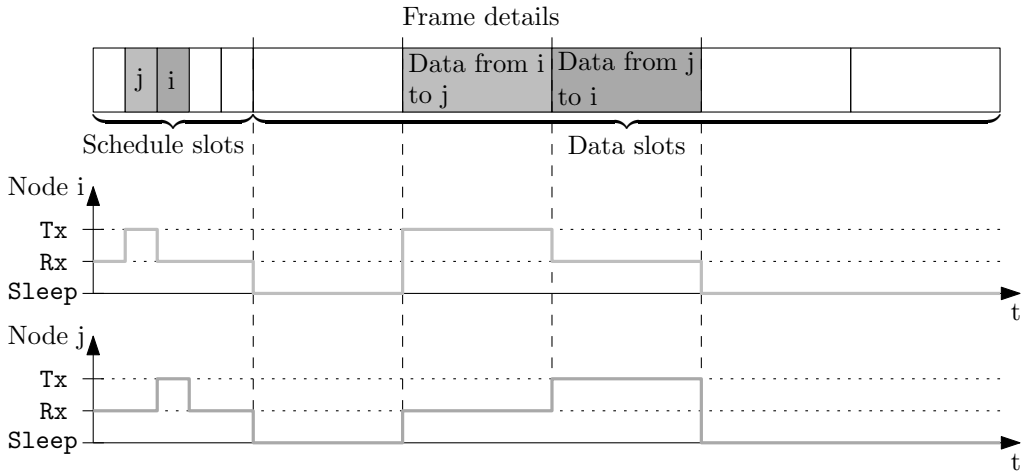


***Figure 1.6** – Illustration of TDMA-based sleeping policy. Top figure gives the details of one frame, and bottom figure shows the corresponding activities for nodes i and j. The first part of the frame contains the schedule slots. All the nodes are in receiving mode (marked as* `Rx`*) during this stage. A schedule slot is assigned for each node, and it is used to inform a destination node to be awake during the corresponding data slot. For instance, node i uses its schedule slot to say (switching to transmitting mode,* `Tx`*) that it wants to send data to j. The second part of the frame contains the data slots. In this stage, nodes which have nothing to transmit nor to receive stay in the* `Sleep` *mode.*

	Processor	**Memory**	**Sensor**	**Radio**
Mode 1	Active	Active	On	`Tx`,`Rx`
Mode 2	Idle	Sleep	On	`Rx`
Mode 3	Sleep	Sleep	On	`Rx`
Mode 4	Sleep	Sleep	On	Off
Mode 5	Sleep	Sleep	Off	Off

***Table 1.2** – Details of the activity modes from [Sinha and Chandrakasan, 2001]. Activity modes concern the whole node.* `Tx`*,* `Rx` *stand for* `Transmit` *and* `Receive`*.*

MAC protocol tuning

The MAC protocol parameters can be adapted to balance reliability and latency in order to meet the control requirements and to minimise the energy consumption. The protocols considered in this section cover both the Physical and the Data Link (MAC) layers, and sometimes also the Network (routing) layer. As concluded in [Liu and Goldsmith, 2003], there is no standard protocol dedicated to NCS[2]. The efforts done so far consisted in, either using standard protocols (such as 802.15.4, 802.11, ZigBee, Bluetooth, B-MAC, X-MAC) to exploit the tunable parameters they have to offer [Fischione et al., 2009a], or developing novel protocols dedicated to NCS, such as Breath [Park et al., 2008] or TrEND [Di Marco et al., 2010].

The authors in [Fischione et al., 2009a] have derived analytical models of the delay, the reliability and the energy consumption in unslotted 802.15.4 protocol, see Fig 1.7. From these models, the sleep time R_s and the listening time R_l of the Carrier Sense Multiple Access (CSMA)[3]-based MAC protocol can be chosen to minimise the energy consumption E under the constraints of desired probabilities for the latency $\tau_{\max}$ and for the reliability $\psi_{\min}$:

$$\min_{R_l,R_s} E(R_l,R_s,c,b,\lambda) \text{ s.t. } \begin{cases} D_{\max}(R_l,R_s,c,b,t_m) \leq \tau_{\max} \\ R_{\min}(R_l,R_s,c,b) \geq \psi_{\min} \end{cases}$$

[2] At the time this thesis is written, there is not enough information to consider the so-called RPL protocol, announced to fit the NCS requirements by the ROLL workgroup (Routing Over Low power and Lossy networks workgroup) from the Internet Engineering Task Force (IETF).

[3] A node, having data to be sent, checks if the medium is available; if the medium is free, then it starts transmission, otherwise it waits for the end of the current transmission.

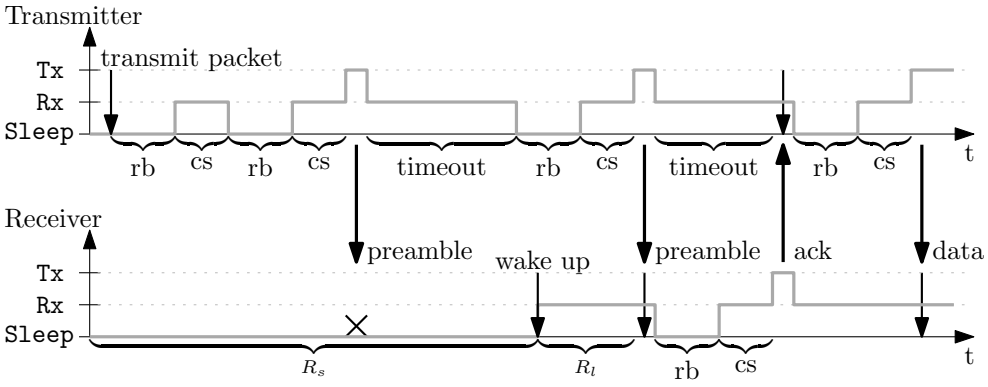


***Figure 1.7** – Illustration of CSMA-based sleeping policy with preamble messages. When the transmitter has a packet to transmit, it performs random backoff (rb, waiting period) and channel sense (cs, channel listening) until the channel is free, to send a preamble message to the receiver. If no ACK is received within the timeout period, the procedure is repeated. The receiver wakes up at the end of its sleep time R_s and listens to the channel during the listening time R_l. If a preamble is received during the listening time, it sends an ACK and the data transmission starts.*

c being the busy channel probability, b the preamble or ACK collision probability, λ the packet generation rate per node, and t_m the maximum delay of the preamble message.
This scheme is more energy-efficient than B-MAC or X-MAC, it uses CSMA, which has better scalability than TDMA, and it lets the control application adapt the latency and the reliability according to the control requirements. The models, derived for 802.15.4, have not been tested in experimentation.

On the other hand, dedicated protocols have been derived in order to be used in NCSs, such as *Breath* [Park et al., 2008] and *TrEND* [Di Marco et al., 2010] protocols. Both protocols cover the three communication layers (Physical, Data Link (MAC) and Network) and expose tunable parameters to the fourth layer (Application), and thus are suitable for control applications. They aim at minimising the network consumption under latency and reliability constraints. The network topology is based on clustering in both works but [Park et al., 2008] uses CSMA-based MAC protocol whereas [Di Marco et al., 2010] uses a hybrid TDMA/CSMA one[4]. The protocol variables (degrees of freedom) are the wake-up rate and the clustering in [Park et al., 2008], and the slot duration, the access probability, and the wake up probability in [Di Marco et al., 2010]. These variables are used by the protocols to ensure minimum energy consumption under delay and reliability constraints. These constraints are given by the application and can be adapted in a trade-off between energy saving and control performance.

1.2.3 Network layer

The Network (routing) layer is in charge of choosing a path (*i.e.*, a series of relay nodes) in the network to transmit packets between the transmitter and the receiver, in an energy-efficient manner. The usual metric at this layer is the network life-time, *i.e.* the time between the commissioning and the moment when a node first runs out of battery. Energy can be saved by using routing protocols which balance the traffic load on nodes with the higher amount of energy left, or by limiting the amount of data transmitted via network coding.

Energy-efficient routing

Literature abounds with routing protocols for WSN. A comprehensive study can be found in [Al-Karaki and Kamal, 2004] where the authors distinguish protocols based on the network structure from the ones based on protocol operation. In general terms, protocols based on network structure use the topology of the network and the amount of power available at each node to choose a route, see Fig 1.8. Although one can think of adding a weighting factor on the use of each node which could be adapted by the control application, these protocols do not offer tunable parameters and thus are not interesting in a NCS point of view.

[4]Hybrid TDMA/CSMA approach is also addressed in [Wang et al., 2006]. The hybrid approach benefits from the periodicity of TDMA, and the scalability of CSMA but is more complex to implement and application-specific.

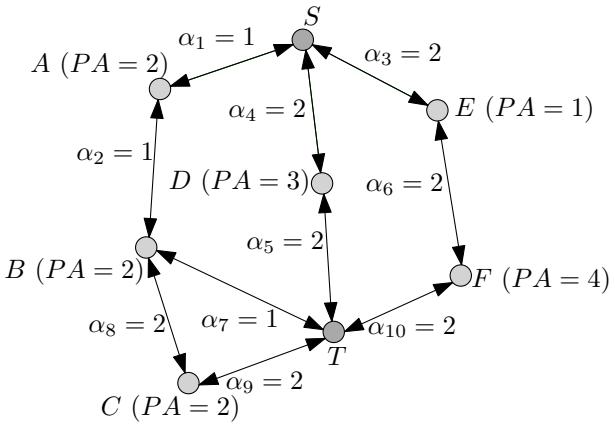


***Figure 1.8** – Network topology from [Akyildiz et al., 2002] considering the power available (PA) at each node, and the energy cost α_i to send one packet over a link i. The transmitter T should choose a route to send data to the sink S, between:*
Route 1: T-B-A-S, $PA_{tot} = 4$, $PA_{\min} = 2$, $\alpha_{tot} = 3$,
Route 2: T-C-B-A-S, $PA_{tot} = 6$, $PA_{\min} = 2$, $\alpha_{tot} = 6$,
Route 3: T-D-S, $PA_{tot} = 3$, $PA_{\min} = 3$, $\alpha_{tot} = 4$,
Route 4: T-F-E-S, $PA_{tot} = 5$, $PA_{\min} = 1$, $\alpha_{tot} = 6$.
The route is chosen according to the routing scheme. Maximum PA_{tot} route *is the* 2^{nd}. Minimum energy route *is the* 1^{st}. Minimum hop route *is the* 3^{rd}. Maximum $PA_{\min}$ route *is the* 3^{rd}.

On the other hand, protocols based on protocol operation have some parameters which allow to dynamically control the routing policy or the amount of data transmitted. Two interesting protocols are proposed in [Al-Karaki and Kamal, 2004]. The first one, called Sequential Assignment Routing, takes into account: the energy resources of the nodes, the Quality of Service (QoS) on each path, and the priority level of each packet. The second protocol, called SPEED, focuses on end-to-end packet delay. It strives to ensure minimum latency, saving energy by limiting packet overhead[5]. However, as they are defined, these protocols do not expose their tunable parameters to the application layer, meaning that the energy cannot be balanced according to the control performance in the framework of NCS. Among all the protocols surveyed by [Al-Karaki and Kamal, 2004], only *query based routing* protocols offer tunable parameters to the Application layer. In this approach, the controller node is the source of all the traffic in the network, since it requests measurements from the sensor nodes. The controller is free to ask for measurements or to run open loop. However, the delay on the measurements is doubled, and the controller has to waste energy to detect unpredictable disturbances.

An interesting approach is studied in [Pei and Chien, 2001] where a cross-layer design is considered between the Data Link (MAC) and the Network layers. This work considers the network topology where neighbouring nodes are grouped in clusters to relieve the nodes from the multi-hop routing burden. One node takes the role of cluster-head in each cluster, and is the only one in charge of multi-hop routing, letting other nodes sleep more often. The role of cluster-head is shared among all nodes to balance node discharge due to this extra load. This approach is especially interesting in networks with many nodes involving significant multi-hop communication. This is a way to keep a reasonable connectivity at the network level, while turning off an important number of nodes. Again, this approach does not offer tunable parameters to the application layer.

Other efforts have been made to derive new *ad hoc* protocols to tackle energy-efficient routing in NCS: Breath and TrEND, in [Park et al., 2008] and [Di Marco et al., 2010] respectively. They cover the Physical, Data Link (MAC) and Network layers, and they have been discussed in Subsection 1.2.2.

In spite of all these achievements, there is not yet any standard protocol that accommodates all the desired requirements from NCS.

Network coding

In the routing protocols discussed previously, the relay nodes just forward the packets they receive. Under the network coding paradigm, the nodes can perform some operations on the data from the

[5] Additional data, other than the sensed information, transmitted in the network for routing purpose.

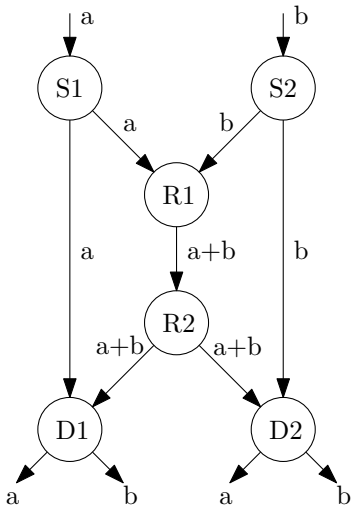


Figure 1.9 *– The butterfly network, an often used basic example to show how network coding can outperform routing schemes.*

packets they receive before forwarding them. This is generally illustrated with the butterfly network example [Yeung, 2008], as shown in Figure 1.9 and commented hereafter. Assume that two source nodes, called $S1$ and $S2$ each have an information (denoted a and b respectively) that should be both transmitted to the destination nodes $D1$ and $D2$ through the relay nodes $R1$ and $R2$. Assume also that the edges of the network can only transmit a single value per time slot. A routing scheme uses two time slots to the node $R1$ to transmit both values a and b. However, by using a simple code at the relay node $R1$ consisting in computing the sum of the two values $a + b$, a time slot is saved in order to transmit the data to the destination nodes which are able, from a and $a + b$ for destination node $D1$ (or b and $a + b$ for $D2$) to compute b (a respectively).
A complete introduction about network coding can be found in [Yeung, 2008].

The aim of network coding is to maximise the throughput of the network. It is achieved by dropping redundant information, or by implementing some computations on the data received from the neighbouring nodes. Examples of functions used for the computations are average, min, max or linear combinations. Data aggregation (or data fusion) refers to networks where some nodes implement such computations while collecting the data from neighbouring nodes, as depicted in Figure 1.10. These special nodes are called cluster-heads.

A difficulty of data aggregation is to choose the number of cluster-heads, and to select the nodes taking this role. This issue is addressed in [Chen et al., 2006] and solved with a distributed algorithm. However, authors did not consider to change the nodes playing the role of cluster-head in time, to balance the extra load implied by this task.

In [Abdelzaher et al., 2004], the degree of data aggregation refers to the amount of data aggregated (which is related to the delay needed for data to be available), and the compression

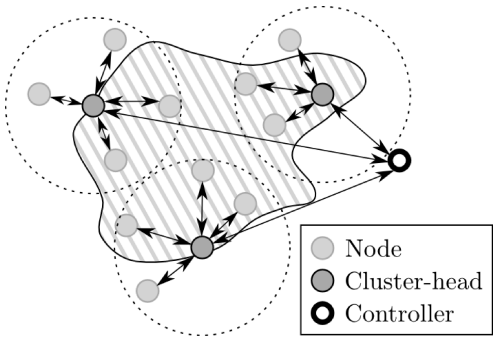


Figure 1.10 *– Illustration of data aggregation via cluster-heads in a Wireless Sensor Network to limit the traffic in the network. Cluster-heads are regular nodes with a special role. They are the only nodes able to communicate with the controller, and therefore they relay the data of the neighbouring nodes, after compression.*

scheme (lossless, all received data are sent, or lossy, *e.g.* averaging). An interesting work proposed by [Abdelzaher et al., 2004] controls the degree of data aggregation by using the traffic load as a feedback variable in order to meet some end-to-end delay constraints.

1.2.4 Application layer

The Application layer hosts the executive part of the sensing and the control, that can be divided into two categories; the sensor node that does the signal sensing and processing, and the control node that decodes the received information and computes the control action. This section is divided into a part that presents works related with energy-aware source coding at the sensor node, and another part that addresses the Application layer at the control node.

Source coding

While network coding aims at maximising the throughput, the source coding has two main objectives. The first one is to convert the sensor measurement from analog to digital (this does not apply of course when the sensor provides a digital signal). The second one is to compress the information in order to reduce the occupancy of the transmission channel. The purpose of the compression schemes is mainly to reduce the total amount of bits and, in some cases, the cost per bit associated to the transmission. The compression schemes are said either *lossless* or *lossy*. The source information is exactly recovered after the decompression process in the lossless case, whereas the source information is rounded or approximated in the lossy case. Source coding is different from the coding strategy that can be encountered in the other layers in the sense that it generally depends on the data that is transmitted.

Fundamental works Theoretical limits of source coding have been given in the famous work by Shannon, [Shannon, 1948]. In the particular case of NCS, we are not only interested in the minimum number of bits that should be considered to describe the source, but also to stabilise the closed-loop system, or to attain other control performance indexes. An overview of feedback control under data rate constraints is given in [Nair et al., 2007].
Moreover the Information Theory literature is rich of various types of coding strategies like: Huffman Coding, Arithmetic Coding, Lempel-Ziv codes, Run-Length Coding, Tunstall Coding. A comprehensive summary of some of these codings can be found in [Gersho and Gray, 1992, Barr and Asanović, 2006]. Discussing these results is out of the scope of this thesis and the author refers the reader to the references given in this paragraph for details.

Distributed Source Coding One can notice that the coding strategies mentioned in the previous paragraph mainly address topologies considering a single source. In the case where several sources provide information (which is naturally the case in sensor network), other approaches rise. For instance the Slepian-Wolf theorem [Slepian and Wolf, 1973] allows to reduce the amount of data travelling in the network (and thus to save energy). This is the purpose of Distributed Source Coding (DSC) which exploits the fact that measurements performed by neighbouring nodes are correlated. Part of the sensor nodes sends uncompressed data (called side information) and the other part compresses it using Slepian-Wolf theorem. The compressed data can only be decompressed by the receiver if it owns the side information. A review of DSC is given in [Xiong et al., 2004]. DSC with noise, error, and packet losses, studied in [Tang et al., 2007] shows up to $31,6\%$ of energy savings without reducing the signal to distortion ratio. To the author's knowledge, DSC has been applied to sensor network but not to close the loop over the network. The lossy version of DSC is called Wyner-Ziv coding, see [Li and Ramamoorthy, 2011] for a review about Distributed Source Coding.

It is worth mentioning that distributed estimation can be seen as a Distributed Source Coding in the case of NCS. As it is shown in [Fischione et al., 2009b], distributed estimation can outperform centralised estimation (in terms of estimation error variance) when the network is subject to packet losses.

Uniform vs. non-uniform energy coding The coding strategy depends on the distribution of the transmission cost. Uniform energy coding assumes that the transmission cost is uniformly distributed among all the codewords, or equivalently that bits "0" and "1" are associated to an equal energy transmission cost. In this context, the coding strategy consists in trying to minimise the total number

of bits that the codewords will use without any other consideration about the nature of the bits value themselves. On the other hand, non-uniform energy coding assumes that the transmission cost is not uniformly distributed among all the codewords, *i.e.* some codewords (or bits) are more costly than others. These differences can be captured by assuming some particular probability distribution, or by changing the distribution via the well known process of transform coding, see [Goyal, 2001].

Uniform energy coding Differential coding is an example of uniform energy coding. It encodes differences (or variations) of a signal rather than its absolute value. Delta (or Delta/Sigma) modulation is one of the simplest form of differential coding. It uses a 2-level (1-bit) quantiser and a dynamic predictor implemented on both transmitter and receiver sides. The node sends "0" if the difference between the measure and the prediction is positive, and it sends "1" if it is negative. This reduces the amount of data transmitted to only one bit per sample. This scheme exhibits a trade-off between stability and precision tuned via the predictor's gain. The closed-loop stability of this scheme in the context of NCS has been studied in [Canudas De Wit et al., 2009] and extension to MIMO systems can be found in [Jaglin et al., 2008]. An improvement, proposed in [Gomez-Estern et al., 2007], uses an adaptive mechanism to change the value of the gain on the run.

Non-uniform energy coding Minimum Energy (ME) coding is an example of non-uniform energy coding. It combines the On/Off Keying (OOK) modulation with a particular energy-efficient coding method. OOK modulation emits a fixed signal with a given power to transmit a high bit (*i.e.* "1") and no signal to transmit a low bit (*i.e.* "0"). The goal of ME coding is to produce codewords formed by the minimum number of high-cost bits (or equivalently with the maximum numbers of "0"). The first work on ME coding [Erin and Asada, 1999] assumes that the source symbol probabilities are known. For a given dimension of alphabet, codewords with fewest high bits are assigned to the symbols with the highest probabilities and inversely for codewords with fewest low bits. This approach has been extended by considering unknown source probabilities, and by including error detection and correction mechanisms [Prakash and Gupta, 2003]. Extension to improve energy savings on the receiver side [Kim and Andrews, 2005] have also been considered. However, this technique is only used in sensor network without feedback control.

Even if the transmission costs of the bits or the codewords at the physical layer are uniform, it is possible to implement a source coding strategy to reduce the *average amount* of bits to be sent. Indeed, an event-based detector can be associated with a variable length encoding process to save energy when transmitting a certain event (*i.e.* stand-still event) which has a higher probability to occur. The event-based detector has the ability to quantify and to differentiate stand-still signal events from changes in the source (level crossing detector), see Figure 1.11. Coding is effectuated by defining a 3-valued alphabet for the minimum bit case, and $(2L+1)$-valued alphabet for a general case with a precision inversely proportional to $L \in \mathbb{Z}^+$. Hence, the stand-still signal event is modulated with a low-power signal, whereas the changes of levels are modulated with high-power ones. Entropy coding is added to improve the energy savings by assigning a probability distribution to the events. In that way, the mean transmission energy can be substantially improved for systems where the stand-still events may have high probability to occur (*i.e.* stable systems). An example of an entropy coding mechanism is the run-length encoding as shown in Table 1.3. This results in transmission mechanism that is inherently asynchronous as the transmission period depends on the signal events, as it can be see in Table 1.3. In the context of NCS, contributions along that direction can be found in [Canudas De Wit et al., 2007a], [Canudas De Wit et al., 2007b] and [Canudas De Wit and Jaglin, 2009].

Authors in [Canudas De Wit and Jaglin, 2009] propose a closed-loop control implementing 3-levels and entropy codings, as shown in Figure 1.12. In that work, the authors have reported details on the closed-loop properties of such arrangement, and quantified the possible improvements in term of energy savings that the scheme may provide.

Cross-layer approach It is shown in [Varagnolo et al., 2008] that the performance of an estimator (in terms of the estimation error covariance) using measurements from a WSN is related to the ratio between process and measurement noises, the dynamics of the system estimated, and the size of the buffer of the estimator. This exhibits the need for cross layer design between the Network and the Application layers to increase performance and to allow trade-off between performance and energy consumption. Network Aware Source Coding (NASC), introduced in [Li, 2006], aims at reducing the total traffic load or the total required energy rather than minimising each transmission cost. As an initialisation stage, the codewords are assigned to the senders according to their symbols probabilities,

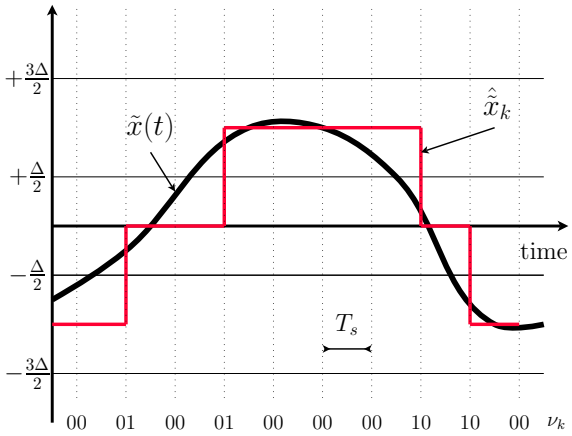


Figure 1.11 *– Illustration of the 3-levels coding scheme, from [Canudas De Wit and Jaglin, 2009]. The signal detection levels are spaced by the quantum Δ. Uniform samples are taken every time T_s triggered by a local synchronous clock. The level detector produces a signal ν_k (identified by "01" or "10") whenever a level crossing takes place, and a "00" if the signal remains within the level. The parameter Δ defines a trade-off between stability and precision.*

and their distances and number of hops to the receivers. Shortest codewords are assigned to symbols expecting to travel the longest path. Although this approach considers several senders and receivers, one limitation is that it assumes the ith sender is always sending data to the ith receiver. The codewords assignment is then not relevant if the ith sender communicates with the jth receiver. Moreover, the solution derived in [Li, 2006] has not been tested in simulation.

Video encoding In the special case where the sensors are camera nodes, source coding refers to video encoding, whose goal is the similar to generic source coding: compressing the data captured by the sensor to reduce the bit-rate. However, video coding differs from source coding because there exists a strong temporal correlation between two images taken from a single camera, and the amount of data to be processed is huge. Video coding is the rare case where compressing the data may consume more energy than sending the uncompressed one [Chiasserini and Magli, 2004]. There exists compression schemes devoted to the video case (*e.g.* MPEG, H.264), whose discussion is out of the scope of this thesis.

Networked control schemes

The distributed nature of sensor and control nodes in the context of NCS makes each of these components operate with their own clock and react asynchronously to events. This forces the nodes on the network either to perform a synchronisation, or to operate under asynchronous updating. The clock synchronisation problem can be tackled, for instance, with consensus [Carli et al., 2011]. Having synchronised nodes permits to use the standard synchronous control designs, even though these methods have to be adapted to the network limitations (such as delay, limited bandwidth, packet loss). There

Input (Run sequence)	Transmission period	Output
01	T_s	000
10	T_s	001
00 01	$2T_s$	010
00 10	$2T_s$	011
00 00 01	$3T_s$	100
00 00 10	$3T_s$	101
00 00 00	$3T_s$	110
unused	–	111

Table 1.3 *– Run-Length Encoding, from [Canudas De Wit and Jaglin, 2009]. T_s is the sampling time. The coding scheme is based on the consecutive number of zeros.*

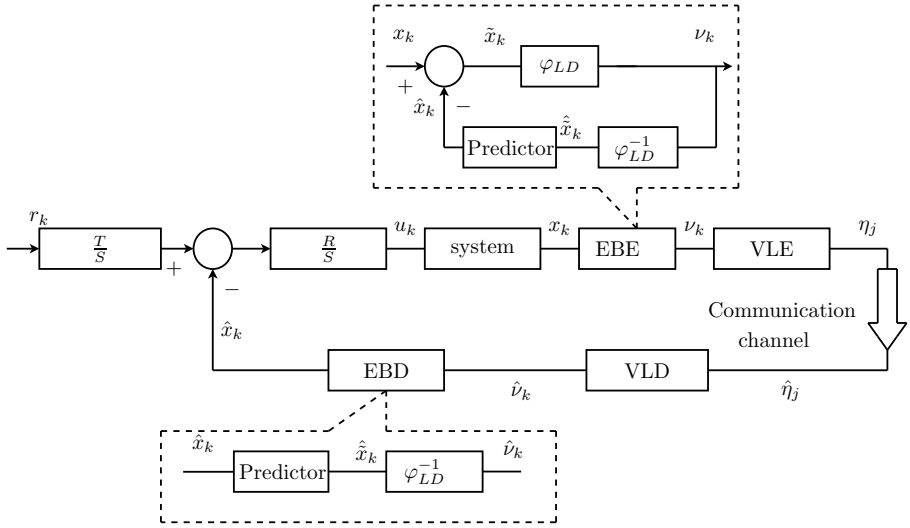


Figure 1.12 *– Block diagram of the implementation of 3-levels and entropy codings in a feedback loop, from [Canudas De Wit and Jaglin, 2009]. The measurements of the systems go through an Event-Based Encoder and an Variable-Length Encoder before being sent via the Communication channel. Two decoders are needed at the receiver side to retrieve the data and to compute the control inputs.*

are two main directions to avoid synchronisation. The first one consists in handling the asynchronous communication in a synchronous setup, using *e.g.* time-varying delay or Zero Order Holder (ZOH), as explained hereafter. In the second direction, the time-triggering paradigm is relaxed to Event-Based Control (EBC). The nodes in the network operate under asynchronous sampling, *i.e.* the samples are received aperiodically as they are triggered by the occurrences of events. This technique is more evolved but it is also more energy-efficient, and with bad channel conditions, it is likely to perform better than a synchronous sampling which temporary loses the synchronisation or often loses packets.

Varying-delay approach The measurements from the sensor to the controller may be delayed or dropped when sent over a communication channel. A natural approach in NCS is to consider that the measurements are received with a time-varying delay [Chaillet and Bicchi, 2008, Bemporad, 1998]. This approach is a technique that received great attention to address the problem of asynchronous communication in a network. It is used to model the asynchronous arrivals, at time t_k, of measurements at the control node [Fridman et al., 2004]. The time-varying delay is then modeled as:

$$\tau(t) = t - t_k, \quad 0 \leq \tau(t) \leq h$$

where h is the maximum time interval between two measures. The asynchronous arrival of the sensed information can thus be modeled as: $x(t_k) = x(t - \tau(t))$, and then a static feedback K can be designed to cope with this delayed state and to stabilise the resulting closed-loop system [Fridman et al., 2004]:

$$\dot{x}(t) = Ax(t) + BKx(t - \tau(t))$$

A limitation of this approach is due to long stand-still signals (in particular when system is at its equilibrium) in event-based sensing, as they will introduce too long h leading to a very conservative control design.

To zero, to hold, or to compensate Another natural approach to cope with asynchronous measurement arrivals at the controller side is to implement a periodic control scheme that uses measurements when they are available and implement another behaviour when no new measurement is received. Among the possible open-loop strategies, the most natural ones are: to do zero (*i.e.* to apply a zero control input to the plant), to hold the last control input, or to compensate (*e.g.* to predict the state of the plant). Even though one can think that to hold the control input gives better closed-loop performance than to zero, author in [Schenato, 2009] shows that it is not always true.

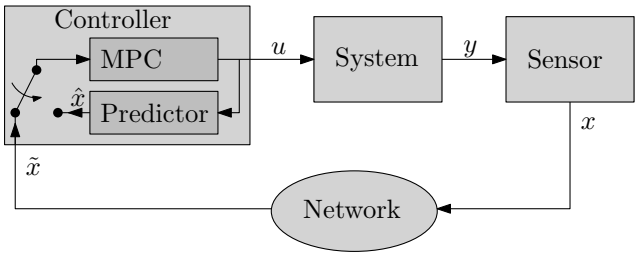


Figure 1.13 *– Illustration of split sensing and control approach. The sensor transmits measurements asynchronously to save communication energy. The MPC controller uses local predictions when no measurement is received.*

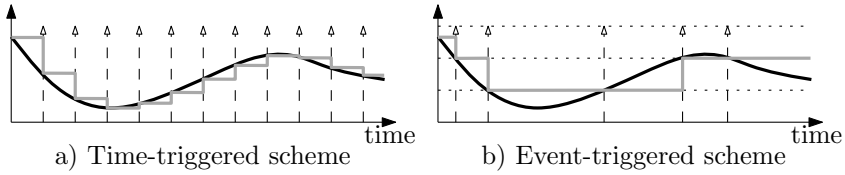


Figure 1.14 *– Comparison of time-triggered and event-triggered discretisation; blank arrows indicate time instant where a sample is taken.*

Note that an alternative implementation of the holding strategy consists in holding the last received measurement rather than the last updated control input. In many cases, both approaches end up to the same action (*e.g.* in the case of linear state feedback control).

Examples of compensation schemes are given in [Bernardini and Bemporad, 2008], [Gommans et al., 2012] where a predictor runs on the controller side to predict the state of the system between two consecutive measurement arrivals. To this aim, a Model Predictive Control (MPC) controller is derived in [Bernardini and Bemporad, 2008]. It generates periodic control inputs using predictions when no measurement is received, as shown in Figure 1.13. On the sensor side, the same predictor is implemented, and new measurements are only transmitted when the difference between the prediction and the measurement exceeds a given threshold, in order to avoid communications that are not useful. Because the sensor limits the measurement sending to the occurrence of a given event (threshold crossing), this last scheme can be considered as an Event-Based Control (EBC) scheme. A different approach is used in [Gommans et al., 2012] where the erasure channel can be on both link from the sensor to the controller and from the controller to the plant. The controller runs an observer that predicts the plant state when no measurement is received. Authors in [Gommans et al., 2012] propose a synthesis of the model-based compensator and show that it outperforms both the zero and the hold strategies.

Asynchronous paradigm - Event-Based Control Event-Based Control (EBC) (also called Event-Triggered Control) is a new paradigm in Control Theory that brings many benefits and challenges to NCSs. It tackles asynchronous measurement arrivals and reduces the network bandwidth occupancy. While historically control analysis and design were developed in continuous time, their practical use in a digital world made the community extend the results to discrete-time control. But discrete-time control relies on perfectly synchronous sensors and controllers (with the possibility to consider some delays, as explained above), which is a constraint difficult to meet in networked control. As an alternative point of view, EBC proposes to discretise the continuous signals, not in the time domain, but in the value domain, as depicted in Figure 1.14.

The early researches on asynchronous sampling rate were known as “adaptive sampling”, see *e.g.* [Gupta, 1963, Liff and Wolf, 1966, Bekey and Tomovic, 1966, Ciscato and Martiani, 1967]. The recent framework of NCS makes the use of asynchronous sampling a hot topic [Yook et al., 2002, Tabuada, 2007, Heemels et al., 2008, Henningsson et al., 2008]. Indeed, this approach is well suited to setups where sensing and controlling tasks are done in separated nodes. The sensor nodes only measure and process the sensed data to be transmitted to the control node, in charge of updating the control law. Energy-efficient asynchronous control relies then on reducing the communication load needed to update the control law. Energy-efficiency sensing schemes must be implemented in the sensor nodes to decide when to send new measurements to ensure closed-loop stability and given control performance.

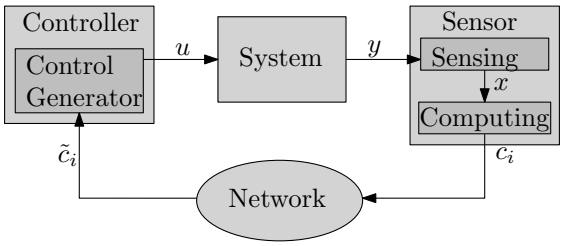


Figure 1.15 *– Illustration of the cooperative sensing and control approach. The sensor computes parameters c_i from the measurements and sends them asynchronously to the controller to generate the control time-profiles.*

These schemes are often event-based and application specific. The controller must be designed to tackle aperiodic sampling arrivals, and acknowledgements may be needed in the case of lossy channel.

An illustration of EBC can be found in [Cogill, 2009]. In this paper, new measurements are used by the state feedback controller of gain K when they are received. The closed-loop system, subject to noise w_t, is given by:

$$x_{t+1} = Ax_t + BK\hat{x}_t + w_t$$
$$\hat{x}_{t+1} = (1 - a_t)\hat{x}_t + a_t x_t$$

$a_t \in \{0,1\}$ describes the sampling instant: the state x_t is actually transmitted to the controller if $a_t = 1$, otherwise the controller keeps using the estimate $\hat{x}_t$. A policy to transmit the measurement to the controller is derived from a classical quadratic cost function involving x_t and a_t which captured the cost associated to the transmitting mode, assuming that the node is sleeping when not transmitting. This formulation results in a natural trade-off between the control performance and the energy spent for the communication. The policy to select a_t is:

$$a_t = \begin{cases} 0, & \text{if } f(x_t) \leq \lambda, \qquad \lambda > 0 \\ 1, & \text{otherwise} \end{cases} \tag{1.1}$$

where λ is a weighting constant to balance the penalties on x_t and a_t, and f is an analytic function capturing the differences of dynamics of the system when a sampling occurs or not. The mechanism to determine when the sensor should send a measurement is thus based on a trade-off between the control performance and the communication consumption. This trade-off is tuned by the parameter λ in equation (1.1).

EBC can be used in a cooperative sensing and control approach to save more energy. The sensor node processes the measurements but also computes parameters for the control law. Whereas in the previous approach the sensor node transmits the state of the system, in this one, it pre-computes some parameters that will be used to generate the control law, see Figure 1.15. The time-profile of the control evolution is here pre-constrained to some specific profiles such as series of impulses, piece-wise constant, or linear growth. The sensor node sends parameters describing the time-profiles. Energy is saved by the asynchronous nature of the communication, but also by reducing the total amount of data transmitted.

The cooperative sensing and control approach has been introduced in [Åström and Bernhardsson, 2002], even though the problem is not explicitly addressed in the framework of NCS, using impulses-shaped time-profile for the control. The parameters provided by the sensor node are the time instant to apply the impulses, and their magnitudes. The asynchronous sampling shows a better output variance than the periodic one.

The case of PID-control applied to EBC has also been considered [Årzén, 1999, Rabi and Johansson, 2008]. A PI-control structure is introduced in [Rabi and Johansson, 2008] where two level-crossing detectors are implemented on the sensor node. The first, called the P-part, is triggered by the regulation error while the second, the I-part, is triggered by the integral of the error. The sensor node sends time intervals for updating the control waveform which is on the form:

$$u_t = \begin{cases} -K_P\xi_{\tau_\lambda} - K_I\rho_{\tau_\lambda}, & t \in [\tau_\lambda, \tau_\lambda + \epsilon(K_P, K_I)] \\ -K_I\rho_{\tau_\lambda}, & t \in [\tau_\lambda + \epsilon(K_P, K_I), \tau_{\lambda+1}] \end{cases}$$

where τ_λ is the last sampling instant, ξ_t is the error and ρ_t its integral, K_P and K_I are the proportional and integral gains, and ϵ is the duration of the impulse triggered by the P-part. Trade-off between control performance and communication consumption can be balanced through the thresholds used at the sensor side. However the authors do not propose specific rules to tune the gains, and they show that sustained oscillations can result from bad tuning of the gains and the thresholds.

It has to be noticed that [Åström and Bernhardsson, 2002] is actually one of the ground work of EBC, with the contribution [Otanez et al., 2002] (where EBC is named *deadband control*), but the cooperative aspect is absent from these works. The approach considered in [Rabi and Johansson, 2008] is one of the few contributions in that direction, and open issues, such as multiple sensor nodes or real channel conditions, are to be investigated.

While Event-Based Control (EBC) is an approach that limits the amount of communication, in its practical implementation, it needs the sensor and the computing unit to be always active in order to monitor the state of the system, *i.e.* the occurrence of the triggering events. In order to save more energy at the sensor side, self-triggered control considers to turn off completely the sensor node between the control updates [Anta and Tabuada, 2010, Velasco et al., 2003]. In this approach, when the node is active, it senses the system, it sends the measurement to the controller, it computes the next time instant when a new measurement will be sent to the controller, and then it completely turns off until the next waking-up instant that has just been computed. While this behaviour is very energy-efficient, it makes the system running open loop between the waking-up instants. There is thus a natural trade-off between energy consumption and reactivity to unpredictable disturbances. Also, proving the stability of the closed-loop system is not trivial.

Sensor scheduling Sensor scheduling is mainly addressed to perform estimation in systems with communication limitation, for instance bandwidth limitation in a wireless case. Because of this constraint, only one sensor (or one group of sensors) is allowed to send a measurement to the estimator at each sampling time. Sensor scheduling can be seen as a similar problem to Event-Based Control (EBC) in the sense that the scheduler has to choose the communicating sensor in an event-based fashion.

A first approach to tackle sensor scheduling is to search for the best scheduling sequence among all the possible sequences. Two major techniques are considered in that direction. The first one consists in building a combinatory tree of all possible schedulings, and pruning it in order to limit the computational burden to find the optimal scheduling while ensuring optimality with respect to a given criterion [Huber, 2012]. Another (and possibly complementary) way to further lighten the computation complexity consists in relaxing the optimality and deriving sub-optimal schemes [Alriksson and Rantzer, 2005]. In the second approach, the scheduling is chosen in a stochastic manner, where the probability distribution is determined in order to obtain the best expected steady-state performance [Gupta et al., 2006].

Other works use the framework of network scheduling, considering protocol like Round-Robin, see *e.g.* [Nešić and Teel, 2004, Heemels et al., 2010, Donkers et al., 2011], and some contributions consider the co-design problem of scheduling and control feedback, see *e.g.* [Dačić and Nešić, 2007, Antunes et al., 2012a]

Sensor scheduling is mainly motivated by the use of a communication channel that only allows one transmission per sampling time, *i.e.* transmission techniques which take time and forbid interferences, such as echo-based sensors as sonars. Even though the use of such transmission technology may be limited, the problem of sensor scheduling can be applied to NCS to limit the amount of communication, thus reducing the energy consumption and the bandwidth occupancy.

1.2.5 The influence of the hardware

Cross-layer design can also be considered in order to take into account the battery, or some extra hardware, or the network topology.

Energy can be saved by taking into account the hardware characteristics, and especially the battery which can recover energy when it is not used. Based on this observation, authors in [Sausen et al., 2008] introduce a Dynamic Power Management (DPM) which turns the radio off during a little amount of time (1 ms in the experiment results) after each transmission. This permits to have 35.7% of additional transmissions before running out of battery.

Another technique taking into account the battery is derived in [Lahiri et al., 2002]. The authors propose a battery-aware power management which adapts the components utilisation according to

the specific battery characteristics. Indeed, most of the energy-efficient schemes are interested in the amount of energy that is consumed by the sensor's components regardless of the ability of the battery to provide the corresponding voltage and current. In this contribution, the consumption at the component level is considered on the light of the current variation. It is interesting indeed to consider the current variation since the battery-life can be extended by working around a nominal point, avoiding current picks. In the results, the authors exhibit a trade-off between the battery life-time and the achievable data rate, using their power management. Their solution is compared to classical power management techniques and shown to be better.

As it has been said, the major part of the energy in a node is consumed for communication, and a great part is wasted by idle-listening. To eliminate this waste, an extra hardware is presented in [Ruzzelli et al., 2007]. It is composed of a RFID (Radio-Frequency IDentification) tag and reader, called RFIDImpulse. The passive tag is used to receive wake up demands, and the active RFID reader is used to send wake up demands to a particular tag or a set of tags. This allows to turn off the radio in the nodes systematically when not used to maximise time spent in sleep mode, as they can be waken by the RFID tags. However a cross-layer design is needed to take into account the RFID hardware, the activity modes and the MAC and routing protocols. Indeed the stage of waking up a node must be handled in the MAC and routing protocols before initiating a communication.
A comparative study of RFIDImpulse, BMAC and IEEE 802.15.4 is held in [Jurdak et al., 2008], showing that RFIDImpulse consumes less energy than the others for low and medium data traffic load whereas 802.15.4 performs better for high load. More generally, they observe that the depth of the sleeping mode should be related to the traffic rate.
The same mechanism can be obtained with Infra-Red light emitting devices (*e.g.* photodiodes) instead of the RFID devices.

1.3 Room for improvement

1.3.1 Need for multi-layer approaches

We have reviewed the four layers of the NCS stack: Physical, Data Link (MAC), Network and Application layers, emphasising on energy consumption. Energy-efficient techniques exist at each layer but they are rarely suitable for the NCS framework.

First, it has to be noticed that an energy-efficient coding scheme implemented on a single layer is likely to penalise the other layers from efficiently saving energy. In other words, saving energy at the Data Link layer with an appropriate MAC energy-aware protocol limits the degrees of freedom of the routing protocol at the Network layer to also save energy. For this reason, energy-efficient communications consider multi-layer protocols.

However, another important point is that there appears to be a gap between the application layer, mostly treated by the Control Community, and the ones below, addressed by the Communication community. On the one hand, control applications mainly use only one sensor node and one control node, avoiding medium sharing and routing issues, and simple channel and radio models. Even though they consider some imperfections of the communication channel (such as packet loss, delay, or limited data rate), imperfections are often treated once at a time and low-level imperfections (such as interferences or retransmission schemes) are generally not considered. On the other hand, MAC and routing protocols often control their energy consumption themselves, leaving few or no degrees of freedom to the Application layer. This is acceptable in a monitoring configuration, and more generally in WSNs, but as soon as the loop is closed in a NCS, the energy saved at the communication level may deteriorate the control performance.

Frameworks that encompass several network-induced imperfections for control applications have been studied only recently [Heemels et al., 2010, Donkers et al., 2011, 2012]. These contributions deals with sensor scheduling, varying delays, varying transmission intervals (that can include dropouts). Although these works (and the references therein) are very interesting, they were not available at the beginning of this thesis, and thus they are not considered any further in this document.

A possible step toward the unification of control and communication lies in a multi-layer protocol dedicated to NCS, allowing the control application to change the latency and the reliability to meet the control requirements under energy constraints. However, this is a challenging problem since, as it can be seen with homemade protocols in [Park et al., 2008, Fischione et al., 2009a, Di Marco et al.,

2010], the solutions existing so far are often application-oriented and may not be applied in a general context.

Even though energy-efficiency in wireless NCSs has received a lot of interest in the past decade, it is still an open problem. It is stated in [Fischione et al., 2006] that the communication parameters need to be derived considering the application in order to improve the control performance. Authors in [Goldsmith and Wicker, 2002] show that cross-layer design is imperative to satisfy application requirements with limited energy resources. The main challenge is to successfully merge the advances that have been made in both Control and Communication literature.

1.3.2 Focus of the thesis

From the elements given in the previous sections, an important observation is that a significant amount of energy is wasted in the radio unit. Indeed, the time-triggered paradigm forces communications even when it is not necessary. It has been seen that Event-Based Control (EBC) allows to limit the communications to time instants when important events occur with respect to the closed-loop behaviour, thus providing good control performance with drastic energy savings. While the Communication Community exploits intermediate radio modes (which exists even in cheap and simple radio chips), the Control community limits the radio to the two basic modes, *i.e.* `On` and `Off`.

We thus decide to consider an event-based approach with the explicit purpose of saving energy. Indeed, while EBC may be used to limit communications, our goal is to exhibit a trade-off between control performance and energy savings. To further increase the energy efficiency, in this thesis, we take into account a more detailed radio chip model that considers intermediate radio modes, such as Idle or Fast Tx[6]. To the author's knowledge, joint radio-mode management and EBC has never been addressed before.

In the rest of this work, we focus on a single loop problem, namely a single sensor transmitting its measurements to a single controller over a wireless channel. Indeed, by keeping the problem in its simplest form (but yet challenging), we can have a general understanding about the co-influence of the control application and the radio unit, in terms of behaviour, parameter tuning, or degrees of freedom. Also, it has been seen in Section 1.2.3, that the Network layer is not offering conclusive parameters to be adapted at the application Layer. Then, considering more than two nodes would have brought fundamental issues at the Network level preventing to focus on the problem at the radio level.

[6] Fast Tx is a radio-mode allowing rapid switching to the transmitting mode while consuming more energy than the other low-consuming modes.

Chapter 2

Problem formulation and control approach

Contents

2.1 Mathematical model for the closed-loop system

We consider the problem of energy efficiency in wireless Networked Control Systems (NCSs). We restrict our focus to a setup composed of two nodes, as depicted in Figure 2.1 and described hereafter. A two nodes setup captures the challenges of energy efficiency without introducing the difficulties appearing in a multi-nodes setup (such as Media Access Control (MAC) or routing).
The first node, called the smart sensor node, has sensing and computing capabilities, as it has been introduced in Figure III. It is in charge of sensing the system output, computing the feedback law and deciding whether or not to send the control input to the second node, in charge of applying the control input to the actuator. It is assumed that the receiver node is co-located with the actuator, and thus it has access to an unlimited energy source. This configuration, also called *one-channel feedback NCS*, is commonly considered in the literature, as it is described in [Hespanha et al., 2007]. Thus, we are only interested in saving energy at the smart sensor side.

To this end, we combine two techniques. Firstly, the communications from the smart sensor to the receiver node are event-based. This means that the smart sensor can decide to let the system run open loop when the control performance is good enough. We define in Section 2.4 a cost function that is used by the smart sensor to determine how good is the control performance with respect to the energy needed for a transmission. When the receiver node does not receive any new update from the smart

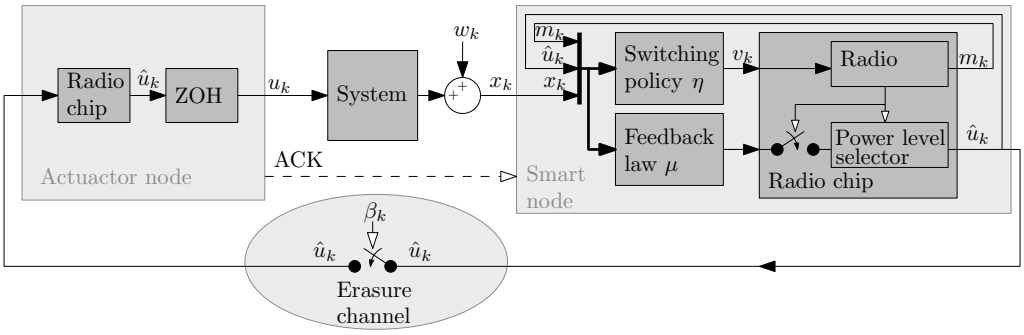


***Figure 2.1** – Block diagram of the problem setup. The smart node measures the noisy state x_k from the system, it computes the feedback law $\hat{u}_k$ and decides whether to send it or not to the actuator node. $\hat{u}_k$ can actually be $\varnothing$ when no transmission is scheduled or when the transmitted message is dropped. Then, the receiver is able to determine if it has received an update or not. β_k equals 1 when the transmission is successful or 0 when there is a dropout.*

sensor, it holds the memory of last received control input. The memory is implemented in a Zero Order Holder (ZOH) device in Figure 2.1. The control update computed at the smart sensor side, denoted $\hat{u}_k$, can be different from the control input actually applied to the system, denoted u_k. Both $\hat{u}_k$ and u_k take values in $\mathbb{R}^{n_u}$.
This event-based behaviour is considered on the base of a discrete-time monitoring. This means that the smart sensor only monitors the system at a given sampling time. Then, depending on the state of the system, it decides whether or not a transmission occurs at the current sampling time. This last point is important, as highlighted by the authors in [Heemels et al., 2013], since it relaxes a widely used assumption in Event-Based Control (EBC) that the events are monitored in continuous time, while only discrete time monitoring can be implemented in practice.

Then, the second technique considered jointly with the Event-Based approach is the radio-mode management. Indeed, not only the smart node decides whether to use or not the radio, but also it decides upon the radio-modes. Indeed, when the radio is active, a transmission power level is chosen. When it is not active, it is switched to one of the several low-consuming radio-modes. The radio can use different power levels to change the transmission success probability in the case of unchanged channel conditions. In case of varying channel, the transmission power can be increased to face bad channel conditions and keep the transmission success probability constant. We define N_1 as the number of transmission levels. The low-consuming modes (*e.g.* `Idle`, `Sleep`), explained in details in Section 2.1.2, allow to save energy by turning off some components in the radio chip. However, a certain time and energy is needed to switch between active and inactive modes and vise versa. This implies that switching to a low-consuming mode, in addition to making the system run open loop, can cause more energy waste than staying in a transmitting mode. We define N_2 as the number of low-consuming modes. The total number of radio-modes is given by $N = N_1 + N_2$.

Hereafter, *feedback law* refers to the feedback control law that is used to compute the control input $\hat{u}_k$ possibly sent to the receiver node and applied to the system. *Switching policy* refers to the node's decision to switch to a given mode. The switching decision is denoted $v_k \in \{1, 2, \cdots, N\}$, where $v_k = i$ means that the radio-mode is switched to mode i at time k.

2.1.1 System model

The system to be controlled is a linear discrete-time system with an additive zero-mean white Gaussian noise, described by Equation (2.1):

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{2.1}$$

where x_k is the system state (fully observed) and u_k is the control input taking values in $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_u}$ respectively, and $w_k \sim \mathcal{N}(\mathbf{0}, W)$ is the measurement noise. A and B have appropriate dimensions and the system is controllable, and may be unstable.

2.1.2 Radio chip model

The state of the radio chip is the mode at time k, m_k:

$$m_k \in \mathbb{M} \triangleq \mathbb{M}_1 \cup \mathbb{M}_2$$

with $\mathbb{M}_1 \triangleq \{1, 2, \cdots, N_1\}$ being the set of transmitting modes and $\mathbb{M}_2 \triangleq \{N_1+1, N_1+2, \cdots, N\}$ the set of low-consuming modes.

The radio-mode is updated according to the switching decision: $m_{k+1} = \nu_k$. This means that choosing for $\nu_k \in \mathbb{M}_1$ includes that the decision to transmit is taken.

The consumption of the radio chip during any sampling interval, called the transition cost and denoted $\theta_{m_k,m_{k+1}}$, depends on the radio-mode m_k and on the switching decision ν_k. As it is explained in details in Section 2.1.5, the modes and the transition costs we consider here, at the control application level, are different from the ones at the radio chip level. For instance, we consider at the control application level that a transition cost from a low consuming mode to a transmitting mode includes the cost to change mode at the radio and the cost for the transmission.

The amount of energy E consumed since the commissioning (where $E_0 = 0$) can be computed as follows:

$$E_{k+1} = E_k + \theta_{m_k,\nu_k} = E_k + \theta_{m_k,m_{k+1}}.$$

2.1.3 Channel model

As it is done in [Imer et al., 2006], the channel is modeled as a simplified memoryless erasure channel where the message $\hat{u}_k$ is dropped with probability $\epsilon(m_k)$ for $m_k \in \mathbb{M}_1$, and otherwise is correctly received. We assume that the channel gain process is i.i.d (independent and identically distributed).

The dropout probabilities depend on the transmission power used by the radio chip, *i.e.* the transmitting mode. Higher transmission power implies higher success probability, *i.e.* $\epsilon(1) < \epsilon(2) < \ldots < \epsilon(N_1)$. We consider a model where the dropout concerns the real-valued message $\hat{u}_k$, not single bits or packets. The transmission successes are modeled by Bernoulli random variables β_k, where 1 denotes success and

$$\mathbb{P}\{\beta_k = 0 | m_k = m\} = \epsilon(m),$$
$$\mathbb{P}\{\beta_k = 1 | m_k = m\} = 1 - \epsilon(m).$$

Given the mode m_k, β_k is conditionally independent of the past $\{\beta_h\}_{h<k}$, $\{w_k\}_{h<k}$; the mapping $\epsilon(m)$, $m \in \mathbb{M}_1$, is known for design purposes[1]. Acknowledgments (ACKs) are sent by the receiver node to confirm to the smart sensor that a new control update has been applied to the system. These ACKs are assumed to be reliable, *i.e.* always correctly received by the smart node. This is a reasonable assumption since the actuator node has access to an infinite amount of energy to send the ACK with enough power. Moreover, an ACK is a short message (possibly reduced to a single bit), and hence it is possible to encode it with a very large redundancy for error correction.

If ACKs were not reliable, then the smart node would not know if the control update has been applied when no ACK is received. This would result in having the control memory at the smart node possibly different from the actual control input applied to the system, and the associated decision not optimal. However, successful control update transmission and ACK suffice to synchronise the control memory at the smart node side with the actual value applied to the system[2].

2.1.4 Switching policy and feedback law

The sensor node embeds a switching policy η (whose joint-design with the feedback law μ, described hereafter, is the goal of this thesis) to assign the radio-mode. The decision to switch between modes is

[1] It is reasonable to assume that the $\epsilon(m)$ are known as motivated in [Quevedo et al., 2010].

[2] When considering unreliable ACKs, several behaviours can be considered for the smart sensor in the case where no ACK is received. They depend mostly on the system and thus will not be discussed in details here. In a few words, depending on the criticality of the system, the difference between the memory and the actual value can be ignored, or the system can force a transmission until an ACK is successfully received. Again, a trade-off appears between energy consumption and closed-loop performance.

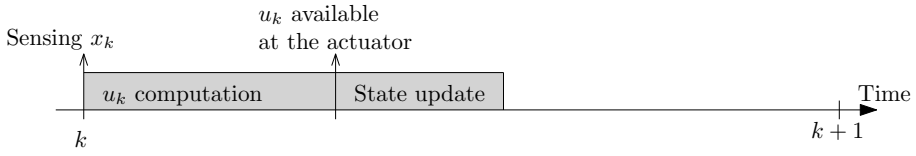
based on the actual system output x_k, the last control input u_{k-1} applied to the system and the current radio-mode, denoted m_k. Introducing $\tilde{u}_k = u_{k-1}$, the memory of the last control input, the switching decision is given by $\nu_k = \eta(x_k, \tilde{u}_k, m_k)$. The ACKs sent by the actuator node when a transmission is successful let the smart sensor have a perfect knowledge of the last control input applied to the system.

The control input applied to the system, denoted u_k, depends on the arrival of the update $\hat{u}_k$, which depends on the transmission success and on the decision to send an update, as described in Equation (2.2). If an update is received, then the control law is the optimal law computed by the smart sensor, denoted $\hat{u}_k = \mu(x_k, \tilde{u}_k, m_k)$ and derived on the next chapters[3]. Otherwise, the control input is held to its previous value as long as no update is received from the smart node:

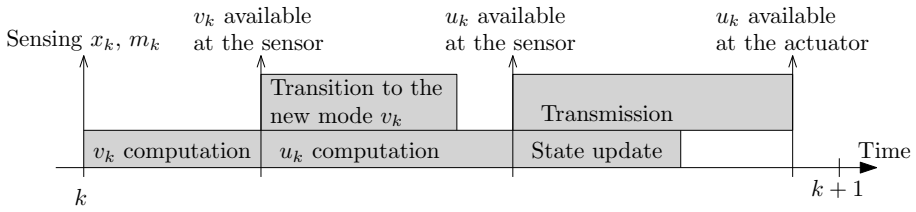
$$u_k = \begin{cases} \beta_k \hat{u}_k + (1-\beta_k) u_{k-1}, & \text{in case of transmission, } \textit{i.e.} \text{ if } \nu_k \in \mathbb{M}_1, \\ u_{k-1},\ \forall \beta & \text{otherwise.} \end{cases} \tag{2.2}$$

Thus, we have a Jump Linear System (JLS), where we decide upon the transition probabilities. This stands in contrast to many Markov JLS approaches, where transition probabilities are assumed given; see, *e.g.* [Do Valle Costa et al., 2005, Smith and Seiler, 2003].

Figure 2.2 clarifies how the computations of the control feedback and the switching policy are scheduled within a sampling interval with respect to the radio-mode transition in the case where $\nu_k \in \mathbb{M}_1$. In the common discrete-time control case (Figure 2.2(a)), the control input u_k is available after some computations and applied to the plant as soon as it is available. The rest of the sampling interval is used to update the state of the controller and to complete any computation that must be done before the next sampling interval. In our wireless case (Figure 2.2(b)), the switching decision ν_k is computed first. The state of the radio is switched according to the switching decision as soon as is available. In the case that a transmission is scheduled, this allows to start the transmission of the control input u_k as soon as it is available. The control input is only available at the actuator when the transmission is completed (and if it is successful).
m_k is the mode of the radio at the beginning of the sampling interval. ν_k is the mode to which the transition is made, within the sampling interval. Hence, clearly, m_{k+1} will be equal to ν_k.



(a) Without wireless transmission.



(b) With wireless transmission.

***Figure 2.2** – A timeline of a standard control application (Figure 2.2(a)) compared to our wireless NCS case (Figure 2.2(b)) when a transmission is scheduled (*i.e. $\nu_k \in \mathbb{M}_1$*). This figure details at what time instant each information becomes available. Moreover, it illustrates that the state of the radio changes over the sampling interval.*

[3]Note that the control law μ may not depend on $x_k, \tilde{u}_k$ or m_k, but to keep consistency with the notation of the switching policy η, we denote $\mu(x_k, \tilde{u}_k, m_k)$ in the rest of this thesis.

Finally, Figure 2.2 just depicts that:

- the computation of the switching decision v_k must precede the computation of the control input u_k,
- the control input may need more time to be available (since more computation is needed),
- the mode switching at the radio chip can be operated in parrallel to the computation.

2.1.5 Transition costs model

A radio chip can be modeled by an automaton where:

- the states describe the states of the radio (*e.g.* `Transmitting`, `Idle`, `Sleep`),
- and the transitions model events, such as exogenous signals sent by the micro-processor (radio-mode switching requests) or intern stack overflows.

Some of the transitions are time-constrained, which implies that a given time is needed to switch from a state to another. Each of the states is characterised by a current consumption, so as the time-constrained transitions. Figure 2.3 provides a complete example of an automaton for the real radio chip CC1100 Low-Power Sub- 1 GHz RF Transreceiver from Texas Instrument [Chipcon Products]. This chip is considered for the numerical applications for the rest of this thesis, however, any other chip can be considered. An example of time constraint from [Chipcon Products] is that $721\mu s$ are needed to calibrate the frequency synthesiser while switching from a low-consuming mode to a transmitting mode. The automaton in Figure 2.3 does not capture the time constraints nor the fact that the radio is able to use several transmission levels. The state *Tx 19,20* holds several options about the transmission level that are not shown in this figure.

However, the level of detail of Figure 2.3 is not suited to the present work. Thus we derive our own automaton in Figure 2.4, similar to the one from [Chipcon Products].

The number of states of the automaton is exactly the number of radio-modes under consideration, N, with several states for the several transmitting modes (one per power level). The modes that allow a transmission are numbered 1 to N_1 and the ones that do not transmit and save energy are numbered N_1+1 to $N = N_1 + N_2$, as explained before.

The derivation of the transitions in our automaton is more evolved. On one hand, we consider that the state of the automaton only changes at the sampling instant. In other words, the radio chip stays in a given state for the whole sampling interval and may only switch periodically at the sampling instant. On the other hand, the cost associated to the transitions have to take into account the current and time constraints imposed by the radio chip, and to respect the automaton as given by Figure 2.3. The transition costs in Figure 2.4 give the energy consumed during one sampling interval for any transition pattern.

These costs can be represented with a matrix, Θ, where the elements, denoted $\theta_{i,j}$, describe the energy needed to switch from the mode i to the mode j. The matrix that corresponds to Figure 2.4 for $N_1 = 3$ and $N_2 = 2$ is given by:

$$\Theta = \begin{bmatrix} \theta_{\mathtt{L1}} & \theta_{\mathtt{L2}} & \theta_{\mathtt{L3}} & \theta_{\mathtt{L1,Idle}} & \theta_{\mathtt{L1,Sleep}} \\ \theta_{\mathtt{L1}} & \theta_{\mathtt{L2}} & \theta_{\mathtt{L3}} & \theta_{\mathtt{L1,Idle}} & \theta_{\mathtt{L1,Sleep}} \\ \theta_{\mathtt{L1}} & \theta_{\mathtt{L2}} & \theta_{\mathtt{L3}} & \theta_{\mathtt{L1,Idle}} & \theta_{\mathtt{L1,Sleep}} \\ \theta_{\mathtt{Idle,L1}} & \theta_{\mathtt{Idle,L2}} & \theta_{\mathtt{Idle,L3}} & \theta_{\mathtt{Idle}} & \theta_{\mathtt{Idle,Sleep}} \\ \theta_{\mathtt{Sleep,L1}} & \theta_{\mathtt{Sleep,L2}} & \theta_{\mathtt{Sleep,L3}} & \theta_{\mathtt{Sleep,Idle}} & \theta_{\mathtt{Sleep}} \end{bmatrix}$$

where $\theta_{\mathtt{mode}}$ is a shortcut notation for $\theta_{\mathtt{mode,mode}}$ which is the energy needed to stay in `mode`, $\theta_{\mathtt{mode1,mode2}}$ is the energy needed to switch from `mode1` to `mode2`, and $\theta_{\mathtt{Li}}$ is the energy needed to send a control update using power level i.

The matrix Θ is just a notation to gather the cost terms.

At the radio chip level, the cost to change the transmission level can be neglected, which implies that $\theta_{\mathtt{Li,Lj}} = \theta_{\mathtt{Lj}}$, $\forall \mathtt{i} \in \mathbb{M}_1$. Similarly the cost $\theta_{\mathtt{Li,mode}}$ is the same for all $\mathtt{i} \in \mathbb{M}_1$ and for all $\mathtt{mode} \in \mathbb{M}_2$. However, a transition to a transmitting mode embeds both the transition cost to the new mode and the transmission cost, as it is explained in the next paragraph. For this reason $\theta_{\mathtt{Li,mode}} \neq \theta_{\mathtt{Lj,mode}}$ for $(\mathtt{i},\mathtt{j}) \in \mathbb{M}_1^2$ and for $\mathtt{mode} \in \mathbb{M}_2$. Even though it is often impossible to switch between modes that are

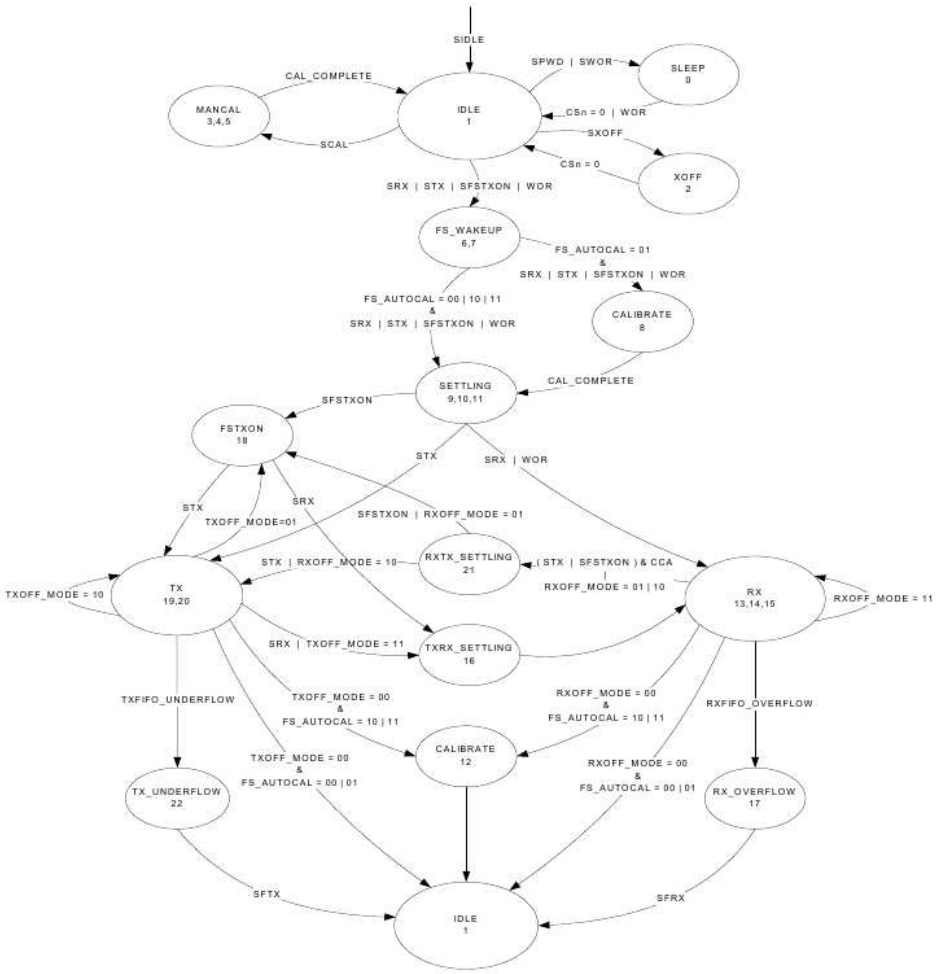


Figure 2.3 *– Figure 16 from [Chipcon Products] describing the automaton of the radio chip CC1100 Low-Power Sub- 1 GHz RF Transreceiver from Texas Instrument. The states are the radio-modes while the transitions are events. Time-constraints and current consumption are omitted. The details can be found in [Chipcon Products].*

not consecutive at the radio chip level, one can consider a transition from `Sleep` to any transmitting mode by considering an implicit transition to the `Idle` mode. Indeed, the costs $\theta_{i,j}$ describe the consumption of the radio over a sampling interval, which may include several state switchings, as long as the sampling interval is large enough to allow this.

This model assumes that the mode transition time-constraints are smaller than the sampling interval. This implies that the smart node has enough time to switch to the desired mode (and possibly send the update) before the next sampling time.

This model captures the exact behaviour of the radio as it is intended to be used in our setup. Some of the costs are easily computed from the data sheet. For instance, the energy cost to stay in the `Sleep` mode is directly computed from the current consumption of this mode, the operating voltage of the chip (*e.g.* both given in the data sheet [Chipcon Products]) and the duration of a sampling interval. However, it has to be mentioned that the computation of certain transition costs is not straightforward. Indeed, the costs result in the consumption of the radio chip on a sampling interval for a given behaviour, which is left to the designer's choice. For instance, when considering the cost $\theta_{\texttt{Sleep,Idle}}$, one can consider to switch the state of the radio at the end of the sampling interval rather than at the beginning to save more energy, as it is illustrated in Figure 2.5. Another example is the calibration step needed to perform a transmission which is usually proceeded when switching towards a transmitting mode after stayed in a low-consuming mode; this task can also be proceeded when transiting from `Tx` to `Idle` or when staying in `Idle` to save time at the transmission instant.

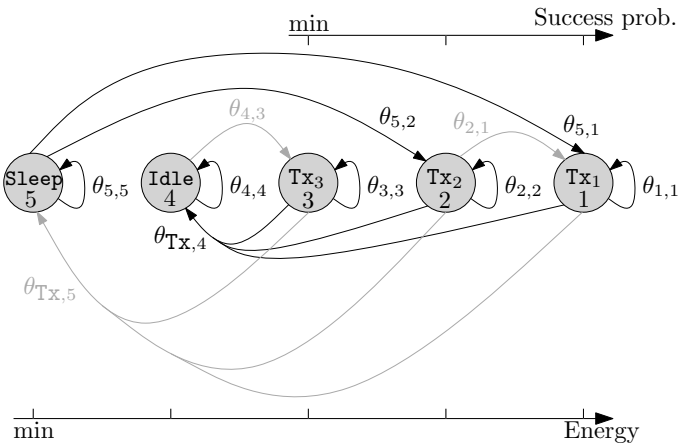


***Figure 2.4** – Illustration of the transition costs with $N_1 = 3$ and $N_2 = 2$.* `Idle` *is an intermediate mode between the transmitting modes and the* `Sleep` *mode. Modes are ordered according to their energy consumption,* e.g. *$\theta_{5,5} < \theta_{4,4}$ where $\theta_{5,5}$ is the energy cost to stay in mode* `Sleep` *and $\theta_{4,4}$ is the energy cost to stay in mode* `Idle`*. Transmitting modes with a higher energy consumption have a higher probability to transmit successfully. The arrows represent the transition costs. A more detailed figure would make appear every transition $\theta_{i,j}$ from any mode i to any mode j.*

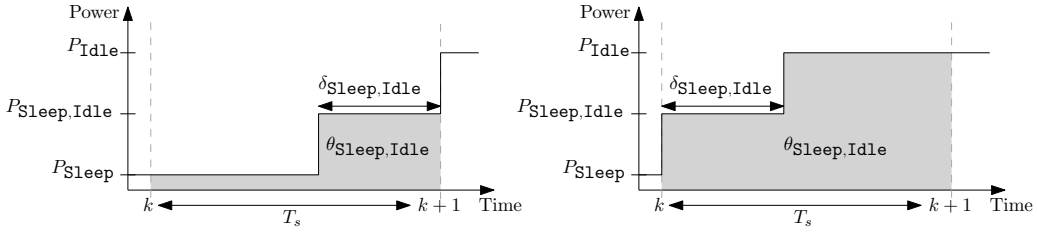


***Figure 2.5** – Radio-mode transition on a timeline. $\theta_{\texttt{Sleep,Idle}}$ is computed for a given sampling time T_s given the power consumption at the* `Idle` *state $P_{\texttt{Idle}}$, the* `Sleep` *state $P_{\texttt{Sleep}}$ and the power $P_{\texttt{Sleep,Idle}}$ and delay $\delta_{\texttt{Sleep,Idle}}$ needed for the transition. This figure gives two different ways to compute $\theta_{\texttt{Sleep,Idle}}$ (the area under the curve), showing that the computations of the transition costs are subject to the designer's choices. It is purely illustrative since in this precise case, the left hand side choice is obviously the most energy efficient.*

In other words, one has to understand that the modes (and their costs) at the application layer are different from the modes at the radio chip level. The fact that the designer can tune the transition costs is an advantage. It offers flexibility to adapt the costs according to the radio specification but also to the application needs. As an illustration, we can mention that in the rest of this work the transmitting modes (at the application layer) are derived to consider the reception of the ACK, by switching to the receipting mode and waiting for an appropriate duration.

Scenario in which to apply our technique

Our approach considers several low-consuming modes in order to save more energy than a simple `On/Off` pattern and which provides a higher resolution grid to trade energy cost with quality of the sended information. However, as it has been highlighted previously, because of transition costs and time-constraints, switching to intermediate low-consuming modes may result in more energy waste than holding a transmitting mode. The joint switching policy and feedback control that we derive in this thesis allow to reduce the expended energy only under some assumptions on the transition costs of the Θ matrix.

Assumption 2.1. *$\theta_{i,i} \geq \theta_{j,j} > 0$, $\forall i \in \mathbb{M}_1$ and $j \in \mathbb{M}_2$. This means that the transmitting modes consume more than the non-transiting modes and that the cost to stay in a given mode for a sampling period is always positive.*

We introduce the convention to number the modes according to their energy costs, namely $\theta_{1,1} \geq \theta_{2,2} \geq \ldots \geq \theta_{N,N}$. The amount of energy that can be saved when using the low-consuming modes is directly related to the difference between $\theta_{1,1}$ and $\theta_{N,N}$.

Assumption 2.2. *For any $j_1 \in \mathbb{M}_2$ and $j_2 \in \mathbb{M}_2$, if $j_1 < j_2$ then $0 < \theta_{i,j_1} \leq \theta_{i,j_2}$, $\forall i \in \mathbb{M}_1$. This means that the transition cost from a transmitting mode to a low-consuming mode is larger for deeper low-consuming mode. The symmetric condition (from a low-consuming mode to a transmitting one) is also assumed to hold,* i.e. *for any $j_1 \in \mathbb{M}_2$ and $j_2 \in \mathbb{M}_2$, if $j_1 < j_2$ then $0 < \theta_{j_1,i} \leq \theta_{j_2,i}$, $\forall i \in \mathbb{M}_1$.*

The way radio chips are designed enforces that the transition from or to a low-consuming mode has a larger cost when the low-consuming mode saves more energy (*i.e.* turn on or off more components in the radio chip). However, this does not imply that Assumption 2.2 holds because the costs represented by the Θ matrix are based on the energy consumed over an entire sampling interval. As it has been introduced in the previous subsection and in Figure 2.5, the costs depend on the sampling interval, the time-constraints and the choices of the design (which is related to the application). The time-constraints in the radio chip are commonly accounted in microseconds while control applications often consider sampling interval in the range of milliseconds, this makes the costs of the intermediate low-consuming modes very close to each others, which induced that the deeper low-consuming mode may be mostly preferred over the other intermediate modes. In other words, the energy savings introduced with the use of the intermediate modes are more important when the sampling interval at the control level is close to the time-constraints of the radio chip.

2.2 Switched model

The evolution of the system under the different choices of radio-mode is now formulated as a switched linear system, with as many systems as the number of modes N. From a control point of view, the different modes are actually reduced to two cases: when a successful transmission occurs (*i.e.* the control loop is closed) and when the system runs open loop. The different modes affect the energy consumption and also the success probability for the modes allowing transmission.

Choosing the switching policy at time k is equivalent to choosing the radio-mode. The evolution of the switched system depends on x_k, the state of the system, on $\tilde{u}_k = u_{k-1}$, the memory keeping track of the last applied control input, and on m_k the mode of the radio chip. We define z_k as the system state augmented with the control memory:

$$z_k = \begin{bmatrix} x_k \\ \tilde{u}_k \end{bmatrix} \in \mathbb{R}^{n_x+n_u}, \text{ and also } \omega_k = \begin{bmatrix} w_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_x+n_u}.$$

Note that we also augment the noise w_k to the vector ω_k to fit the dimension of z_k, see Equation (2.4).

Then, the state of the switched system is the following:

$$(z_k, m_k),$$

so that the state space of the switched system is $\mathbb{X} = \mathbb{R}^{n_x+n_u} \times \mathbb{M}$. The evolution of the system given in Equation (2.1) with the feedback law $\mu(x_k, \tilde{u}_k, m_k)$ described in Equation (2.2), together with the radio-mode switching policy $\eta(x_k, \tilde{u}_k, m_k)$ (both to be derived in the next chapters), give rise to the following switched system:

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ \hat{u}_k = \mu(z_k, m_k), \end{cases} \tag{2.3}$$

where the function f_{ν_k} is defined as

$$f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) = \Phi_{\nu_k}(\beta_k) z_k + \Gamma_{\nu_k}(\beta_k) \hat{u}_k + \omega_k, \tag{2.4}$$

and the matrices $\Phi_{\nu_k}(\beta_k), \Gamma_{\nu_k}(\beta_k)$, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $v_k \in \mathbb{M}_1$, *i.e.* if there is a transmission, then

$$\Phi_{v_k}(\beta_k) = \begin{bmatrix} A & (1-\beta_k)B \\ \mathbf{0} & (1-\beta_k)\mathbf{I} \end{bmatrix} = \begin{cases} \Phi_{CL} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \text{if } \beta_k = 0, \end{cases}$$

$$\Gamma_{v_k}(\beta_k) = \beta_k \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} = \begin{cases} \Gamma_{CL} = \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} & \text{if } \beta_k = 1 \\ \Gamma_{OL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 0, \end{cases}$$

2. if $v_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$\Phi_{v_k}(\beta_k) = \Phi_{OL}, \ \forall \beta_k,$$
$$\Gamma_{v_k}(\beta_k) = \Gamma_{OL}, \ \forall \beta_k.$$

2.3 Models for alternative architectures

The problem formulation described in Section 2.1 and 2.2 is based on the assumption that the computation of the control law μ is performed at the smart node side. However, the same model can be easily adapted to describe some cases where this computation is done at the actuator side. In this section, we discuss different alternative architectures, that we will not consider any further in the rest of this thesis. First, we detail the different systems that can be considered with the same switched formulation. Then, we discuss other systems that are related to our model explaining the difficulties that they bring with respect to our switched formulation.

2.3.1 Other setups described by the same model

a) Equivalent setups where sensor communicates state instead of control If the dimension of the control space is larger than the dimension of the state space, and if the actuator has computation capabilities, then the sensor can send x_k, the system state, instead of $\hat{u}_k$, the proposed control input, and let the receiver node compute the control update to be applied to the system, as illustrated in Figure 2.6 and Figure 2.7. However, it has to be noticed that m_k, the state of the radio chip, is also needed to compute the feedback law, and it has to be sent with the system state. An advantage to this setup is that the controller has a perfect knowledge of the last control input applied to the system. However, the smart node still needs to know what control update has been applied to the system in order to run the radio-mode switching policy. In Figure 2.6, the smart node computes the control update itself, and wait for the ACK from the controller to confirm that the new system state has been received by the actuator node and then that the control input has indeed been updated.

The main drawback of this approach is that the control update is computed at both sides. This can be relaxed by separating the control law (at the actuator side) from the radio-mode switching policy

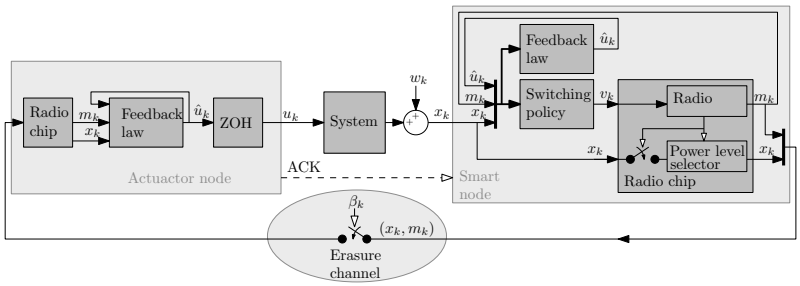


Figure 2.6 *– Block diagram of an alternative setup where the control update is computed at the actuator side rather than at the sensor side. In this case, the system state is sent instead of the control update, and the sensor also compute the control update in order to know the control input applied to the system (which is possible thanks to the ACK).*

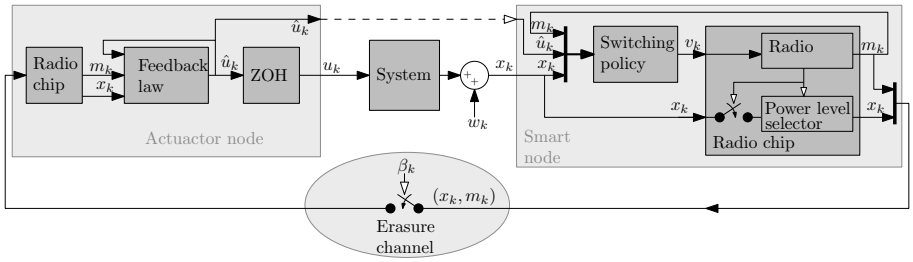


Figure 2.7 *– Block diagram of an alternative setup where the control update is only computed at the actuator side and sent back to the smart sensor as an ACK, needed to compute the switching policy.*

(at the sensor side), as proposed in Figure 2.7. This can be done under the condition that the controller sends back to the sensor the control law that has been applied to the system, rather than a simple ACK. However, while it is reasonable to assume that the ACK is always successfully transmitted from the actuator to the smart sensor, this may not be the case for a real value or a vector. Possible dropout from the controller to the smart sensor brings the drawback to have different control input memory at the sensor and the controller side, as it has already been discussed in Section 2.1.3. An important advantage of this alternative is that the computation at the smart sensor side is lightened. It can be the approach with the lightest computation burden at the sensor side if the radio-mode switching policy is reduced to a look-up table, as it is explained in Section 3.2.3.

2.3.2 Simplified setups

b) Memoryless actuation A Zero Order Holder (ZOH) is used in our model in order to hold the control input to the previous value when no new update is received from the smart sensor. However, as it is discussed in [Schenato, 2009], in terms of control performance, holding the control input is not necessarily better than applying a zero input when the update is dropped in lossy NCS. For that reason, the current alternative proposes to apply a zero control input when no update is received, as depicted in Figure 2.8. This alternative being a simplification of the general model, the results proposed in the rest of this thesis also hold in this case. One just has to simplify the matrices of the switched model as follows:

$$\Phi_{CL} = \Phi_{OL} = A,$$
$$\Gamma_{CL} = B,$$
$$\Gamma_{OL} = \mathbf{0}.$$

In this case the memory part of z_k, the augmented state, is no longer needed. The dimension of the switched system is equal to the dimension of the system to be controlled: $x_k = z_k$.

c) Static state feedback controller The goal of the next chapters is to derive a joint radio-mode switching policy and feedback control law. However, a slight modification of the switched model allows to use the same method in order to derive only the switching policy, under the assumption that the control law is a linear feedback with a given static gain. This is illustrated in Figure 2.9. By denoting K the static state feedback gain, so that $\hat{u}_k = -Kx_k$, the corresponding switched model has

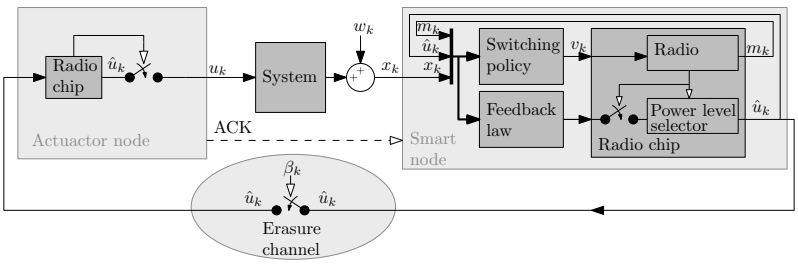


Figure 2.8 *– Block diagram of an alternative setup where the control input applied to the system is zero when no control update is received.*

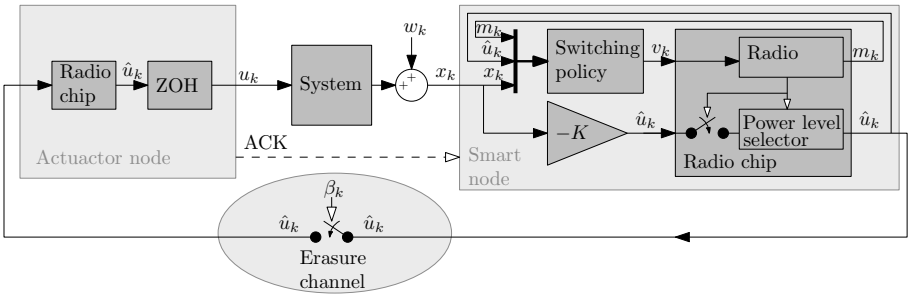


Figure 2.9 *– Block diagram of an alternative setup where the control law is a static gain feedback.*

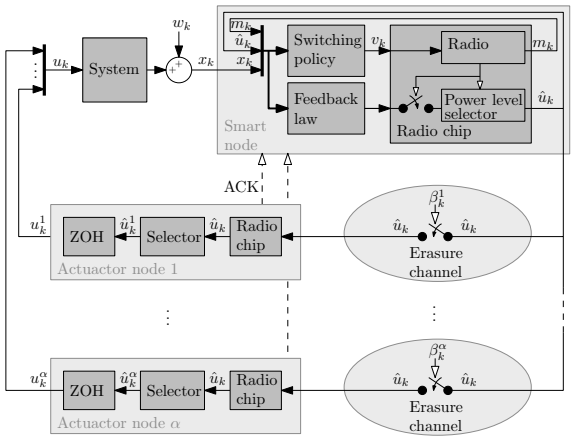


Figure 2.10 *– Block diagram of an alternative setup which includes several actuators. The control input is broadcasted from the smart sensor to all actuators, introducing different erasure probabilities. Each actuator embeds a selector that is used to select only the components of the control input vector that correspond to itself. The smart sensor receives an ACK from each actuator.*

the following matrices:

$$\Phi_{CL} = \begin{bmatrix} A - BK & \mathbf{0} \\ -K & \mathbf{0} \end{bmatrix}$$
$$\Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$
$$\Gamma_{CL} = \Gamma_{OL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Clearly in this case, the feedback law is given by $\hat{u}_k = \mu(x_k, \tilde{u}_k, m_k) = -Kx_k$.
We will use this setup in Section 3.2.

d) Several actuator nodes Our model assumes that the network is only composed of two nodes, the smart sensor and the actuator node. The natural extension to several nodes can consider several sensors or several actuators. The former is discussed in the next subsection while the latter is considered in the current alternative. As it can be seen on Figure 2.10, several actuator nodes can be considered under the assumption that the sensor broadcasts the control input vector to all actuator nodes at once. Each actuator node only uses the element of the control input vector that is intended to itself. This alternative is quite similar to our model, except for the radio-mode management of the several transmitting modes. Indeed, while $\epsilon(\nu_k)$ captures the dropout probability (under radio-mode ν_k) for the only link that exists in our model, in this case, there are as many links as the number of actuator nodes (denoted α), and thus as many dropout probabilities $\epsilon^i(\nu_k)$ (for $i \in \{1, \ldots, \alpha\}$) as the number of actuator nodes. In this case, the Bernoulli random variable is described by a diagonal matrix Υ_k which

the i^{th} diagonal terms, denoted β_k^i, are given by the following probabilities[4]:

$$\mathbb{P}\left\{\beta_k^i = 0 | m_k = m\right\} = \epsilon^i(m),$$
$$\mathbb{P}\left\{\beta_k^i = 1 | m_k = m\right\} = 1 - \epsilon^i(m).$$

Then the matrices $\Phi_{\nu_k}(\Upsilon_k), \Gamma_{\nu_k}(\Upsilon_k)$, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k \in \mathbb{M}_1$, *i.e.* if there is a transmission, then

$$\Phi_{\nu_k}(\Upsilon_k) = \begin{bmatrix} A & (\mathbf{I} - \Upsilon_k)B \\ \mathbf{0} & \mathbf{I} - \Upsilon_k \end{bmatrix},$$
$$\Gamma_{\nu_k}(\Upsilon_k) = \begin{bmatrix} \Upsilon_k B \\ \Upsilon_k \end{bmatrix}.$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$\Phi_{\nu_k}(\Upsilon_k) = \Phi_{OL},\ \forall \Upsilon_k,$$
$$\Gamma_{\nu_k}(\Upsilon_k) = \Gamma_{OL},\ \forall \Upsilon_k.$$

Another difference from our model is that the smart sensor needs to receive the ACKs from every actuator node in order to update correctly its own memory of the control input actually applied to the system.

Finally, it can be noticed that rather than broadcasting the control input vector, the smart sensor can broadcast the state of the system, letting each controller to compute the control update. This brings the same benefits and drawbacks than the alternative discussed in Subsection a).

2.3.3 Discussion about other setups

This section discusses other setups, that are natural extensions from our model, but which need important work to be addressed. Therefore we only introduce these extensions and we rapidly discuss why the material presented in this thesis is not sufficient to address them. The aim of this subsection is to clarify the reason why such natural extensions are not treated in this thesis.

Quantisation An important simplifying assumption of the present work is that it considers that the transmitted data are real-valued. However in practice, information is quantised. Considering quantisation brings important effects on the closed-loop system, and it prevents from applying easily the current framework. More information about quantisation in NCS can be found in [Ishii and Basar, 2002, Yüksel, 2011].

Battery-driven actuator case Our model is focusing on saving energy at the sensor side. Depending on the application, the energy consumption may be more critical at the actuator side. Implementing radio-mode management on the controller rather than the sensor is not straightforward. Indeed, if the controller switches its radio off, then it remains unconnected from the network, and especially from the sensor. In our model, the fact that the sensor keeps monitoring the system even when its radio is switched off permits the smart node to re-activate the radio in case of need. When turning off the controller, only prediction can be used for waking up the radio. This problem is similar in the case where both sensor and actuator are battery-driven.

Interesting approaches to tackle this problem may be encountered in the framework of self-triggered control. A different approach can be considered where an extra hardware device can wake up the nodes. Indeed, some works are interested in considering nodes equipped with an RFID device which provides a way to wake up a sleeping node with an external signal, see [Jurdak et al., 2008, Ruzzelli et al., 2007]. In this particular context, when a node wants to send an information to a sleeping node, it first wakes it up with an RFID impulse and then it sends the information over the wireless channel. With this technique, the solution proposed in the next chapters can be adapted to save energy on both the smart sensor and the actuator node.

[4] We assume here that $\alpha = n_u$ to ease the definition of Υ_k but one can easily extend this scenario to $\alpha \leq n_u$. Having $\alpha < n_u$ implies that at least one of the control input signal is a vector (rather than a scalar as in the case where $\alpha = n_u$). In that case, the size of Υ_k is again n_u, but the element β_k^i for i being an actuator taking a vector as control input is repeated in the Υ_k matrix to fit the size of the vector.

Several smart sensor nodes While the case of several actuator nodes has already been discussed in the previous subsection, here the case of several smart sensor nodes is discussed. The approach that is used throughout this thesis assumes that the smart sensor (when single) has a perfect knowledge of the system state, the last control input applied and its current radio-mode to take a switching decision. In the case of several smart sensors, each of the sensor only has a partial knowledge of the state of the system. Although the sensors could exchange their knowledge, we precisely want to limit the amount of communications. This limitation prevents the solution derived in the next chapters to be extended to the several smart sensors case. This kind of problem is often tackled with a distributed approach.

2.4 Optimisation problem

In a general way, turning off the radio at the sensor side saves energy but makes the system run open loop, and thus deteriorates the feedback performance. The goal of the jointly designed switching policy and feedback law is to obtain a trade-off between the energy consumption and the feedback performance. The framework of Optimal Control is chosen to derive the joint policy since it solves an optimal problem based on a cost function. The cost function used throughout the rest of this thesis explicitly accounts for the two criteria under focus, namely the feedback performance and the energy consumption. First, we define the cost-to-go, also called step cost, denoted ℓ, as the cost that is payed by the system over one sampling interval. This cost-to-go depends on the the state of the system x_k, on the state of the radio chip m_k, on the control input u_k, and on the switching decision ν_k:

$$\ell_{\nu_k}(x_k, m_k, u_k) = \underbrace{x_k^\top \bar{Q} x_k}_{\text{performance}} + \underbrace{u_k^\top \bar{R} u_k}_{\text{control energy}} + \underbrace{\theta_{m_k,\nu_k}}_{\text{transition energy}}. \tag{2.5}$$

$\bar{Q}$ and $\bar{R}$ are symmetric positive definite matrices which can be tuned to give different trade-offs between the feedback performance and energy consumption. The transition energy, or transition cost, θ_{m_k,ν_k} has been introduced in Subsection 2.1.5, and it includes the transmission energy when a transmission is scheduled.

As detailed in Equation (2.2), the control input depends on the switching decision ν_k and on the success of the transmission described by β_k. Thus the cost-to-go can be written as follows:

$$\ell_{\nu_k}(x_k, m_k, u_k, \beta_k) = \begin{cases} x_k^\top \bar{Q} x_k + \beta_k \hat{u}_k^\top \bar{R} \hat{u}_k + (1-\beta_k)\tilde{u}_k^\top \bar{R} \tilde{u}_k + \theta_{m_k,\nu_k}, & \text{in case of transmission,} \\ x_k^\top \bar{Q} x_k + \tilde{u}_k^\top \bar{R} \tilde{u}_k + \theta_{m_k,\nu_k}, & \text{otherwise.} \end{cases}$$

Finally, using the notation introduced in Section 2.2:

$$\ell_{\nu_k}(z_k, m_k, \hat{u}_k, \beta_k) = z_k^\top Q_{\nu_k}(\beta_k) z_k + \hat{u}_k^\top R_{\nu_k}(\beta_k) \hat{u}_k + \theta_{m_k,\nu_k}, \tag{2.6}$$

where the matrices $Q_{\nu_k}(\beta_k)$ and $R_{\nu_k}(\beta_k)$, for $\nu_k \in \mathbb{M}$, are defined as follows:

1. if $\nu_k \in \mathbb{M}_1$, *i.e.* if $u_k = \beta_k \hat{u}_k + (1-\beta_k)\tilde{u}_k$, then

$$Q_{\nu_k}(\beta_k) = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & (1-\beta_k)\bar{R} \end{bmatrix} = \begin{cases} Q_{CL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} & \text{if } \beta_k = 0, \end{cases}$$

$$R_{\nu_k}(\beta_k) = \beta_k \bar{R} = \begin{cases} R_{CL} = \bar{R} & \text{if } \beta_k = 1 \\ R_{OL} = \mathbf{0} & \text{if } \beta_k = 0. \end{cases}$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if $u_k = \tilde{u}_k$, then

$$Q_{\nu_k}(\beta_k) = Q_{OL}, \ \forall \beta_k,$$
$$R_{\nu_k}(\beta_k) = R_{OL}, \ \forall \beta_k.$$

The cost function $J_{\mathscr{U},\mathscr{V}}$ accounts for the expected cost the system has to pay when controlling the system (2.1) with the feedback sequence $\mathscr{U} = \{\mu_0, \mu_1, \ldots, \mu_{H-1}\}$ and the switching sequence $\mathscr{V} = \{\eta_0, \eta_1, \ldots, \eta_{H-1}\}$, over a time horizon H with the initial condition z_0, m_0:

$$J_{\mathscr{U},\mathscr{V}}(z_0, m_0) = \mathop{\mathsf{E}}_{\substack{\beta_k, \omega_k \\ k=0,1,\ldots}} \left[\ell_F(z_H, m_H) + \sum_{k=0}^{H-1} \lambda^k \ell_{v_k}(z_k, m_k, \hat{u}_k, \beta_k) \right], \tag{2.7}$$

with $\hat{u}_k = \mu_k(z_k, m_k)$, $v_k = \eta_k(z_k, m_k)$, $z_{k+1} = f_{v_k}(z_k, \hat{u}_k, \beta_k, \omega_k)$, and $\ell_F(z_H, m_H)$ being the final cost, *i.e.* a function that accounts for the state of the system at the end of the horizon, and $\lambda \in [0, 1)$ is a discount factor discussed in Section 3.1. $J_{\mathscr{U},\mathscr{V}}$ is a function of the initial conditions only, computed with the expectation of the random variables. The time horizon H is finite, but can be considered infinite by taking the limit of the right hand side of (2.7) when H goes to infinity. In the infinite case, no final cost is considered.
The optimisation problem considered for the rest of this work consists in finding a feedback sequence $\mathscr{U}^*$ and a switching policy $\mathscr{V}^*$ such that:

$$J^*(z_0, m_0) \triangleq J_{\mathscr{U}^*,\mathscr{V}^*}(z_0, m_0) = \min_{\mathscr{U},\mathscr{V}} J_{\mathscr{U},\mathscr{V}}(z_0, m_0). \tag{2.8}$$

Note that the minimum is indeed attained, *i.e.* it is actually a minimum and not an infimum, as proved in Section 3.3.3.

Remark 2.1. *Chapter 3 considers that H is infinite while Chapter 4 considers a finite receding time horizon. In the infinite case, the sequence $\mathscr{U} = \{\mu_0, \mu_1, \ldots\}$ and $\mathscr{V} = \{\eta_0, \eta_1, \ldots\}$ would have an infinite number of elements. For a matter of implementation, we consider stationary policies μ and η such that $\mathscr{U} = \{\mu, \mu, \ldots\}$ and $\mathscr{V} = \{\eta, \eta, \ldots\}$, and we simply denote the cost function $J_{\mu,\eta}$. In the finite case, since we will consider a receding horizon, we will also derive stationary policies. This is discussed more in more details in each of these chapters.*

Remark 2.2. *According to Assumptions 2.1 and 2.2, the optimisation problem is considered under the non-triviality assumption that transmissions have a non-zero cost whatever the previous mode,* i.e. $\theta_{m,m_1} > 0$, $\forall m \in \mathbb{M}$, $\forall m_1 \in \mathbb{M}_1$. *Indeed, not only the case where a mode allows transmission free of cost is not realistic, but also it results in an optimal radio-management which always transmits.*

The control space is constrained by the switching decision. Indeed, if no transmission is scheduled, then the control input is forced to the memory value $\tilde{u}$. We thus define the control space $\mathbb{U}(\tilde{u})$ as follows:

Definition 2.1.

$$\mathbb{U}(\tilde{u}) \triangleq \left\{(u, v) : u \in \mathbb{R}^{n_u},\ v \in \mathbb{M}_1\right\} \cup \left\{(\tilde{u}, v) : v \in \mathbb{M}_2\right\}.$$

For ease of notation, we will use $\mathbb{U}(z)$ instead of $\mathbb{U}(\tilde{u})$. The joint policy (μ, η) must take its values in $\mathbb{U}(z)$.

2.5 Dynamic Programming

2.5.1 Bellman's Principle of Optimality

The framework of Dynamic Programming provides methods to solve optimisation problems which can be decomposed in nested sub-problems. We consider the Value Iteration method to solve the optimisation problem described in Section 2.4. This method is based on Bellman's Principle of Optimality [Bellman, 1957], stated as follows:

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

In other words, if a trajectory from a point A to a point C is the optimal path between these points, then for any point B on this trajectory, the optimal path between B and C is on the same trajectory, as depicted in Figure 2.11.

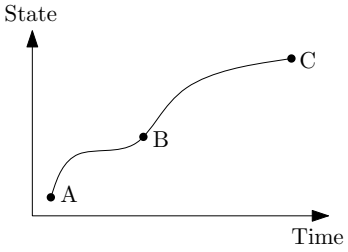


***Figure 2.11** – Illustration of the Principle of Optimality, the trajectory that is drawn is the optimal path between A and C, then the optimal path from B to C is what remains of the same path.*

2.5.2 The Value Iteration method in the finite horizon case

Note that in the finite horizon case, the discount factor is taken equal to 1 and thus ignored in Equation (2.9) and in the rest of this subsection, as motivated in Section 4.1.

Based on Bellman's Principle of Optimality, the Value Iteration method is a tool provided by the framework of Dynamic Programming that exploits the fact that if we know all the optimal paths from time $k+1$ to time H, then every optimal path from time k must use one of the optimal paths from time $k+1$. We denote $J^*_{H-i}(z,m)$ the optimal cost function for the same problem starting at time $k=H-i$ over an horizon i, $0<i\leq H$. The Bellman's Principle of Optimality leads to the following relation:

$$J^*_{H-i}(z,m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + J^*_{H-i+1}(f_v(z,\hat{u},\omega,\beta),v) \right] \right\}. \tag{2.9}$$

The Value Iteration method consists in computing backward in time (starting from the final cost at time H, $J^*_H(z_H,m_H)=\ell_F(z_H,m_H)$) the optimal cost $J^*=J^*_0$. The Value Iteration method is a recursion that computes the so-called Value Function, denoted $V_i(z,m)$, at each time step i from H to 0, based on the recursion (2.9).
The recursion of the Value Function, up to iteration H, $\forall (z,m)\in\mathbb{X}$, is given by (see [Bertsekas, 2005a, Prop. 1.3.1]):

$$\begin{aligned} V_0(z,m) &\triangleq \ell_F(z,m) \\ V_{i+1}(z,m) &= \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + V_i(f_v(z,\hat{u},\omega,\beta),v) \right] \right\}. \end{aligned}$$

Note that the iteration index of the Value Function i goes backward in time. This means that the Value Function and the cost function have the following relation: $V_i(z,m)=J^*_{H-i}(z,m)$ and we notice that $V_H(z,m)=J^*(z,m)$. Note also that the minimum is indeed attained, as proved in Section 3.3.3.

The Value Function does not only provide the optimal cost, but also an optimal associated policy. We define $(\mu^*_i(z,m),\eta^*_i(z,m))$ as the joint policy at iteration i:

$$(\mu^*_i(z,m),\eta^*_i(z,m)) \triangleq \mathop{\arg\min}_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + V_i(f_v(z,\hat{u},\omega,\beta),v) \right] \right\}, \tag{2.10}$$

then an optimal policy over horizon H is given by the sequence $(\mu^*_0,\eta^*_0),\ldots,(\mu^*_{H-1},\eta^*_{H-1})$. This technique does not ensure that this optimal sequence is unique. Especially if several policies provide the same minimum, any of them can be used.

2.5.3 The Value Iteration method in the infinite horizon case

A similar procedure can be derived in the infinite case, taking into account the discount factor λ which is no longer assumed to be equal to 1. The recursion from Bellman's Principle of Optimality, given by Equation (2.9) in the finite case, becomes with the discount factor:

$$J^*_{H-i}(z,m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\lambda^{H-i}\ell_v(z,m,\hat{u},\beta) + J^*_{H-i+1}(f_v(z,\hat{u},\omega,\beta),v) \right] \right\}.$$

The recursion on the Value Function, as detailed in [Bertsekas, 2007, Chap. 1], is given,

$\forall (z,m) \in \mathbb{X}$, by:

$$\begin{aligned} V_0(z,m) &= g_F(z,m) \\ V_{i+1}(z,m) &= \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + \lambda V_i(f_v(x,\hat{u},\omega,\beta),v) \right] \right\}. \end{aligned} \tag{2.11}$$

The function $g_F(z,m)$ is a re-scale final cost function which definition comes naturally from the relation that exists between the Value Function and the cost function, as it will be highlighted later in Equation (2.12).

The Value Iteration method ensures that with the initialisation $V_0(z,m) = 0$ for all $(z,m) \in \mathbb{X}$, $V_i(z,m)$ converges, as $i \to \infty$, to $J^*(z,m) \triangleq \min_{\mu,\eta} J_{\mu,\eta}(z,m)$. This is detailed in Section 3.3.2.

The method also provides an optimal policy. Similarly to the finite case, we define $(\mu_i(z,m), \eta_i(z,m))$ the joint policy at iteration i:

$$(\mu_i(z,m), \eta_i(z,m)) \triangleq \mathop{\arg\min}_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + \lambda V_i(f_v(z,\hat{u},\omega,\beta),v) \right] \right\},$$

then an optimal policy is given by any limit point of the sequence (μ_i, η_i), denoted $(\mu_\infty, \eta_\infty)$. As in the finite case, this optimal policy is not necessarily unique.

In the infinite case, the Value Function is related to the cost function with the following relation:

$$V_i(z,m) = \frac{J^*_{H-i}(z,m)}{\lambda^{H-i}}.$$

And we define that the re-scaled final cost $g_F(z,m)$ as a function of the final cost $\ell_F(z,m)$:

$$\ell_F(z,m) = \lambda^H g_F(z,m). \tag{2.12}$$

Note that the notion of final cost is only used for the initialisation of the Value Iteration method. Indeed, the horizon length H will be taken to infinity, which means that there is no final stage.

It is worth noting that, under proper assumption on the initialisation stage, the computation of the optimal cost function over an horizon $H+1$ is directly obtained from Equation (2.11), when the optimal cost function over an horizon H is already computed. This supports the intuition that taking the limit of the recursion as H goes to infinity converges to the optimal cost function J^*. The convergence is proved in Section 3.3.3.

Finally, it can be noticed that the Bellman's equation in the infinite case is given by:

$$J^*(z,m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + \lambda J^*(f_v(z,\hat{u},\beta,\omega),v) \right] \right\}.$$

Comprehensive details about the Value Iteration method both in the finite and infinite cases can be found in [Lincoln, 2003, Bertsekas, 2005a, 2007].

2.5.4 Implementation

The method we have presented to solve the optimisation problem is composed of two parts.

The first part is run offline, and it provides the optimal joint policy to switch between modes and to control the feedback loop. It consists in iterating the Value Function $V_i(z,m)$ to derive an optimal feedback law $\mu^*(z,m)$ and switching policy $\eta^*(z,m)$ for all $(z,m) \in \mathbb{X}$. This part involves heavy computations that can be run offline on a computer. It provides a solution for a set of state (z,m). Note that we want to implement a stationary policy, even in the finite case which provides a sequence of policies rather than a single policy. But, as it will be explained in Chapter 4, the use of the receding horizon provides a stationary policy.

The second part, run online on the smart sensor, consists in computing the optimal switching decision, v_k^*, only for the current switched system state (z_k, m_k) at time k, according to the switching policy derived offline. If the switching decision schedules a transmission, then the update of the control law is computed for the current state, also from the feedback law derived offline.

This two parts procedure will be described in details in the next chapters.

A drawback of Dynamic Programming is that it may be difficult to derive an analytical closed-form for the Value Function. This can increase the computational burden in the smart sensor. To tackle this

issue, it is possible to compute offline optimal switching decisions and feedback control inputs for a finite number of states, on a given grid on a portion of the state space. This ends up with a look-up table, easily implementable on nodes with limited computational units. The number of values on the grid can be set arbitrarily to meet the precision needed by the final application.

Chapter 3

Infinite horizon optimisation for energy-aware control

Contents

The optimal problem introduced in Chapter 2 consists in deriving a joint radio-mode switching policy and feedback control law, in order to save energy at the smart sensor side in a two-nodes Networked Control System (NCS). The problem is formulated over a certain horizon, which we propose to consider infinite in this chapter, and finite in Chapter 4. For the rest of this chapter, the cost function (2.7) is considered when $H \to \infty$, and the sequences $\mathscr{U}, \mathscr{V}$ are replaced by the stationary policy (μ, η), as explained in Remark 2.1.

After discussing the validity of the cost function, we consider first the ideal case (without dropouts nor feedback control) in order to understand how to derive an optimal solution of the joint radio-mode management and feedback control problem described by (2.3). Then we increase the complexity to end up with the complete model formulated in the previous chapter. This chapter concludes on simulation results after discussing stability and practical implementation aspects.

3.1 The cost function in the infinite case

In the infinite horizon case the cost function (2.7) is considered without final cost and taking the limit as H goes to infinity, so that:

$$J_{\mu,\eta}(z_0, m_0) = \lim_{H\to\infty} \mathop{\mathsf{E}}_{\substack{\beta_k,\,\omega_k \\ k=0,1,\dots}} \left[\sum_{k=0}^{H-1} \lambda^k \ell_{\nu_k}(z_k, m_k, \hat{u}_k, \beta_k) \right]. \tag{3.1}$$

We are only interested in deriving a solution of the optimal problem (2.8) in the case where this solution leads to a finite cost. Indeed, if the cost function $J_{\mu,\eta}(z,m)$ is always infinite for any policy μ, η and any state (z,m), then any policy is considered as optimal in the sense that any policy leads to $J^* = \min_{\mu,\eta} J_{\mu,\eta}(z,m)$.
The purpose of the discount factor is precisely to force the cost function to be finite for some policies. It weights the importance of immediate actions versus long-term decisions. It is generally not used (*i.e.* taken equal to 1) when the cost function is naturally finite for some policies.

Unfortunately, given the structure of the cost-to-go (2.6), one cannot ensure that such cost function is finite if $\lambda = 1$. Indeed, when $\lambda = 1$, one can distinguish two cases from the basic formulation of the cost-to-go (2.5), that we recall hereafter:

$$\ell_{\nu_k}(x_k, m_k, u_k) = x_k^\top \bar{Q} x_k + u_k^\top \bar{R} u_k + \theta_{m_k,\nu_k}.$$

- The first case considers that an infinite number of transmissions is scheduled over the infinite horizon. The term related to the energy cost of the radio chip, $\theta_{m_k,\nu_k} > 0$, makes the infinite sum infinite, since we have assumed that transmissions have a non-null cost.
- In the second case, the number of transmissions is finite, which implies that there exists a time index k_0 such that, for all $k > k_0$, the system runs open loop since no more transmission update the control input. From that observation, the conditions for the cost function to converge to a finite value are that the system is open-loop stable despite the noise and that one of the radio-modes have a null cost that is scheduled an infinite number of times.

Because we want to be as general as possible, and particularly we want to consider open-loop unstable systems, the discount factor is taken such that $\lambda < 1$ in order to admit policies that make the cost function finite.

However, it has to be noticed that introducing the discount factor $\lambda < 1$ prevents from proving stability with the standard argument in Linear Quadratic (LQ) optimal control, using the cost-to-go as a Lyapunov function. To the authors knowledge, stability proof in discounted problems solved with Dynamic Programming is an open issue. Authors in [Görges et al., 2011, Bertsekas, 2005b] discuss about stability of control laws derived using Dynamic Programming approaches.

Remark 3.1. *We are only searching for a stationary policy (μ^*, η^*), because a time-dependent policy (μ_k, η_k) on an infinite time-horizon would not be implementable. Fortunately, as it is explained in Section 3.3, a stationary policy exists, which is optimal among all policies.*

3.2 Infinite horizon solution - Ideal case

In order to introduce the method that is used to solve the optimisation problem, we first consider the problem under its simplest form, with the following simplifications from the general formulation that has been introduced in the previous chapter.

- The feedback law is static and given by a state feedback: $\hat{u}_k = -Kx_k = -\begin{bmatrix} K & \mathbf{0} \end{bmatrix} z_k$.
- The system is not subject to noise, namely, $w_k = \mathbf{0}$ for all $k \geq 0$.
- The wireless channel is perfect, which implies that the control inputs are never dropped. Under thus assumption, it is natural to consider only one transmitting mode ($N_1 = 1$), as there are not different dropout probabilities.

Under these simplifications, the problem we consider is actually a problem of optimal intermittent control in a simple controlled system. Nevertheless, the problem still captures the need to save energy and brings the novelty of considering several low-consuming modes.

3.2.1 Mathematical model for the closed-loop system

The simplified mathematical model is given hereafter.

System model The controlled system is the same as in the general model, but without additive noise:

$$x_{k+1} = Ax_k + Bu_k.$$

Channel model When a transmission is scheduled, the control input computed by the smart sensor $\hat{u}_k$ is always received at the actuator side, *i.e.* $\beta_k = 1,\ \forall k$.

Switching policy The switching decision is still given by $\nu_k = \eta(x_k, u_{k-1}, m_k)$.

Feedback law It is a static state feedback control law: $\hat{u}_k = -Kx_k = -\begin{bmatrix} K & \mathbf{0} \end{bmatrix} z_k$. It is applied to the system only if a transmission is scheduled by the smart node:

$$u_k = \begin{cases} \hat{u}_k, & \text{in case of transmission,} \\ u_{k-1}, & \text{otherwise.} \end{cases}$$

Radio chip model It is similar to the general model with $N_1 = 1$, *i.e.* considering only one transmitting mode. The transition costs between any of the radio modes is captured by the matrix Θ. The state of the radio is the mode at time k, $m_k \in \mathbb{M}$.

Switched model The switched model is now deterministic. We recall that z_k is the augmented state with the control memory $\tilde{u}_k = u_{k-1}$, $z_k = \begin{bmatrix} x_k^\top & \tilde{u}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x+n_u}$. The switched model is given by:

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ \hat{u}_k = \mu(z_k, m_k), \end{cases}$$

where the function f_{ν_k} is defined as

$$f_{\nu_k}(z_k) = \Phi_{\nu_k} z_k,$$

and the matrices Φ_{ν_k}, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k = 1$, *i.e.* if there is a transmission, then

$$\Phi_{\nu_k} = \Phi_{CL} = \begin{bmatrix} A - BK & \mathbf{0} \\ -K & \mathbf{0} \end{bmatrix},$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$\Phi_{\nu_k} = \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

In the present case, the cost-to-go is given by:

$$\ell_{\nu_k}(z_k, m_k) = z_k^\top Q_{\nu_k} z_k + \theta_{m_k,\nu_k},$$

where the matrices Q_{ν_k}, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k = 1$, *i.e.* if there is a transmission, then

$$Q_{\nu_k} = Q_{CL} = \begin{bmatrix} \bar{Q} + K^\top \bar{R} K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

2. if $v_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$Q_{v_k} = Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix}.$$

Optimal problem

The problem consists in finding a deterministic stationary policy $\eta^*(z_k, m_k)$ such that

$$J^*(z_0, m_0) \triangleq J_{\eta^*}(z_0, m_0) = \min_{\eta} J_\eta(z_0, m_0),$$

where the cost function $J_\eta(z_0, m_0)$ is defined as:

$$J_\eta(z_0, m_0) = \lim_{H\to\infty} \sum_{k=0}^{H-1} \lambda^k \ell_{v_k}(z_k, m_k) = \sum_{k=0}^{\infty} \lambda^k \ell_{v_k}(z_k, m_k),$$
$$\text{where } v_k = \eta(z_k, m_k),\ \forall k \geq 0.$$

3.2.2 The Value Iteration method

As it has been introduced in Section 2.5.3, the Value Iteration method is a tool that consists in running a recursion on the Value Function $V_i(z, m)$ to derive the optimal cost function J^* and an associated optimal policy (μ^*, η^*).

The convergence of the recursion on the Value Function given in Section 2.5.3 is proved in [Bertsekas, 2007, Chap. 3] from the following facts:

- the cost-to-go satisfies the positivity assumption, *i.e.* $\ell_v(z, m) \geq 0$ for all $(z, m) \in \mathbb{X}$ and $v \in \mathbb{M}$,
- the policy $\eta(z, m)$ takes values in the finite set $\mathbb{M}$.

Thus, in the ideal case, the Value Iteration method consists in the following recursion:

$$\begin{aligned} &\textit{after initialising } V_0(z, m) \textit{ to be the all-zero function,} \\ &\textit{compute } V_{i+1}(z, m) = \min_{v\in\mathbb{M}} \left\{\ell_v(z, m) + \lambda V_i(f_v(z), v)\right\}. \end{aligned} \tag{3.2}$$

This recursion converges to the optimal cost function:

$$\lim_{i\to\infty} V_i(z, m) = J^*(z, m).$$

The initialisation of the Value Function at stage 1 to the all-zero function is needed for the convergence of the scheme, as stated in [Bertsekas, 2007, Chap. 3]. Also, the minimum is indeed attained in Equation (3.2).

The computation of the Value Function also provides an optimal radio-mode switching policy $\eta^*(z, m)$, which is given by:

$$\eta^*(z, m) = \arg\min_{v\in\mathbb{M}} \left\{\ell_v(z, m) + \lambda J^*(f_v(z), v)\right\}, \tag{3.3}$$

where the operator "arg min" picks any of the possible solutions if several policies minimise the right hand side of Equation (3.3). Note that the resulting policy is stationary.

3.2.3 Computation of the Value Function iterations

In theory, the Value Iteration Method gives us an iterative algorithm, converging to the solution of our optimisation problem. A caveat is that, at each iteration, we need to compute a function of (z, m), where (z, m) takes values in an uncountable space $\mathbb{X}$.

A first way to implement such iterations in practice, is to partition (a portion of) $\mathbb{X}$ in a grid, then compute the Value Function $V_{i+1}(z, m)$ at the grid points only, by using interpolation to find $V_i(f_v(z), v)$ when $f_v(z)$ is not on the grid. This approach has been taken *e.g.* in [Sundström, O. et al., 2010] and provides a look-up table for $\eta_i(z, m)$ at all grid points.
Computationally, it is very heavy, although this is not a major issue since the long computations are

done offline, while online the switching is chosen by accessing the look-up table. Some minor drawbacks of this approach are that the solution is limited to a finite domain, and that the numerical approximations and interpolation might introduce some errors giving a sub-optimal solution.

A second approach, inspired both by the results in classic LQ optimal control and by the work in [Lincoln, 2003], is to try to compute the functions $V_i(z,m)$ by exploiting some structure that they might have, if any structure exists that is preserved along iterations. For example, in LQ control, for any i the Value Function is a quadratic function of z, *i.e.* $V_i(z) = z^\top \Pi_i z$.
For our problem, the structure is more evolved, as we show hereafter, $V_i(z,m)$ is the minimum on some finite set of quadratic functions of the form $z^\top \Pi z + \pi_m$. Indeed, if $V_0(z,m) \equiv 0,\ \forall (z,m)$, then the iterations (3.2) give Value Functions $V_i(z,m)$ such that

$$V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}, \tag{3.4}$$

where the set $\mathscr{P}_i$ is composed of elements (Π,π), where Π is a symmetric matrix and $\pi = \left[\pi_1, \pi_2, \ldots, \pi_N\right] \in \mathbb{R}^N$ is a vector of non-negative scalars, and π_m represents the m^{th} component of π.

The correctness of the above expressions for $\mathscr{P}_i$ can be proved using mathematical induction, which also gives an explicit recursive construction of the set $\mathscr{P}_i$:

a) Base case:

By definition, $V_0(z,m) = 0$ for all $(z,m) \in \mathbb{X}$. $V_1(z,m)$ can be computed with the iteration algorithm (3.2):

$$\begin{aligned}
V_1(z,m) &= \min_{v\in\mathbb{M}} \left\{ \ell_v(z,m) + \lambda V_0(f_v(z),v) \right\} = \min_{v\in\mathbb{M}} \left\{ \ell_v(z,m) \right\} = \min_{v\in\mathbb{M}} \left\{ z^\top Q_v z + \theta_{m,v} \right\} \\
&= \min \left\{ z^\top Q_{CL} z + \theta_{m,1};\ \min_{v\in\mathbb{M}_2} \left\{ z^\top Q_{OL} z + \theta_{m,v} \right\} \right\} \\
&= \min \left\{ z^\top Q_{CL} z + \theta_{m,1};\ z^\top Q_{OL} z + \min_{v\in\mathbb{M}_2} \left\{ \theta_{m,v} \right\} \right\} \\
&= \min_{(\Pi,\pi)\in\mathscr{P}_1} \left\{ z^\top \Pi z + \pi_m \right\},
\end{aligned}$$

where

$$\mathscr{P}_1 = \left\{ \left(Q_{CL}, \begin{bmatrix} \theta_{1,1} \\ \theta_{2,1} \\ \vdots \\ \theta_{N,1} \end{bmatrix}^\top \right), \left(Q_{OL}, \begin{bmatrix} \min_{v\in\mathbb{M}_2} \left\{ \theta_{1,v} \right\} \\ \min_{v\in\mathbb{M}_2} \left\{ \theta_{2,v} \right\} \\ \vdots \\ \min_{v\in\mathbb{M}_2} \left\{ \theta_{N,v} \right\} \end{bmatrix}^\top \right) \right\} \tag{3.5}$$

Notice that the matrices Q_{CL} and Q_{OL} are indeed symmetric positive semidefinite (as $\bar{Q}$ and $\bar{R}$ are symmetric positive definite).

b) Inductive step:

We assume that the Value Function at i^{th} iteration is given by (3.4):

$$V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}.$$

The Value Function at the next iteration can be computed with the iteration algorithm (3.2): Indeed, assuming that Equation (3.4) holds at iteration i, the Value Function at iteration $i+1$ is given by:

$$\begin{aligned} V_{i+1}(z,m) &= \min_{\nu\in\mathbb{M}} \left\{\ell_\nu(z,m) + \lambda V_i(f_\nu(z),\nu)\right\} \\ &= \min_{\nu\in\mathbb{M}} \left\{ z^\top Q_\nu z + \theta_{m,\nu} + \lambda \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Phi_\nu^\top \Pi \Phi_\nu z + \pi_\nu \right\} \right\} \\ &= \min_{\nu\in\mathbb{M},\,(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \left(\lambda \Phi_\nu^\top \Pi \Phi_\nu + Q_\nu\right) z + \lambda\pi_\nu + \theta_{m,\nu} \right\} \\ &= \min \left\{ \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \left(\lambda \Phi_{CL}^\top \Pi \Phi_{CL} + Q_{CL}\right) z + \lambda\pi_1 + \theta_{m,1} \right\}; \right. \\ &\qquad \left. \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \left(\lambda \Phi_{OL}^\top \Pi \Phi_{OL} + Q_{OL}\right) z + \lambda \min_{\nu\in\mathbb{M}} \left\{ \pi_\nu + \theta_{m,\nu} \right\} \right\} \right\} \\ &= \min_{(\Pi',\pi')\in\mathscr{P}_{i+1}} \left\{ z^\top \Pi' z + \pi'_m \right\}, \end{aligned} \tag{3.6}$$

where

$$\mathscr{P}_{i+1} = \mathscr{P}_{i+1}^{(1)} \cup \mathscr{P}_{i+1}^{(2)} \tag{3.7}$$

$$\mathscr{P}_{i+1}^{(1)} \triangleq \left\{ \left(\lambda \Phi_{CL}^\top \Pi \Phi_{CL} + Q_{CL},\ \begin{bmatrix} \lambda\pi_1 + \theta_{1,1} \\ \lambda\pi_1 + \theta_{2,1} \\ \vdots \\ \lambda\pi_1 + \theta_{N,1} \end{bmatrix}^\top \right) \text{ such that } (\Pi,\pi) \in \mathscr{P}_i \right\}$$

$$\mathscr{P}_{i+1}^{(2)} \triangleq \left\{ \left(\lambda \Phi_{OL}^\top \Pi \Phi_{OL} + Q_{OL},\ \begin{bmatrix} \min_{\nu\in\mathbb{M}_2} \left\{ \lambda\pi_\nu + \theta_{1,\nu} \right\} \\ \min_{\nu\in\mathbb{M}_2} \left\{ \lambda\pi_\nu + \theta_{2,\nu} \right\} \\ \vdots \\ \min_{\nu\in\mathbb{M}_2} \left\{ \lambda\pi_\nu + \theta_{N,\nu} \right\} \end{bmatrix}^\top \right) \text{ such that } (\Pi,\pi) \in \mathscr{P}_i \right\}$$

Notice that the minimisations in (3.6) can be exchanged since they both apply to finite sets.

Remark 3.2. *The Π matrices in the set $\mathscr{P}_i$ are symmetric positive semidefinite by definition. This implies that the Π' matrices in the set $\mathscr{P}_{i+1}$ are also symmetric positive semidefinite (Q_{CL} and Q_{OL} being symmetric positive semidefinite themselves).*

Remark 3.3. *It can be seen from Equation (3.7) that the number of elements in the set $\mathscr{P}_{i+1}$ at iteration $i+1$ is twice the number of elements in the set $\mathscr{P}_i$ at iteration i.*

3.2.4 Optimal switching policy computation

The optimal switching policy is obtained with a two-phases computation process.

Offline computation

The first stage, proceeded offline, consists in computing the Value Function $V_i(z,m)$ given by:

$$V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}.$$

where the computation of $\mathscr{P}_i$ is done recursively. The initialisation is given by Equation (3.5) and the recursion by Equation (3.7). The Value Function converges to the optimal cost function as i goes to infinity. As it is discussed in Section 3.4.3, in the case of practical computations, the recursion is stopped after a sufficient amount of iterations, that can be denoted I, and the Value Function at iteration I is considered as optimal. In the following subsection, we call $\mathscr{P}_\infty$ the set $\mathscr{P}_I$.

Online computation

The second stage is run online. It consists in computing the optimal switching decision at time k, v_k^* as a function of the current state of the system (z_k, m_k), as given by the optimal switching policy:

$v_k^* = \eta^*(z_k, m_k)$. In order to compute v_k^*, one first needs to compute the value of the Value Function for the current system state, or more precisely to determine the couple (Π_k, π_k) that results from the minimisation of the Value Function:

$$(\Pi_k, \pi_k) \triangleq \underset{(\Pi,\pi)\in\mathscr{P}_\infty}{\arg\min} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\}. \tag{3.8}$$

Then the optimal switching decision is computed as follows:

$$v_k^* = \eta^*(z_k, m_k) \triangleq \begin{cases} 1 & \text{if } (\Pi_k, \pi_k) \in \mathscr{P}_\infty^{(1)}, \\ \underset{\nu\in\mathbb{M}_2}{\arg\min} \left\{ \lambda\ \pi_k\big|_\nu + \theta_{m_k,\nu} \right\} & \text{if } (\Pi_k, \pi_k) \in \mathscr{P}_\infty^{(2)}. \end{cases}$$

The notation $\pi_k\big|_\nu$ refers to the ν^{th} element of the vector π_k.

Notice that, by construction, $\mathscr{P}_\infty$ is the union of two subsets, $\mathscr{P}_\infty^{(1)}$ and $\mathscr{P}_\infty^{(2)}$. The definition of η^* is obtained naturally from the recursion (3.7).

3.3 Infinite horizon solution - General case

Now we focus on the general model, as defined in Chapter 2, which takes into account additive Gaussian noise, and an erasure channel. The feedback control law is not constrained to a state feedback and has to be derived jointly with the switching policy from the optimisation problem. The radio-chip has several transmitting modes that affect the probability of successful transmission. The mathematical model is recalled hereafter.

3.3.1 Mathematical model for the closed-loop system

System model — The controlled system is disturbed with a zero-mean white Gaussian noise w_k:

$$x_{k+1} = Ax_k + Bu_k + w_k. \tag{3.9}$$

Channel model — We consider a memoryless erasure channel where the messages u_k are dropped with probabilities $\epsilon(m_k)$ for $m_k \in \mathbb{M}_1$, and otherwise are correctly received.

The dropout probabilities depend on the power used by the radio chip, *i.e.* the transmitting mode. Higher transmission power implies higher success probability, *i.e.* $\epsilon(1) < \epsilon(2) < \ldots < \epsilon(N1)$. We consider a model where the dropout concerns the real-valued message u_k, not single bits or packets. The transmission successes are modeled by Bernoulli random variables β_k, where 1 denotes success and

$$\mathbb{P}\{\beta_k = 0 | m_k = m\} = \epsilon(m).$$

Given the mode m_k, β_k is conditionally independent of the past $\{\beta_h\}_{h<k}$, $\{w_k\}_{h<k}$; the mapping $\epsilon(m)$, $m \in \mathbb{M}_1$, is known for design purposes[1]. We assume that the Acknowledgments (ACKs) from the receiver to the sensor node are reliable.

Switching policy — The switching decision is given by $\nu_k = \eta(x_k, u_{k-1}, m_k)$.

Feedback law — The control input applied to the system depends on the radio-mode switching decision and on the transmission success. If no update is received by the actuator node, it holds the last control input received:

$$u_k = \begin{cases} \beta_k \hat{u}_k + (1-\beta_k) u_{k-1}, & \text{in case of transmission,} \\ u_{k-1}, & \text{otherwise,} \end{cases}$$

where $\hat{u}_k = \mu(x_k, u_{k-1}, m_k)$.

[1] It is reasonable to assume that the $\epsilon(m)$ are known as motivated in [Quevedo et al., 2010] with i.i.d. channel power gains with known distribution.

Radio chip model

We recall that N_1 is the number of transmitting modes while N_2 is the number of low-consuming modes. The transition costs between any pair of radio-modes are captured by the matrix Θ. The state of the radio is the mode at time k, $m_k \in \mathbb{M}$.

Switched model

We recall that z_k is the augmented state, with the control memory $\tilde{u}_k = u_{k-1}$, $z_k = \begin{bmatrix} x_k^\top & \tilde{u}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x+n_u}$ and we define $\omega_k = \begin{bmatrix} w_k^\top & \mathbf{0} \end{bmatrix}^\top$. The switched model is given by:

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ \hat{u}_k = \mu(z_k, m_k), \end{cases}$$

where the function f_{ν_k} is defined as

$$f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) = \Phi_{\nu_k}(\beta_k) z_k + \Gamma_{\nu_k}(\beta_k)\hat{u}_k + \omega_k, \tag{3.10}$$

and the matrices $\Phi_{\nu_k}(\beta_k), \Gamma_{\nu_k}(\beta_k)$, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k \in \mathbb{M}_1$, *i.e.* if there is a transmission, then

$$\Phi_{\nu_k}(\beta_k) = \begin{bmatrix} A & (1-\beta_k)B \\ \mathbf{0} & (1-\beta_k)\mathbf{I} \end{bmatrix} = \begin{cases} \Phi_{CL} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \text{if } \beta_k = 0 \end{cases}$$

$$\Gamma_{\nu_k}(\beta_k) = \beta_k \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} = \begin{cases} \Gamma_{CL} = \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} & \text{if } \beta_k = 1 \\ \Gamma_{OL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 0. \end{cases}$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$\Phi_{\nu_k}(\beta_k) = \Phi_{OL}, \ \forall \beta_k,$$
$$\Gamma_{\nu_k}(\beta_k) = \Gamma_{OL}, \ \forall \beta_k.$$

Finally, the cost-to-go is given by:

$$\ell_{\nu_k}(z_k, m_k, \hat{u}_k, \beta_k) = z_k^\top Q_{\nu_k}(\beta_k) z_k + \hat{u}_k^\top R_{\nu_k}(\beta_k)\hat{u}_k + \theta_{m_k,\nu_k},$$

where the matrices $Q_{\nu_k}(\beta_k)$ and $R_{\nu_k}(\beta_k)$, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k \in \mathbb{M}_1$, *i.e.* if $u_k = \beta_k \hat{u}_k + (1-\beta_k)\tilde{u}_k$, then

$$Q_{\nu_k}(\beta_k) = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & (1-\beta_k)\bar{R} \end{bmatrix} = \begin{cases} Q_{CL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} & \text{if } \beta_k = 0, \end{cases}$$

$$R_{\nu_k}(\beta_k) = \beta_k \bar{R} = \begin{cases} R_{CL} = \bar{R} & \text{if } \beta_k = 1 \\ R_{OL} = \mathbf{0} & \text{if } \beta_k = 0. \end{cases}$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if $u_k = \tilde{u}_k$, then

$$Q_{\nu_k}(\beta_k) = Q_{OL}, \ \forall \beta_k,$$
$$R_{\nu_k}(\beta_k) = R_{OL}, \ \forall \beta_k.$$

Optimal problem The problem consists in finding a joint stationary policy $(\mu^*(z_k, m_k), \eta^*(z_k, m_k))$ such that

$$J^*(z_0, m_0) \triangleq J_{\mu^*,\eta^*}(z_0, m_0) = \min_{\mu,\eta} J_{\mu,\eta}(z_0, m_0),$$

where the cost function $J_{\mu,\eta}(z_0, m_0)$ is defined as:

$$J_{\mu,\eta}(z_0, m_0) = \lim_{H\to\infty} \mathop{\mathsf{E}}_{\substack{\beta_k,\omega_k \\ k=0,1,\dots}} \left[\sum_{k=0}^{H-1} \lambda^k \ell_{v_k}(z_k, m_k, \hat{u}_k, \beta_k)\right]. \tag{3.11}$$

3.3.2 The Value Iteration method

The method that we propose to solve the optimisation problem is similar to the ideal case. The Value Iteration method provides a recursion to compute the optimal cost function $J^*(z_0, m_0)$, and the associated joint switching policy and feedback control law (μ^*, η^*), which take values in $\mathbb{U}(z)$, as defined in Definition 2.1.

The main differences with the ideal case are that the general case considers stochastic variables and that the optimal policy does not take values in a finite set anymore.

In the general model case, the Bellman's Equation is given by:

$$J^*(z, m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z, m, \hat{u}, \beta) + \lambda J^*(f_v(z, \hat{u}, \beta, \omega), v)\right] \right\}, \ \forall (z, m) \in \mathbb{X}.$$

From this equation (proven to hold in Section 3.3.3), one can derive a Value Iteration recursion similarly to the ideal case. This recursion is given in Theorem 3.1 which states that the recursion provides the optimal policy as it converges. Corollary 3.1 states that the Value Iteration method converges to the optimal cost function. Theorem 3.1 and Corollary 3.1, given hereafter, are proven in Section 3.3.3.

Theorem 3.1. *Given the initial state (z_0, m_0), an optimal feedback control law μ^* and an optimal radio-mode switching policy η^* of the switched system (3.10), such that $J^*(z_0, m_0) \triangleq J_{\mu^*,\eta^*}(z_0, m_0) = \min_{\mu,\eta} J_{\mu,\eta}(z_0, m_0)$ for the cost function (3.11), can be found as limit points of the policies $\{\mu_i, \eta_i\}$ computed with the Value Iteration method. This method consists in defining a function V_i, such that $\forall (z, m) \in \mathbb{X}$:*

$$\begin{aligned}
&V_0(z, m) = 0, \\
&V_{i+1}(z, m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z, m, \hat{u}, \beta) + \lambda V_i(f_v(z, \hat{u}, \beta, \omega), v)\right] \right\}, \\
&(\mu_{i+1}(z, m), \eta_{i+1}(z, m)) \triangleq \arg\min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z, m, \hat{u}, \beta) + \lambda V_i(f_v(z, \hat{u}, \beta, \omega), v)\right] \right\}.
\end{aligned}$$

Corollary 3.1. *Given the initial state (z_0, m_0), the optimal Value Function V^* for the switched system (3.10), such that $J^*(z_0, m_0) \triangleq J_{\mu^*,\eta^*}(z_0, m_0) = \min_{\mu,\eta} J_{\mu,\eta}(z_0, m_0)$ for the cost function (3.11), is given by:*

$$V^*(z, m) = \lim_{i\to\infty} V_i(z, m), \ \forall (z, m) \in \mathbb{X},$$

where V_i is computed with the Value Iteration method, as described by Theorem 3.1, and it holds that:

$$J^*(z, m) = V^*(z, m), \ \forall (z, m) \in \mathbb{X}.$$

3.3.3 Proof of the convergence of the Value Iteration method

This subsection proves Theorem 3.1 and Corollary 3.1. The proof is based on [Bertsekas, 2007, Chap. 3]. However, the problem that is considered in this book is slightly different from ours. Indeed, our model state is composed of a vector part z_k and a discrete part m_k taking values in a finite set whereas the state space in [Bertsekas, 2007, Chap. 3] is only $\mathbb{R}^n$ for any n. Moreover, in addition to the system noise ω, we consider another stochastic variable β (describing the message dropout) which is not considered in [Bertsekas, 2007, Chap. 3]. Finally, we have two control variables, the switching decision v and the feedback control input u, whereas only a control input is considered in [Bertsekas,

2007, Chap. 3].
For these reasons, we need to prove that the Value Iteration method given in [Bertsekas, 2007, Chap. 3] and reformulated in Theorem 3.1 actually provides the optimal cost function and joint policy. The reasoning that is used to prove our main result is similar to the one in [Bertsekas, 2007, Chap. 3], and in this section, we refer to the propositions and corollaries from the book. In spite of the differences between our problem and the one considered in this book, the proofs of certain propositions that are proposed in the book can be extended to our case without any difficulty, and only the proofs that need details are given here.
Another comment has to be made about the nature of the noise ω_k in Equation (3.9). It is considered as a vector of real values in this thesis whereas it takes values in a countable space in the book [Bertsekas, 2007], on which the proof is based. It can be seen in the book [Bertsekas and Shreve, 1978, Chap. 9] that the same tools can be used in the case of noise taking values in a real space.

After proving that the cost function is well defined, we show that the optimal solution is the smallest fixed point of the Bellman's equation, and that the Value Iteration method converges to a fixed point which is the smallest one, therefore that the Value Iteration method converges to the optimal value.

We introduce the shorthand notation:

$$(TV)(z,m) = \min_{(\hat{u},\nu)\in\mathbb{U}(z)} \mathsf{E}_{\beta,\omega}\left[\ell_\nu(z,m,\hat{u},\beta) + \lambda V(f_\nu(z,\hat{u},\beta,\omega),\nu)\right],$$
$$(T_{\mu,\eta}V)(z,m) = \mathsf{E}_{\beta,\omega}\left[\ell_{\eta(z,m)}(z,m,\mu(z,m),\beta) + \lambda V(f_{\eta(z,m)}(z,\mu(z,m),\beta,\omega),\eta(z,m))\right].$$

To ease the notation, we can write $(T_{u,\nu}V)(z,m)$ where u and ν are values rather than laws.
For the operators T and $T_{\mu,\eta}$, we will denote by the "power" notation T^i and $T^i_{\mu,\eta}$ the composition, *e.g.* $(T^2V)(z,m) = (T(TV))(z,m)$ and $(T^0V)(z,m) = V(z,m)$.
We recall that a fixed point for the operator T is a function $F(z,m)$ such that $F = TF$.
We define V_0 as the zero function on $\mathbb{X}$,

$$V_0(z,m) = 0,\ \forall (z,m)\in\mathbb{X}.$$

Our problem satisfies the positivity assumption, as defined in [Bertsekas, 2007, Ch. 3.1] by:

Assumption 3.1. *Positivity: the cost per stage ℓ_ν satisfies*

$$0 \le \ell_\nu(z,m,\hat{u},\beta), \text{ for all } (z,m)\in\mathbb{X},\ \hat{u}\in\mathbb{R}^{n_u},\ \nu\in\mathbb{M} \text{ and } \beta\in\{0,1\}.$$

Remark 3.4. *The positivity assumption implies the monotonicity of T and $T_{\mu,\eta}$:*

$$V(z,m) \le V'(z,m) \Rightarrow TV(z,m) \le TV'(z,m) \text{ and } T_{\mu,\eta}V(z,m) \le T_{\mu,\eta}V'(z,m). \tag{3.12}$$

Moreover, we notice that:

$$V_0(z,m) \le TV_0(z,m) \le \ldots \le T^iV_0(z,m),\ \forall\ i > 1. \tag{3.13}$$

Remark 3.5. *Given $V(z,m)$ a continuous function of z, for any fixed $\nu\in\mathbb{M}_1$,*
then $\mathsf{E}_{\beta,\omega}\left[\ell_\nu(z,m,\hat{u},\beta) + \lambda V(f_\nu(z,\hat{u},\beta,\omega),\nu)\right]$ is a continuous function of $\hat{u}$, which tends to $+\infty$ for $\|\hat{u}\|\to\infty$.

Moreover, given $V_0(z,m) = 0,\ \forall\ (z,m)$, it holds that, for all $k\ge 0$, $T^kV_0(z,m)$ is a continuous function of z.

We define the limit function V_∞ as:

$$V_\infty(z,m) = \lim_{i\to\infty}(T^iV_0)(z,m).$$

In the last equation, the limit is actually a limit (and not a limit superior) thanks to the monotonicity of T, see Equation (3.12). We notice that V_∞ can take the value ∞.

The proof follows the 5 steps:

- J^* is a fixed point of T,

- J^* is the smallest fixed point of T,
- V_∞ is a fixed point of T,
- $V_\infty = J^*$,
- the Value Iteration method also provides an optimal policy.

As a first step, we recall the Bellman's Equation, which states that J^* is a fixed point of T:

Proposition 3.1 (Proposition 3.1.1 from [Bertsekas, 2007]). *The optimal cost function J^* is a fixed point for T,* i.e. *J^* satisfies:*

$$J^*(z,m) = \min_{(\hat{u},\nu)\in\mathbb{U}(z)} \mathop{\mathsf{E}}_{\beta,\omega}\left[\ell_\nu(z,m,\hat{u},\beta) + \lambda J^*(f_\nu(z,\hat{u},\beta,\omega),\nu)\right],\ \forall (z,m)\in\mathbb{X}$$

or, equivalently,

$$J^* = TJ^*.$$

Proof. See proof of Proposition 3.1.1 in [Bertsekas, 2007]. □

In the second step, we establish that J^* is the smallest fixed point.

Proposition 3.2 (Proposition 3.1.2 from [Bertsekas, 2007]). *If $\tilde{J}:\mathbb{X}\to(-\infty,\infty]$ satisfies $\tilde{J}\geq T\tilde{J}$ and $\tilde{J}\geq 0$, then $\tilde{J}\geq J^*$.*

Proof. See proof of Proposition 3.1.2 in [Bertsekas, 2007]. □

This proposition leads to the following corollary:

Corollary 3.2 (Corollary 3.1.2.1 from [Bertsekas, 2007]). *If $\tilde{J}:\mathbb{X}\to(-\infty,\infty]$ satisfies $\tilde{J}\geq T_{\mu,\eta}\tilde{J}$ and $\tilde{J}\geq 0$, then $\tilde{J}\geq J_{\mu,\eta}$.*

In the third step, we show that V_∞ is a fixed point of T, *i.e.* that V_∞ satisfies $V_\infty = TV_\infty$.

Proposition 3.3. *V_∞ is a fixed point of T,* i.e. *:*

$$V_\infty = TV_\infty.$$

Moreover, letting $(\bar{\mu},\bar{\eta})$ be the stationary policy corresponding to any limit point of the optimal policies along the Value Iterations, then

$$V_\infty = T_{\bar{\mu},\bar{\eta}}V_\infty.$$

Proof. This proof is inspired from the poofs of [Proposition 3.1.6 from [Bertsekas, 2007]] and [Proposition 3.1.7 from [Bertsekas, 2007]], adapted to our setting.
The positivity assumption 3.1 yields:

$$\begin{aligned} &V_0 \leq TV_0 \leq T^2V_0 \leq \ldots \leq T^iV_0 \leq \ldots \leq V_\infty \\ &\Rightarrow T^iV_0 \leq V_\infty, \\ &\Rightarrow T^{i+1}V_0 \leq TV_\infty. \end{aligned} \tag{3.14}$$

The last implication results from the monotonicity of T (see Remark 3.4). Taking Equation (3.14) when i goes to infinity, we obtain:

$$V_\infty \leq TV_\infty.$$

Now we will prove that $V_\infty \geq TV_\infty$.
Consider any $(z,m)\in\mathbb{X}$. If $V_\infty(z,m)=\infty$, then trivially $V_\infty(z,m)\geq TV_\infty(z,m)$.
If $V_\infty(z,m)<\infty$, then consider for all $i\geq 0$:

$$(\hat{u}_i,\nu_i) = \operatorname*{arg\,min}_{(\hat{u},\nu)\in\mathbb{U}(\bar{z})} \mathop{\mathsf{E}}_{\beta,\omega}\left[\ell_\nu(\bar{z},\bar{m},\hat{u},\beta) + \lambda T^iV_0(f_\nu(\bar{z},\hat{u},\beta,\omega),\nu)\right].$$

Notice that this is one of the possibly multiple minimisers and then its existence is ensured by Remark 3.5.
Now we want to prove that:

- there exists a subsequence of $\{(\hat{u}_i, \nu_i)\}$ which converges,
- and that for all limit point $(\hat{\bar{u}}, \bar{\nu})$ of $\{(\hat{u}_i, \nu_i)\}$, $T_{\hat{\bar{u}},\bar{\nu}} V_\infty(z,m) \leq V_\infty(z,m)$.

Notice that ν_i is eventually constant on any convergent subsequence, and there exists subsequences with constant ν_i.

For all $j \geq 0$ such that $j \leq i$, from Equation (3.13), it holds that:

$$(T^j V_0)(z,m) \leq (T^i V_0)(z,m).$$

The monotonicity of $T_{\hat{u}_i,\nu_i}$ yields:

$$(T_{\hat{u}_i,\nu_i}(T^j V_0))(z,m) \leq (T_{\hat{u}_i,\nu_i}(T^i V_0))(z,m).$$

We notice that:

$$(T_{\hat{u}_i,\nu_i}(T^i V_0))(z,m) = (T^{i+1} V_0)(z,m),$$

and that:

$$(T^{i+1} V_0)(z,m) \leq V_\infty(z,m),$$

according to Equation (3.13). This implies the following:

$$(T_{\hat{u}_i,\nu_i}(T^j V_0))(z,m) \leq V_\infty(z,m).$$

Look at any subsequence with $\nu_i = \bar{\nu}$, take $j = 0$ and notice that

$$T_{\hat{u}_i,\bar{\nu}} V_0(z,m) \leq V_\infty(z,m)$$

implies that there exists a subsequence with $i \in \mathbb{K}$, where $\mathbb{K}$ is an infinite subset of integers, such that

$$\lim_{i\to\infty, i\in\mathbb{K}} \hat{u}_i = \bar{\hat{u}}.$$

Indeed, the set $\{u : T_{u,\bar{\nu}} V_0(z,m) \leq V_\infty(z,m)\}$ is compact, and thus such $\bar{\hat{u}}$ exists.

Take any limit point $(\bar{\hat{u}}, \nu)$, take any subsequence $\{(\hat{u}_i, \bar{\nu})\}_{i\in\mathbb{K}}$ converging to $(\bar{\hat{u}}, \nu)$, for all j, for all $i \in \mathbb{K}$, $i \geq j$, then:

$$T_{\hat{u}_i,\bar{\nu}} T^j V_0(z,m) \leq V_\infty(z,m). \tag{3.15}$$

Taking the limit of Equation (3.15) as i goes to infinity, $i \in \mathbb{K}$, $i \geq j$, then:

$$\forall j : \; T_{\bar{\hat{u}},\bar{\nu}} T^j V_0(z,m) \leq V_\infty(z,m).$$

Now taking the limit as j goes to infinity, it holds that:

$$T_{\bar{\hat{u}},\bar{\nu}} V_\infty(z,m) \leq V_\infty(z,m).$$

By noticing that the optimal policy leads to the smallest Value Function, *i.e.* that

$$T V_\infty(z,m) \leq T_{\bar{\hat{u}},\bar{\nu}} V_\infty(z,m),$$

then we have proven that:

$$\begin{aligned} & T V_\infty(z,m) \leq T_{\bar{\hat{u}},\bar{\nu}} V_\infty(z,m) \leq V_\infty(z,m) \\ \Rightarrow & T V_\infty(z,m) \leq V_\infty(z,m) \end{aligned} \tag{3.16}$$

which implies that $T V_\infty(z,m) = V_\infty(z,m)$ since we have already proven that $T V_\infty(z,m) \geq V_\infty(z,m)$.

In addition, we can also conclude the following facts:

- We have constructed a stationary policy $(\bar{\mu}, \bar{\eta})$:

$$(\bar{\mu}(z,m), \bar{\eta}(z,m)) = \begin{cases} (\bar{\hat{u}}, \bar{\nu}) \text{ as defined above when } V_\infty < \infty, \\ \text{anything when } V_\infty = \infty. \end{cases}$$

- When $V_\infty(z,m) < \infty$, inequality (3.16) with the fact that $TV_\infty(z,m) = V_\infty(z,m)$ leads to:

$$\begin{aligned} V_\infty(z,m) = TV_\infty(z,m) \leq T_{\hat{\mu},\bar{\nu}} V_\infty(z,m) \leq V_\infty(z,m) \\ \Rightarrow T_{\hat{\mu},\bar{\nu}} V_\infty(z,m) = V_\infty(z,m) \\ \Leftrightarrow T_{\bar{\mu},\bar{\eta}} V_\infty(z,m) = V_\infty(z,m) \end{aligned}$$

- If $V_\infty(z,m) = \infty$ then we have trivially:

$$T_{\bar{\mu},\bar{\eta}} V_\infty(z,m) \leq V_\infty(z,m).$$

□

The fourth step consists in proving that V_∞ is indeed the optimal cost function. We know that V_∞ is a fixed point, we need to prove that it is the smallest one, to conclude that $V_\infty = J^*$.

Proposition 3.4 (Proposition 3.1.5 from [Bertsekas, 2007])**.** *If $V_0 \leq V \leq J^*$ and:*

$$V_\infty(z,m) = (TV_\infty)(z,m),\ \forall (z,m) \in \mathbb{X},$$

then:

$$\lim_{i\to\infty} (T^i V)(z,m) = J^*(z,m),\ \forall (z,m) \in \mathbb{X}.$$

Proof. See proof of Proposition 3.1.5 in [Bertsekas, 2007], $V_\infty = J^*$ follows taking $V(z,m) = V_0(z,m) = 0$ for all $(z,m) \in \mathbb{X}$. □

In the last step, we prove that the stationary policy derived in the proof of Proposition 3.3 is an optimal policy.

Proposition 3.5. *The policy $(\bar{\mu},\bar{\eta})$ derived in Proposition 3.3 is optimal,* i.e. *:*

$$J_{\bar{\mu},\bar{\eta}}(z,m) = J^*(z,m),\ \forall (z,m) \in \mathbb{X}.$$

Proof. From Proposition 3.3, and noticing that we have proven that $V_\infty = J^*$, it holds that:

$$J^* \geq T_{\bar{\mu},\bar{\eta}} J^*.$$

This results allows to apply Corollary 3.2 with $\tilde{J} = J^*$, which leads to:

$$J^* \geq J_{\bar{\mu},\bar{\eta}}.$$

Obviously, the optimal cost function is the smallest possible cost, *i.e.* $J^* \leq J_{\bar{\mu},\bar{\eta}}$. This implies that $J_{\bar{\mu},\bar{\eta}} = J^*$. □

3.3.4 Computation of the Value Function iterations

As it has been discussed in Section 3.2.3, the computation of the iterations of the Value Function generally relies on interpolation on grid points. In the ideal case, the iterations on the Value Function can be computed by iterating on the sets $\mathscr{P}_i$, as detailed in Section 3.2.3.

In the general case, such relationship between the Value Function and the sets $\mathscr{P}_i$ cannot be exploited anymore. Indeed, the use of stochastic variable for the system noise forces to use expected values. The expected values make the computation intractable. In other words, we cannot prove that the Value Function in the general case can be written on the form $V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}$.

3.3.5 Optimal joint switching policy and feedback control computation

The optimal joint policy is derived in a two-phases computation process, as it is described in Section 3.2.4. It is worth noticing that in the nominal case (without noise), the iteration process given by Equation (3.8) provides a simple formulation of the switching policy and the feedback law. We denote $\mathscr{P}_\infty$ the set $\mathscr{P}_i$ computed offline for i large enough to assume convergence of the iterative scheme, as

explained in Section 3.2.4. Then, online, we denote (Π_k, π_k) the result from the minimisation of the Value Function:

$$(\Pi_k, \pi_k) \triangleq \underset{(\Pi,\pi)\in\mathscr{P}_\infty}{\arg\min} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\}.$$

Then the optimal switching decision is computed as follows:

$$\nu_k^* = \eta^*(z_k, m_k) \triangleq \underset{\nu\in\mathbb{M}_j}{\arg\min} \left\{ \lambda\ \pi_k\big|_\nu + \theta_{m_k,\nu} \right\},$$

where j is such that $(\Pi_k, \pi_k) \in \mathscr{P}_\infty^{(j)}$. We recall that $\pi_k\big|_\nu$ refers to the ν^{th} element of the vector π_k.

If a transmission is scheduled at time k, *i.e.* $\nu_k^* \in \mathbb{M}_1$, then the optimal feedback control is given by:

$$u_k^* = \mu^*(z_k, m_k) \triangleq -\kappa_{\Pi_k} z_k = -\lambda(\lambda\Gamma_{CL}^T \Pi_k \Gamma_{CL} + R_{CL})^{-1} \Gamma_{CL}^T \Pi_k \Phi_{CL} z_k.$$

3.4 Various Discussions

3.4.1 Discussion about stability

The role of the discount factor λ

As it has been motivated in Section 3.1, the discount factor λ in the cost function (3.1) is necessary to obtain an optimal joint switching policy and feedback control law, in the case of open-loop unstable plant. Indeed, we explained in Section 3.1 that the cost function $J_{\mu,\eta}(z_0, m_0)$ (2.7) is always infinite in the case of open-loop unstable system. Then any policy (μ, η) makes the cost function infinite, which is the minimum that can be attained, and can be claimed as optimal. Thus the discount factor allows the problem to be well-posed.
In the case of open-loop stable plant, the discount factor may be dropped under the condition that the most economic low-transmitting mode has a null cost. We can expect that the optimal stationary policy consists in driving the system below a given threshold by closing the loop (*i.e.* by transmitting, and thus consuming energy) and then letting the system converge to the origin in open loop. We have not studied this case in the current work.

It is important to understand the need of the discount factor in the cost function because it prevents from obtaining any stability result. Indeed, the stability is induced by the use of Dynamic Programming in the case where $\lambda = 1$, where the Value Function is naturally a Lyapunov candidate to prove stability. However the introduction of $\lambda < 1$ in the cost function discards the Value Function as a Lyapunov candidate because the discount factor can "compensate" for the instability of the plant. In other words, some switching policies which provide an unstable closed-loop behaviour may also have a finite total cost. However the policies which result in an unstable closed-loop are more likely to have a total cost larger than the policies giving rise to a stable closed-loop, and especially than the optimal policy. But, to the author's knowledge, there exists no tool to prove that result. We insist on the fact that proving the stability of closed-loop system where the control law is optimal in the sense of a cost function considering a discount factor over an infinite horizon is an open problem, and it is beyond the scope of this thesis. We can notice that the notion of stability of control law derived using Dynamic Programming based approaches is not trivial, as it is discussed in [Görges et al., 2011, Bertsekas, 2005b].

A posteriori stability

It is worth mentioning that the commonly used definition for stability is to asymptotically drive the state of the system to the origin as the time goes to infinity. This kind of stability cannot be achieved in our setup when the plant is open-loop unstable. Indeed our scheme bases its energy saving on the action of turning off the radio chip when the transmission cost dominates the cost associated with the control performance. This makes the system run open loop and deviate from the origin. The policy schedules a transmission when the cost associated with the control performance dominates the transmission cost. In our simulations, the system naturally oscillates around the origin without leaving a ball around the origin. Considering this definition, one could expect to check *a posteriori* that the obtained switching policy is actually stable, as done in [Görges et al., 2011]. We propose to have a look at an a posteriori stability simulation in Section 3.5.3.

3.4.2 Computation complexity

In this section we discuss the complexity of the computation of the proposed solutions. We do not provide a comprehensive study of the complexity here, we just give some hints about how the computational load evolves along the iterations in the different cases we consider in this chapter.

General case

As we already explained, the implementation of the joint switching policy and feedback control law as given in Theorem 3.1 is using a grid on a portion of the state space $\mathbb{X}$ and control space $\mathbb{U}(z)$. This computation, done offline, consists in computing the Value Function $V_i(z,m)$ up to an iteration i large enough to observe convergence. The convergence is discussed in the next section. The complexity of such computation mainly depends on the size and the precision of the grid. The computational load is independent of the iteration index, *i.e.* $V_i(z,m)$ and $V_j(z,m)$, with $i \neq j$, requires the same computation load.

Ideal case

As presented in Section 3.2.3, the Value Function in the ideal case can be computed exactly by iterating on the sets $\mathscr{P}_i$. The advantage of this approach upon the discretisation scheme (which is using a grid) is that it provides the Value Function for any point in the state space while the discretisation scheme limits the results to the points on the grid. However, the complexity of the computation increases exponentially across the iterations. Indeed, as it can be seen from Equation (3.7), the number of elements in the set $\mathscr{P}_{i+1}$ at iteration $i+1$ is twice the number of elements in the set $\mathscr{P}_i$ at iteration i. This is a drawback compared to the discretisation scheme whose complexity remains constant along the iterations. However, it can be noticed that the number of possibilities when choosing a radio-mode between N at an iteration i is N^i. By construction, our approach reduces this amount of possibilities to 2^i.

But to limit further the computation burden, we propose a scheme to discard some elements in $\mathscr{P}_i$ which are not useful. Indeed, assuming we are testing the candidate $(\Pi^{\text{cand}}, \pi^{\text{cand}}) \in \mathscr{P}_i$, if there exists $(\Pi, \pi) \in \mathscr{P}_i$ such that:

$$z^\top \Pi^{\text{cand}} z + \pi_m^{\text{cand}} > z^\top \Pi z + \pi_m \qquad \forall (z,m) \in \mathbb{X}$$

then the candidate $(\Pi^{\text{cand}}, \pi^{\text{cand}})$ can be removed from $\mathscr{P}_i$. This is illustrated on a simplified scalar example in Figure 3.1

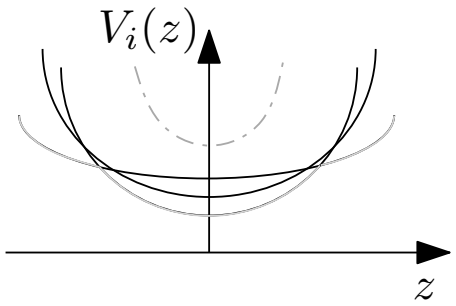


Figure 3.1 *– Illustration of the discard scheme to reduce the amount of elements in the set $\mathscr{P}_i$, the set here is composed of four elements, the Value Function is given by the minimum on these elements (highlighted in light gray), and the element represented as a dashed dark gray line does not contribute to the minimum, thus it can be discarded from the set.*

The comments on the ideal case also apply to the nominal case, *i.e.* the computation of V_i when the noise is considered always equal to zero.

3.4.3 Practical convergence of the Value Iteration method and stopping criterion

The Value Function iteration method consists in running a recursion an infinite number of times in order to obtain the optimal policy. Obviously, the recursion is not run an infinite number of times in

practice, and this raises the question of defining a stopping criterion. This section discusses this point for the cases seen in this chapter.

General case

The general case is implemented using the discretisation scheme (with a grid). We define $\bar{\mathbb{X}}$ as a grid on a portion of $\mathbb{X}$. The stopping criterion is defined by introducing a scalar ξ whose the choice depends on the precision needed.

Stopping criterion

If $\|V_{i+1}(z,m) - V_i(z,m)\| < \xi,\ \forall (z,m) \in \bar{\mathbb{X}}$,
then we conclude that the Value Function is close enought to the optimal value.

Unfortunately, such criterion does not ensure that the distance to the optimal policy (μ^*, η^*) is below some given thresholds.

Ideal case

In the ideal case, we can implement the recursion on the sets $\mathscr{P}_i$ instead of the discretisation scheme (which is using a grid). In this case, having the set $\mathscr{P}_i$ converging to a fixed set would provide a stopping criterion. However, as it is explained in Section 3.4.2, the number of elements in this set increases exponentially. Moreover, there is no subset of $\mathscr{P}_i$ which may converge along the iterations. Even with the discarding scheme proposed in the previous section, the set $\mathscr{P}_i$ still increases. Then, we have to run the recursion for a sufficient amount of iterations. The same approach as the general case may be used as a stopping criterion.

3.5 Simulations

To illustrate the proposed method, we present here an example of a first order unstable system, with the following parameters:

$$
\begin{aligned}
&x_{k+1} = 1.074 x_k - 1.4808 u_k + w_k; \quad T_s = 0.05\text{s};\\
&\bar{W} = 0.02; \quad K = -0.23; \quad \bar{Q} = 0.01; \quad \bar{R} = 0.1;\\
&\Theta = \begin{bmatrix} 2.85 & 1.8 & 1.9 \\ 3.2 & 1.4 & 6{\cdot}10^{-5} \\ 3.5 & 3.7{\cdot}10^{-3} & 6{\cdot}10^{-5} \end{bmatrix}; \quad \begin{matrix} N = 3 \\ \lambda = 0.8 \\ \epsilon = 0.3 \end{matrix}
\end{aligned}
$$

Three radio-modes are considered (`Tx`, `Idle`, `Sleep`), the values of the transition costs are given in the Θ matrix in [mJ] and are computed from the datasheet of the radio chip Texas Instrument CC1100 [Chipcon Products].

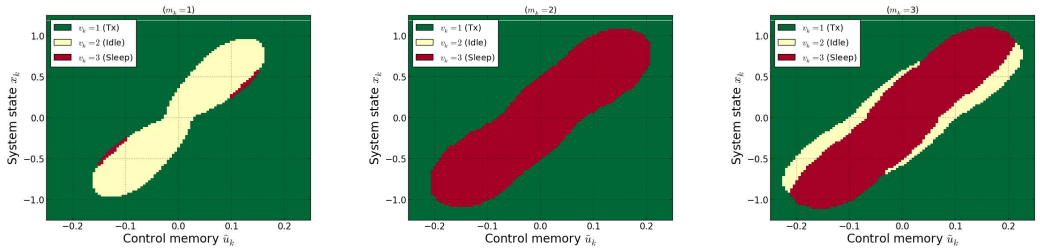
3.5.1 Offline results

The offline computation provides the switching policy $\nu = \eta^*(x, \tilde{u}, m)$. This means that, at any time k, η^* gives the optimal switching decision $\nu^* \in \mathbb{M}$ for the current measurement x_k, the last control input applied to the system $u_{k-1} = \tilde{u_k}$ and the current radio-mode m_k. This is depicted in Figure 3.2 with a sub-figure per each current mode m_k.

We observe that the regions where the radio is switched to low consuming modes (colours red and yellow in Figure 3.2) are finite sets around the equilibrium point $(x_k, \tilde{u}_k) = (0,0)$, and follow the direction $\tilde{u}_k = -Kx_k$ ($K < 0$ in our example). Outside of these regions, a transmission is forced. This means that when the last control input applied to the system is close to what the state-feedback law would have decided, then the switching policy does not send an update.

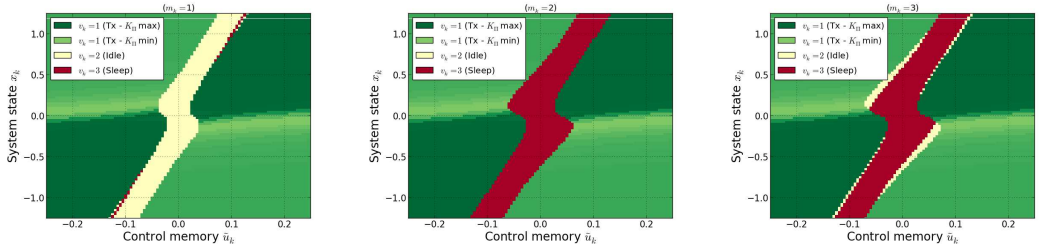
Note that we obtain an event-based radio-mode switching policy. Indeed, a switching occurs only when the state of the system crosses one of the regions in Figures 3.2-3.3.

In the case where we do not force the feedback gain to be a static value, the optimal state feedback gain can be computed from the optimisation problem as the optimal switching policy is. In this case, as a result of the offline computation one also get the value of the optimal feedback gain. In Figure 3.3,



(a) Current mode is `Tx` ($m_k = 1$) (b) Current mode is `Idle` ($m_k = 2$) (c) Current mode is `Sleep` ($m_k = 3$)

Figure 3.2 *– Optimal switching policy derived from the offline computation, green ⟺ switch to* `Tx` *($v_k = 1$), yellow ⟺ switch to* `Idle` *($v_k = 2$) and red ⟺ switch to* `Sleep` *($v_k = 3$), the control memory $\tilde{u}_k = u_{k-1}$ is on the x-axis and the system output x_k on the y-axis.*



(a) Current mode is `Tx` ($m_k = 1$) (b) Current mode is `Idle` ($m_k = 2$) (c) Current mode is `Sleep` ($m_k = 3$)

Figure 3.3 *– Optimal switching policy derived from the offline computation, green ⟺ switch to* `Tx` *($v_k = 1$), yellow ⟺ switch to* `Idle` *($v_k = 2$) and red ⟺ switch to* `Sleep` *($v_k = 3$), the variation of green indicates different feedback gain values, the control memory $\tilde{u}_k = u_{k-1}$ is on the x-axis and the system output x_k on the y-axis.*

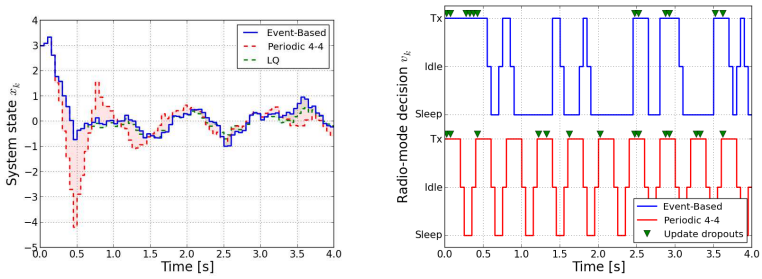
these different gains are represented as variation of green, from light green for the gain with the smallest absolute value to dark green for the gain with the highest absolute value. We observe in this case zones in the state space where a given gain is optimal over the others. The direction $\tilde{u}_k = -Kx_k$ is still observable.

3.5.2 Online results

After deriving the switching policy, we run online temporal simulations to observe the behaviour of the system, using the static feedback gain case. In Figure 3.4 we compare our event-based switching policy with the LQ case and some periodical policy using the same state feedback law $u=-Kx$, where the radio is alternatively switched to `Tx` and low consuming modes. In the periodic case, we consider various periodic patterns for the radio mode: we will denote by periodic i-j a sequence with period $i+j$ where the mode is `Tx` for i consecutive sampling intervals and then is `Sleep` for j intervals. We will then denote by periodic i-j min a sequence with the same period and the same `Tx` intervals (so that the control performance is unchanged), but where the mode for the non-transmitting intervals is chosen in $\mathbb{M}_2$ so as to minimise the energy consumption. The online simulations include channel dropouts and additive output noise on the system which are the same to compare fairly the different policies.

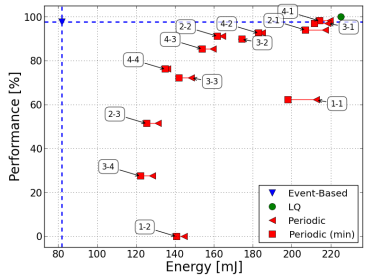
Figure 3.4(a) shows that the system is stabilised in a set around the equilibrium in all the cases. In Figure 3.4(b), one can see the switching decisions for our event-based policy and the periodic 4-4 policy. The green triangles indicate time instants where a transmission was intended but a dropout occurred. Only the periodic 4-4 policy is showed in Figures 3.4(a) and 3.4(b) to note overload the figures. The periodic 4-4 policy has been chosen because it is the one that gives the better trade-off between performance and energy consumption among all the periodic policies. The LQ case is not plotted in Figure 3.4(b) because it only consists of always transmitting.

In Figure 3.4(b), when an update is dropped, the Event-Based Control (EBC) takes again the decision to transmit a new control update, while the periodic scheme is not taking dropouts into account. There is no explicit consideration of dropout in the EBC scheme, but the decision to transmit is taken again until the state returns to a region where the transmission cost is more important than the performance cost. Moreover, the EBC scheme may hold the `Sleep` mode for a long time interval when a



(a) State of the system x_k.

(b) Switching decision v_k.



(c) Performance vs. energy consumption E_{Tmax} for several periodic schemes.

Figure 3.4 *– Online simulations comparing our event-based switching policy with periodic ones. Additive zero-mean white Gaussian noise and channel dropouts are considered. The green triangles in Fig (b) indicate time instants where a transmission was intended but a dropout occurred.*

transmission costs more than the deviation observed on the state.

Finally Figure 3.4(c) compares the EBC scheme to several periodic patterns in terms of performance and energy. In this figure, the performance of the closed-loop system is computed from the cost function $(\sum_{i=0}^{\text{Tmax}}(x_i^\top \bar{Q} x_i + u_i^\top \bar{R} u_i)$ where Tmax=4s) and normalised such that 100% and 0% are the best and the worst performance, respectively. To obtain this figure, 100 runs with different random noise and dropouts generations have been averaged for each policy. As expected, we can see that the LQ policy achieves the best performance, but the worst energy consumption. Indeed, this policy consists in always transmitting a control update. The EBC scheme is very close to the best performance, although not exactly the best one, but it offers the least energy consumption, and especially the best trade-off.

3.5.3 A posteriori stability

In this subsection, we check that, on a given domain around the equilibrium, the intended direction by our policy leads toward the equilibrium. Of course, this cannot be taken as a proof, since we can only plot the mean directions of the state, because the actual direction depends on the random variable β (the channel erasure probability), and on the noise which is not taken into account in Figure 3.5 since it has a zero mean.

Figure 3.5 plots a vector field of the directions of the state on a grid, corresponding to the one considered in Figure 3.2. This figure is not meant to constitute a proof of the stability of our scheme, it is just a hint that shows the behaviour of the policy around the equilibrium point. One can see that the state is driven toward the equilibrium.

3.5.4 Influence of λ

Finally, we keep the same system to study the influence of the discout factor λ. We plot online run of the same system with different values of λ on Figure 3.6. On that figure, we plot an online run without noise nor dropout, and we observe along the time the output of the system, x_k, the cumulated energy and the cumulated cost (which we both want to minimise).

Although it is clear on Figure 3.6 that the performance is better when λ goes closer to 1, the influence on the cumulated energy does not expose a clear trend. In terms of energy, the highest value ($\lambda = 0.95$) shows an interesting behaviour, it preceeds the other runs in terms of energy consumption in order to keep the system close to its equilibrium, but this allows the scheme to run a longer time

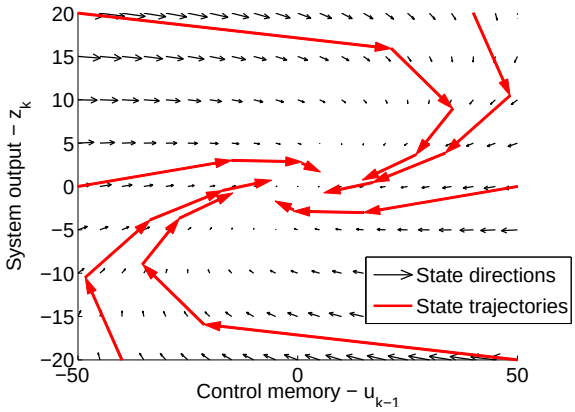


Figure 3.5 *– Stability check - Several trajectories of different states are plotted in a vector fields giving the mean directions of the state for a given radio mode.*

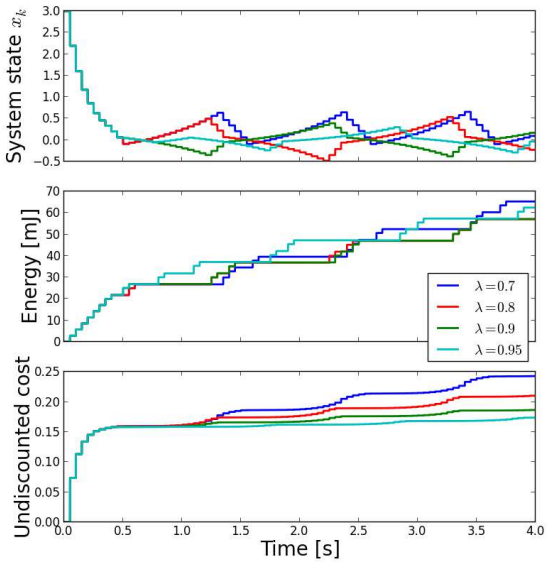


Figure 3.6 *– Illustration of the λ parameters for online runs of the same system using different values of λ. This plot depicts along the simulation time the system state x_k, the cumulated energy consumed by the radio chip in [mJ] and the undiscounted cost which does not account for the radio chip consumption nor the λ parameter.*

period open-loop, which leads to save energy afterwards. We could expect that the cost is lowered with higher value of λ since this parameters is used to weight more the current decisions than the future ones, which intuitively leads to consume more energy now than after. It appears that this trend is beneficial with this system that has a slow inertia around its equilibrium point. In other words, when the state is moved smoothly toward its equilibrium point, it will take more time to drift when running open-loop that it would have taken when the control update is coarse.

Chapter 4

Finite horizon optimisation for energy-aware control

Contents

Chapter 3 proposes a solution to the optimal problem introduced in Chapter 2 by deriving a joint radio-mode switching policy and feedback control law over an infinite horizon. In this chapter, the focus remains on the energy savings at the smart sensor side in a two nodes Networked Control System (NCS), however the horizon is now considered finite $H < \infty$ in the cost function (2.7).

Although we consider a finite horizon, we do not want to limit the analysis of the system to the duration of the horizon. Indeed, a practical application typically runs the system over a time period longer than the horizon which is often limited to 5 to 50 sampling intervals. Among the possible approaches to limit the analysis to a finite horizon while considering longer running time period, we decided to focus on the receding approach, explained in details in Section 4.2. Another common approach consists in averaging the cost function taken on an infinite horizon. This approach cannot be used in the context we consider, as it is explained in Section 4.6.

After defining the cost function associated to the finite horizon case, this chapter introduces the so-called Model Predictive Control (MPC) technique. Then we propose solutions of the optimisation problem in both the ideal case and the general case. Similarly to Chapter 3, the ideal case considers a simplified deterministic setup without additive noise nor message dropout. Then we prove the stability of the deterministic case in the framework of Input-to-State Stability (ISS). Finally, after discussions on the average cost per stage and an alternative implementation, we illustrate the solutions on simulation results.

4.1 The cost function in the finite case

It has to be clarified that we consider a finite receding horizon (explained in details in Section 4.2). Thus the cost function is considered for any time instant k up to $k+H-1$, H being the length of the horizon:

$$J_{\mathscr{U},\mathscr{V}}(z_k, m_k) = \mathop{\mathsf{E}}_{\substack{\beta_k,\omega_k \\ k=0,1,\dots}} \left[\ell_F(z_{k+H}, m_{k+H}) + \sum_{i=k}^{k+H-1} \lambda^i \ell_{v_i}(z_i, m_i, \hat{u}_i, \beta_i) \right].$$

The final cost $\ell_F(z_H, m_H)$ is chosen quadratic in accordance with the cost-to-go ℓ_{v_k} introduced in Equation (2.6):

$$\ell_F(z_H, m_H) = z_{k+H}^\top Q_F z_{k+H}.$$

The symmetric positive semidefinite matrix Q_F is a weight factor on the final augmented state at the end of the horizon. Its design plays a crucial role in guaranteeing stability of the NCS, see Section 4.4.

The discount factor λ is not used and then taken equal to 1. Indeed, the cost function is always finite in this case, and the discount factor is not needed to make the problem well-posed (see Section 3.1). This factor may still be used to weight the importance of immediate actions, however in a receding horizon setup, it is more natural to change the length of the horizon to do so.

The cost function considered for this chapter is then:

$$J_{\mathscr{U},\mathscr{V}}(z_k, m_k) = \mathop{\mathsf{E}}_{\substack{\beta_k,\omega_k \\ k=0,1,\dots}} \left[z_{k+H}^\top Q_F z_{k+H} + \sum_{i=k}^{k+H-1} \ell_{v_i}(z_i, m_i, \hat{u}_i, \beta_i) \right].$$

4.2 Model Predictive Control

Model Predictive Control (MPC) is a powerful tool to design controlled systems in the field of optimal and predictive control. Our main motivation to consider the framework of MPC is the use of the receding horizon, although the MPC approach provides other features that we do not exploit in the present work, such as the possibilities to set hard constraints, to use a prediction horizon different from the control horizon, or to smooth the control action with the introduction of a tracking trajectory converging to the set point. All the features of MPC make the stability analysis difficult. Authors in [Grüne and Pannek, 2011, Rawlings and Mayne, 2009] discuss the theory and the stability of MPC.

The notion of receding horizon is central in the framework of MPC. At time k, the controller computes a sequence of control inputs over the entire horizon, *i.e.* from time k to time $k+H-1$, from the measurement of the sensors at time k. The first element of the control sequence is applied, and then, at the next sampling time $k+1$, the whole procedure is repeated again. The sequence of control input is computed again over the horizon from time $k+1$ to $k+H$. This permits to take into account the new measurements from the sensors at time $k+1$. The procedure is repeated *ad-infinitum*.

A major drawback of MPC is that an optimisation problem over an horizon H has to be solved at each sampling period. A lot of effort has been put in tackling this problem, using various approaches such as fast or explicit MPC which consist in solving a suboptimal or parametrised problem offline in order to lighten the online burden, see *e.g.* [Bemporad et al., 2002b, Alessio and Bemporad, 2009]. Other approaches have also been considered: branch and bound algorithm, Linear Programming approach [Bemporad et al., 2002a], or parameterisation [Alamir, 2012], among others.
The Value Iteration method that we propose includes an offline step (which lighten the online burden), see Section 2.5.4. Moreover, to fasten further the online computation, our approach allows to derive offline a lookup table. This makes the use Dynamic Programming attractive, but the framework of Dynamic Programming prevents from considering constraints on the system.

4.3 Finite horizon solution - Ideal case

4.3.1 Mathematical model for the closed-loop system

The ideal case considers the system introduced in Chapter 2 in the deterministic case ($\beta_k = 1$ and $w_k = \mathbf{0}$ for all $k \geq 0$) but contrary to the ideal case in Chapter 3, the feedback control law is not static,

and it has to be derived with the optimal switching policy. Note that a static gain feedback has been considered in Chapter 3 to introduce the problem in its simplest formulation. However, the derivation of the feedback law jointly with the switching decision is possible, as shown afterwards, and brings interesting mathematical results. Finally, only one mode allows a transmission, *i.e.* $\mathbb{M}_1 = \{1\}$. The mathematical model is recall hereafter.

System model We consider the deterministic model:

$$x_{k+1} = Ax_k + Bu_k.$$

Channel model The channel is considered perfect.

Switching policy The switching decision is given by $\nu_k = \eta(x_k, u_{k-1}, m_k)$.

Feedback law The control input applied to the system depends on the switching decision:

$$u_k = \begin{cases} \mu(x_k, u_{k-1}, m_k), & \text{if } \nu_k = 1 \text{ (\textit{i.e.} Tx)}, \\ u_{k-1}, & \text{otherwise.} \end{cases}$$

Note that because the channel is perfect, when the control update computed by the smart node (called $\hat{u}_k$ previously) is sent to the actuator, it is actually applied to the system, so that $u_k = \hat{u}_k$ for all k where a transmission is scheduled. Hereafter, only the notation u_k will be used.

Radio chip model The state of the radio chip is the mode at time k, denoted as $m_k \in \mathbb{M}$. The radio-mode is updated according to the switching decision: $m_{k+1} = \nu_k$. The number of radio-mode is N and the transition costs between the modes is captured by the matrix Θ.

Switched model We recall that z_k is the augmented state, with the control memory $\tilde{u}_k = u_{k-1}$, $z_k = \begin{bmatrix} x_k^\top & \tilde{u}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x+n_u}$. The switched model is given by:

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k, u_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ u_k = \mu(z_k, m_k), \end{cases} \tag{4.1}$$

where the function f_{ν_k} is defined as

$$f_{\nu_k}(z_k, u_k) = \Phi_{\nu_k} z_k + \Gamma_{\nu_k} u_k,$$

and the matrices $\Phi_{\nu_k}, \Gamma_{\nu_k}$, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k = 1$, *i.e.* if there is a transmission, then

$$\Phi_{\nu_k} = \Phi_{CL} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \Gamma_{\nu_k} = \Gamma_{CL} = \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix}.$$

2. if $\nu_k \in \mathbb{M}_2$, *i.e.* if there is no transmission, then

$$\Phi_{\nu_k} = \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \Gamma_{\nu_k} = \Gamma_{OL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Finally, the cost-to-go is given by:

$$\ell_{\nu_k}(z_k, m_k, u_k) = z_k^\top Q_{\nu_k} z_k + u_k^\top R_{\nu_k} u_k + \theta_{m_k, \nu_k},$$

where the matrices Q_{ν_k} and R_{ν_k}, for $\nu_k \in \mathbb{M}$, are as follows:

1. if $\nu_k = 1$, *i.e.* if there is a transmission, then

$$Q_{\nu_k} = Q_{CL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad R_{\nu_k} = R_{CL} = \bar{R}.$$

2. if $v_k \in \mathbb{M}_2$, *i.e.* if $u_k = \tilde{u}_k$, then

$$Q_{v_k} = Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix}, \quad R_{v_k} = R_{OL} = \mathbf{0}.$$

Optimal problem

The optimisation problem, which is solved at each sampling time, consists in finding the optimal feedback sequence $\mathscr{U}^*_{k,H}$ and the optimal switching sequence $\mathscr{V}^*_{k,H}$ such that:

$$J^*(z_k, m_k) \triangleq J_{\mathscr{U}^*,\mathscr{V}^*}(z_k, m_k) = \min_{\mathscr{U},\mathscr{V}} J_{\mathscr{U},\mathscr{V}}(z_k, m_k),$$

where the cost function $J_{\mathscr{U},\mathscr{V}}(z_k, m_k)$ is given by:

$$J_{\mathscr{U},\mathscr{V}}(z_k, m_k) = z_{k+H}^\top Q_F z_{k+H} + \sum_{i=k}^{k+H-1} \ell_{v_i}(z_i, m_i, u_i). \tag{4.2}$$

It is worth noting that the problem is time invariant, which means that the control inputs and switching decisions depend only on the horizon step and the corresponding state. It is indeed sufficient to solve the problem with the state as a parameter.

4.3.2 Computation of the Value Function iterations

Contrary to the infinite case where one has to make sure that the Value Iteration method (as introduced in Section 2.5) converges to the optimal cost function, in the finite case the Value Iteration method is just a consequence of the Bellman's Principle of Optimality. It can be applied to a receding horizon thanks to the fact that the problem is time invariant [Görges et al., 2011]. Thus the algorithm given in Section 2.5.2 applies here without any further comment. We just notice that the optimal sequences, $\mathscr{U}^*$ and $\mathscr{V}^*$, are given by $u^*_{k+i} = \mu^*_i(z_{k+i}, m_{k+i})$, $v^*_{k+i} = \eta^*_i(z_{k+i}, m_{k+i})$ respectively, for $i \in \{0, \dots, H-1\}$, where μ^*_i and η^*_i are given by Equation (2.10) adapted to the ideal case, and re-computed hereafter in Equations (4.6) and (4.5).

As done in the infinite case, the Value Function takes the following explicit form in the deterministic case:

$$J^*(z_k, m_k) = V_H(z_k, m_k) = \min_{(\Pi,\boldsymbol{\pi}) \in \mathscr{P}_H} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\} \tag{4.3}$$

where $\mathscr{P}_H$ is a set of elements $(\Pi, \boldsymbol{\pi})$ with Π a square matrix of dimension $n_x + n_u$ and $\boldsymbol{\pi}$ a vector of dimension N. π_i refers to the i^{th} component of the vector $\boldsymbol{\pi}$. $\mathscr{P}_H$ is computed with the following recursion:

$$\begin{aligned}
&\mathscr{P}_0 = \{(Q_F, \mathbf{0})\} \\
&\mathscr{P}_i = \mathscr{P}_i^{(1)} \cup \mathscr{P}_i^{(2)} \\
&\mathscr{P}_i^{(1)} = \left\{ \begin{array}{c} \left(Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL} - \kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Phi_{CL}, \left[\theta_{1,1} + \pi_1 \quad \theta_{2,1} + \pi_1 \quad \dots \quad \theta_{N,1} + \pi_1 \right] \right) \\ \text{such that: } (\Pi, \boldsymbol{\pi}) \in \mathscr{P}_{i-1} \\ \text{and } \kappa_\Pi = \left(R_{CL} + \Gamma_{CL}^\top \Pi \Gamma_{CL} \right)^{-1} \Gamma_{CL}^\top \Pi \Phi_{CL} \end{array} \right\} \\
&\mathscr{P}_i^{(2)} = \left\{ \begin{array}{c} \left(Q_{OL} + \Phi_{OL}^\top \Pi \Phi_{OL}, \left[\min_\nu \left\{ \theta_{1,\nu} + \pi_\nu \right\} \quad \min_\nu \left\{ \theta_{2,\nu} + \pi_\nu \right\} \quad \dots \quad \min_\nu \left\{ \theta_{N,\nu} + \pi_\nu \right\} \right] \right) \\ \text{such that: } (\Pi, \boldsymbol{\pi}) \in \mathscr{P}_{i-1} \end{array} \right\}.
\end{aligned} \tag{4.4}$$

Recursively, for $i \in \{0, 1, \dots, H-1\}$ and for a given (z_{k+i}, m_{k+i}), let:
$(\Pi_{H-i}, \boldsymbol{\pi}_{H-i}) = \underset{(\Pi,\boldsymbol{\pi}) \in \mathscr{P}_{H-i}}{\arg\min} \left\{ z_{k+i}^\top \Pi z_{k+i} + \pi_{m_{k+i}} \right\}$.
The optimal switching decision at time $k+i$ is given by:

$$v^*_{k+i} = \eta^*_i(z_{k+i}, m_{k+i}) \triangleq \begin{cases} 1 & \text{if } (\Pi_{H-i}, \boldsymbol{\pi}_{H-i}) \in \mathscr{P}_{H-i}^{(1)} \\ \underset{\nu \in \mathbb{M}_2}{\arg\min} \left\{ \theta_{m_{k+i},\nu} + \pi_\nu \right\} & \text{if } (\Pi_{H-i}, \boldsymbol{\pi}_{H-i}) \in \mathscr{P}_{H-i}^{(2)}. \end{cases} \tag{4.5}$$

When $v^*_{k+i} = 1$, the optimal control input is given by:

$$u^*_{k+i} = \mu^*_i(z_{k+i}, m_{k+i}) \triangleq -\kappa_{\Pi_{H-i}} z_{k+i} = -K_{\Pi_{H-i}} x_{k+i} \tag{4.6}$$

where $\kappa_{\Pi_i} = \begin{bmatrix} K_{\Pi_i} & \mathbf{0} \end{bmatrix} = \left(R_{CL} + \Gamma_{CL}^\top \Pi_i \Gamma_{CL}\right)^{-1} \Gamma_{CL}^\top \Pi_i \Phi_{CL}$.
Finally, $z_{k+i+1} = f_{v^*_{k+i}}(z_{k+i}, u^*_{k+i})$.

The correctness of the above expressions for $\mathscr{P}_i$ can be proved using mathematical induction.

a) Base case:

By definition, $V_0(z,m) = z^\top Q_F z$ for all $(z,m) \in \mathbb{X}$. $V_1(z,m)$ is computed with the iteration algorithm (4.4):

$$\begin{aligned}
V_1(z,m) &= \min_{(u,v)\in\mathbb{U}(z)} \{\ell_v(z,m,u) + V_0(f_v(z,u),v)\} \\
&= \min_{(u,v)\in\mathbb{U}(z)} \Big\{ z^\top Q_v z + u^\top R_v u + \theta_{m,v} + z^\top \Phi_v^\top Q_F \Phi_v z + u^\top \Gamma_v^\top Q_F \Gamma_v u \\
&\quad + 2u^\top \Gamma_v^\top Q_F \Phi_v z \Big\} \\
&= \min_{(u,v)\in\mathbb{U}(z)} \Big\{ z^\top (Q_v + \Phi_v^\top Q_F \Phi_v) z + u^\top (R_v + \Gamma_v^\top Q_F \Gamma_v) u + \theta_{m,v} + 2u^\top \Gamma_v^\top Q_F \Phi_v z \Big\} \\
&= \min \Bigg\{ \min_{u\in\mathbb{R}^{n_u}} \Big\{ z^\top (Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL}) z + u^\top (R_{CL} + \Gamma_{CL}^\top Q_F \Gamma_{CL}) u + \theta_{m,1} \\
&\qquad + 2u^\top \Gamma_{CL}^\top Q_F \Phi_{CL} z \Big\}; \min_{v\in\mathbb{M}_2} \left\{ z^\top (Q_{OL} + \Phi_{OL}^\top Q_F \Phi_{OL}) z + \theta_{m,v} \right\} \Bigg\}
\end{aligned} \tag{4.7}$$

We denote $\Psi(u) = z^\top (Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL}) z + u^\top (R_{CL} + \Gamma_{CL}^\top Q_F \Gamma_{CL}) u + \theta_{m,1} + 2u^\top \Gamma_{CL}^\top Q_F \Phi_{CL} z$. One can compute the value of u that minimises the previous equation, denoted u^*, as follows:

$$\begin{aligned}
\frac{\partial \Psi(u)}{\partial u} &= \left(2(R_{CL} + \Gamma_{CL}^\top Q_F \Gamma_{CL}) u + 2\Gamma_{CL}^\top Q_F \Phi_{CL} z\right) \\
\left.\frac{\partial \Psi(u)}{\partial u}\right|_{u^*} = 0 \quad &\Rightarrow u^* = -(\Gamma_{CL}^\top Q_F \Gamma_{CL} + R_{CL})^{-1} \Gamma_{CL}^\top Q_F \Phi_{CL} z \triangleq -\kappa_{Q_F} z.
\end{aligned}$$

We can check that u^* actually exists and is a minimum. First of all, one should notice that $(\Gamma_{CL}^\top Q_F \Gamma_{CL} + R_{CL})$ is positive definite, as the sum of the positive definite matrices $\Gamma_{CL}^\top Q_F \Gamma_{CL}$ and R_{CL}, then the inverse exists, so as κ_{Q_F}. The same argument can be used to prove that the extremum is actually a minimum since one can check that Hessian matrix of $\Psi(u^*)$ is positive definite. Note also that the minimum in Equation (4.7) is indeed attained for u^*.

We can also check that the minimisation on u provides a state feedback gain depending only on x. Assuming that Q_F can be written $\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix}$, and denoting $\hat{R}_{CL} \triangleq R_{CL} + B^\top Q_{11} B + B^\top Q_{12} + Q_{12} B + Q_{22}$, we have:

$$\begin{aligned}
\kappa_{Q_F} &= \left(R_{CL} + \Gamma_{CL}^\top \Pi \Gamma_{CL}\right)^{-1} \Gamma_{CL}^\top \Pi \Phi_{CL} \\
&= \left(\hat{R}_{CL}\right)^{-1} \begin{bmatrix} (B^\top Q_{11} + Q_{12}) A & 0 \end{bmatrix}
\end{aligned} \tag{4.8}$$

From the form of κ_{Q_F}, one sees that only the component on x from z is used to compute u^*.

We notice that $\Psi(u^*)$ can be written as follows:

$$\begin{aligned}
\Psi(\hat{u}^*) =& z^\top (Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL}) z + u^\top \overbrace{\left[(R_{CL} + \Gamma_{CL}^\top Q_F \Gamma_{CL}) u + \Gamma_{CL}^\top Q_F \Phi_{CL} z\right]}^{=0} \\
&+ u^\top \Gamma_{CL}^\top Q_F \Phi_{CL} z + \theta_{m,1} \\
\Psi(\hat{u}^*) =& z^\top (Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL}) z - z^\top \kappa_{Q_F}^\top \Gamma_{CL}^\top Q_F \Phi_{CL} z + \theta_{m,1}.
\end{aligned} \tag{4.9}$$

Plugging u^* in Equation (4.7) yields:

$$\begin{aligned} V_1(z,m) &= \min\Bigg\{ z^\top (Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL} - \kappa_{Q_F}^\top \Gamma_{CL}^\top Q_F \Phi_{CL}) z + \theta_{m,1}; \\ &\qquad\qquad z^\top (Q_{OL} + \Phi_{OL}^\top Q_F \Phi_{OL}) z + \min_{v \in \mathbb{M}_2} \left\{ \theta_{m,v} \right\} \Bigg\} \\ V_1(z,m) &= \min_{(\Pi,\pi) \in \mathscr{P}_1} \left\{ z^\top \Pi z + \pi_m \right\}, \end{aligned}$$

where

$$\mathscr{P}_1 = \left\{ \begin{array}{l} \left(Q_{CL} + \Phi_{CL}^\top Q_F \Phi_{CL} - \kappa_{Q_F}^\top \Gamma_{CL}^\top Q_F \Phi_{CL}, \begin{bmatrix} \theta_{1,1} & \theta_{2,1} & \dots & \theta_{N,1} \end{bmatrix} \right), \\ \left(Q_{OL} + \Phi_{OL}^\top Q_F \Phi_{OL}, \begin{bmatrix} \min_{v \in \mathbb{M}_2} \left\{ \theta_{1,v} \right\} & \min_{v \in \mathbb{M}_2} \left\{ \theta_{2,v} \right\} & \dots & \min_{v \in \mathbb{M}_2} \left\{ \theta_{N,v} \right\} \end{bmatrix} \right) \end{array} \right\}$$

b) Inductive step:

We assume that the Value Function at i^{th} iteration is given by:

$$V_i(z,m) = \min_{(\Pi,\pi) \in \mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}.$$

The computation of the Value Function at the next iteration, using the iteration algorithm (4.4), is very similar to the base case:

$$\begin{aligned} V_{i+1}(z,m) &= \min_{(u,v) \in \mathbb{U}(z)} \left\{ \ell_v(z,m,u) + V_i(f_v(z,u),v) \right\} \\ &= \min_{(u,v) \in \mathbb{U}(z)} \Bigg\{ z^\top Q_v z + u^\top R_v u + \theta_{m,v} + \min_{(\Pi,\pi) \in \mathscr{P}_i} \Big\{ z^\top \Phi_v^\top \Pi \Phi_v z + u^\top \Gamma_v^\top \Pi \Gamma_v u \\ &\qquad\qquad + 2u^\top \Gamma_v^\top \Pi \Phi_v z + \pi_v \Big\} \Bigg\} \\ &= \min \Bigg\{ \min_{u \in \mathbb{R}^{n_u}, (\Pi,\pi) \in \mathscr{P}_i} \Big\{ z^\top (Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL}) z + u^\top (R_{CL} + \Gamma_{CL}^\top \Pi \Gamma_{CL}) u + \theta_{m,1} \\ &\qquad + 2u^\top \Gamma_{CL}^\top \Pi \Phi_{CL} z + \pi_1 \Big\}; \min_{v \in \mathbb{M}_2, (\Pi,\pi) \in \mathscr{P}_i} \left\{ z^\top (Q_{OL} + \Phi_{OL}^\top \Pi \Phi_{OL}) z + \theta_{m,v} + \pi_v \right\} \Bigg\} \end{aligned}$$

Note that $z \Phi_v^\top \Pi \Gamma_v u = u^\top \Gamma_v^\top \Pi \Phi_v z$ since the Π matrices in $\mathscr{P}_j$ are symmetric for any admissible j, see Lemma 4.1.

As in the base case, we can compute $u^* = -(\Gamma_{CL}^\top \Pi \Gamma_{CL} + R_{CL})^{-1} \Gamma_{CL}^\top \Pi \Phi_{CL} z \triangleq -\kappa_\Pi z$, which yields:

$$\begin{aligned} V_{i+1}(z,m) &= \min\Bigg\{ z^\top (Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL} - \kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Phi_{CL}) z + \theta_{m,1} + \pi_1; \\ &\qquad\qquad z^\top (Q_{OL} + \Phi_{OL}^\top \Pi \Phi_{OL}) z + \min_{v \in \mathbb{M}_2} \left\{ \theta_{m,v} + \pi_v \right\} \Bigg\} \\ V_{i+1}(z,m) &= \min_{(\Pi,\pi) \in \mathscr{P}_{i+1}} \left\{ z^\top \Pi z + \pi_m \right\}, \end{aligned}$$

where

$$
\begin{aligned}
&\mathscr{P}_{i+1} = \mathscr{P}_{i+1}^{(1)} \cup \mathscr{P}_{i+1}^{(2)} \\
&\mathscr{P}_{i+1}^{(1)} \triangleq \Big\{ \big(Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL} - \kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Phi_{CL}, \\
&\qquad \begin{bmatrix} \pi_1 + \theta_{1,1} & \pi_1 + \theta_{2,1} & \dots & \pi_1 + \theta_{N,1} \end{bmatrix} \big) \\
&\qquad \text{such that } (\Pi, \pi) \in \mathscr{P}_i \text{ and } \kappa_\Pi = (\Gamma_{CL}^\top \Pi \Gamma_{CL} + R_{CL})^{-1} \Gamma_{CL}^\top \Pi \Phi_{CL} \Big\} \\
&\mathscr{P}_{i+1}^{(2)} \triangleq \left\{ \left(Q_{OL} + \Phi_{OL}^\top \Pi \Phi_{OL}, \begin{bmatrix} \min_{\nu \in \mathbb{M}_2} \{\theta_{1,\nu} + \pi_\nu\} \\ \min_{\nu \in \mathbb{M}_2} \{\theta_{2,\nu} + \pi_\nu\} \\ \vdots \\ \min_{\nu \in \mathbb{M}_2} \{\theta_{N,\nu} + \pi_\nu\} \end{bmatrix}^\top \right) \text{ such that } (\Pi, \pi) \in \mathscr{P}_i \right\}.
\end{aligned}
$$

It is worth noticing that some parts of the computation are only valid when the matrix Π is symmetric semidefinite positive, which imposes that the matrix Q_F is itself symmetric semidefinite positive. Finally, the optimal cost over the whole horizon is naturally given by:

$$J^*(z_k, m_k) = V_H(z_k, m_k).$$

To conclude, we prove that the Π matrices are symmetric positive definite.

Lemma 4.1. *The Π matrices in $\mathscr{P}_i$ are symmetric positive definite for any admissible $i > 0$.*

Proof. The recursion used to compute the Π matrices is given in Equation (4.4). Each Π matrix in $\mathscr{P}_i$ generates one matrix in $\mathscr{P}_{i+1}^{(1)}$ and one other in $\mathscr{P}_{i+1}^{(2)}$ denoted hereafter $\Pi_+^{(1)}$ and $\Pi_+^{(2)}$, respectively. Equation (4.4) yields:

$$
\begin{aligned}
\Pi_+^{(1)} &= Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL} - \kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Phi_{CL} \\
&= Q_{CL} + \Phi_{CL}^\top \Pi \Phi_{CL} + \kappa_\Pi^\top R_{CL} \kappa_\Pi + \kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Gamma_{CL} \kappa_\Pi - 2\kappa_\Pi^\top \Gamma_{CL}^\top \Pi \Phi_{CL} \\
&\quad \text{(thanks to Equation (4.9) where } Q_F \text{ is replaced by } \Pi) \\
&= Q_{CL} + \Phi'^\top \Pi \Phi' + \kappa_\Pi^\top R_{CL} \kappa_\Pi \qquad (4.10) \\
&\quad \text{where } \Phi' = \Phi_{CL} - \kappa_\Pi \Gamma_{CL}
\end{aligned}
$$

and

$$\Pi_+^{(2)} = Q_{OL} + \Phi_{OL}^\top \Pi \Phi_{OL} \qquad (4.11)$$

We recall that Q_{CL}, Q_{OL}, R_{CL} and Q_F are symmetric semidefinite positive. Also, for any matrices of appropriate dimensions Λ and Ξ not null, if Λ is symmetric positive definite, then $\Xi^\top \Lambda \Xi$ is also symmetric positive definite.
Then, if the Π matrices in $\mathscr{P}_i$ are symmetric positive definite, Equations (4.10)-(4.11) prove that the Π matrices in $\mathscr{P}_{i+1}$ are also symmetric positive definite. Since the Π matrix in $\mathscr{P}_0$ is Q_F (which is symmetric definite positive) then this proves that the Π matrices in $\mathscr{P}_i$ are symmetric positive definite for any $i \geq 0$.

□

4.3.3 MPC implementation

It is important to notice that the optimisation problem provides an optimal feedback sequence $\mathscr{U}^*$ and an optimal switching sequence $\mathscr{V}^*$ along the horizon, while a stationary policy (μ, η) is considered in the closed-loop form (4.1). Indeed, because we consider a receding horizon, only the first elements of the sequences are used and the optimisation problem is solved again at the next sampling period. The stationary joint policy is then given by selecting only the first elements of the optimal sequences $\mathscr{U}^*$, $\mathscr{V}^*$ (computed thanks to Equations (4.6) and (4.5)), *i.e.* :

$$
\begin{cases}
\mu^*(z_k, m_k) = \mu_0^*(z_k, m_k) \\
\eta^*(z_k, m_k) = \eta_0^*(z_k, m_k),
\end{cases} \qquad (4.12)
$$

where $\mu_0^*(z_k, m_k)$ and $\eta_0^*(z_k, m_k)$ are derived in Section 4.3.2. This selection is possible because the problem is time invariant.

As in the infinite case, the computations are performed in two parts. The first part is run offline and consists in pre-computing the set $\mathscr{P}_H$ and the associated gains κ_Π, as detailed in Equation (4.4). The second part is run online and consists in running the following algorithm on the smart node at each sampling time, knowing (z_k, m_k):

1. Compute $(\Pi_k, \pi_k) = \underset{(\Pi,\pi)\in\mathscr{P}_H}{\arg\min} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\}$ for the current state (z_k, m_k).
2. Compute the switching decision, *i.e.* the next radio-mode, $m_{k+1} = v_k^* = \eta^*(z_k, m_k)$ from Equation (4.12) and
   - if $v_k^* = 1$, compute the control input $u_k^* = \mu^*(z_k, m_k)$ from Equation (4.12) and send it to the actuator,
   - if $v_k^* \neq 1$, do nothing.

This leads to an Event-Based Control (EBC) law where the mode switching policy η^* triggers control updates based on the current state (z_k, m_k).

The online burden in the smart node can be lightened by replacing the exact minimisation of Equation (4.3) in step 1) of the previous algorithm with a suboptimal solution by using a simple lookup table based on a grid of the state space, which accuracy depends on the precision needed and the computation resources available.

4.4 Input-to-State practical Stability

This section discusses the stability of the closed-loop system designed in Section 4.3. But before proceeding, we need to define some notations:

- A function $\alpha(s) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be a $\mathscr{K}$-function, or of class $\mathscr{K}$, if it is continuous, strictly increasing and $\alpha(0) = 0$.
- A function $\alpha(s) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be a $\mathscr{K}_\infty$-function, or of class $\mathscr{K}_\infty$, if it is a $\mathscr{K}$-function and $\alpha(s) \to \infty$ as $s \to \infty$.
- A function $\gamma(s, k) : \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be a $\mathscr{KL}$-function, or of class $\mathscr{KL}$, if for each fixed $k \geq 0$, $\gamma(\cdot, k)$ is of class $\mathscr{K}$, for each fixed $s \geq 0$, $\gamma(s, \cdot)$ is decreasing, and $\gamma(s, k) \to 0$ as $k \to \infty$.
- $\alpha^{-1}(s)$ denotes, when it exists, the inverse of function $\alpha(s)$.
- $\alpha_1 \circ \alpha_2(s)$ denotes the composition of functions $\alpha_1(s)$ and $\alpha_2(s)$ such that $\alpha_1 \circ \alpha_2(s) = \alpha_1(\alpha_2(s))$.
- $\mathrm{id}(s)$ is the identity function, $\mathrm{id}(s) = s$.
- $\|s\|$ denotes the Euclidean norm of $s \in \mathbb{R}^n$.

4.4.1 Definitions and stability result

We are interested in the stability of the closed-loop state trajectories of system (4.1) with the policy (4.12) and the initial conditions (z_0, m_0), that we note $z_k(z_0, m_0)$, and which evolves as follows:

$$\begin{cases} z_{k+1}(z_0, m_0) = f_{v_k^*}(z_k(z_0, m_0), u_k^*) \\ m_{k+1} = v_k^* = \eta^*(z_k, m_k) \\ u_k^* = \mu^*(z_k, m_k). \end{cases} \tag{4.13}$$

The stability analysis of that system relies on the framework of ISS. We consider the following definition of practical stability.

Definition 4.1. *The closed-loop system* (4.13) *is said to be Globally Input-to-State practically Stable (GISpS) if there exist a $\mathcal{KL}$-function γ, and a constant $c \geq 0$, such that, for all $(z_0, m_0) \in \mathbb{X}$:*

$$\|z_k(z_0, m_0)\| \leq \gamma(\|z_0\|, k) + c, \quad k \in \mathbb{Z}_{\geq 0}. \tag{4.14}$$

Definition 4.1 implies that the closed-loop system verifies the following proposition:

Proposition 4.1. *If the closed-loop system* (4.13) *is GISpS, as stated in Definition 4.1, then for any positive scalar ϵ and for all initial conditions (z_0, m_0), there exists a finite time $\bar{k}(z_0, m_0, \epsilon)$ such that:*

$$\|z_k\| \leq c + \epsilon, \ \forall k \geq \bar{k}(z_0, m_0, \epsilon).$$

Proof. The proof immediately follows from Equation (4.14) and the fact that $\gamma(\|z_0\|, k) \to 0$ as $k \to \infty$. □

Our stability result is subject to a condition on the final weighting factor Q_F in (4.2), described in the following assumption.

Assumption 4.1. *The weighting factor Q_F in Equation* (4.2) *is such that there exists $\kappa \in \mathbb{R}^{n_u \times (n_x + n_u)}$ such that:*

$$\begin{aligned} Q_F &\succ 0 \\ \underbrace{(\Phi_{CL} - \Gamma_{CL}\kappa)^\top Q_F (\Phi_{CL} - \Gamma_{CL}\kappa) - Q_F + Q_{CL} + \kappa^\top R_{CL}\kappa}_{\triangleq -Q'_F} &\preceq 0 \\ \text{and} \quad \max\{|\operatorname{eigs}(\Phi_{CL} - \Gamma_{CL}\kappa)|\} &\leq 1. \end{aligned} \tag{4.15}$$

The existence of such Q_F in the scalar case is discussed at the end of this section. Note that the existence of the gain κ is only used to prove that Assumption 4.1 holds, this gain is not used afterwards as the closed-loop feedback gain.

To prove the practical stability of the closed-loop system, we use a GISpS-Lyapunov function V, which is defined in the following definition.

Definition 4.2. *$V : \mathbb{X} \to \mathbb{R}_{\geq 0}$ is called a GISpS-Lyapunov function for the system* (4.13) *if*

- *there exists a pair of $\mathcal{K}_\infty$-functions α_1, α_2, and a constant $c_1 \geq 0$ such that, for all $(z, m) \in \mathbb{X}$:*

$$\alpha_1(|z|) \leq V(z, m) \leq \alpha_2(|z|) + c_1 \tag{4.16}$$

- *there exists a suitable $\mathcal{K}_\infty$-function α_3 and a constant $c_2 \geq 0$ such that, for all $(z, m) \in \mathbb{X}$:*

$$\begin{aligned} \Delta V(z, m) &\triangleq V(f_{v^*}(z, m, u^*)) - V(z, m) \\ &\leq -\alpha_3(|z|) + c_2 \end{aligned} \tag{4.17}$$

Now, we give the two main theorems that prove that the closed-loop system is Input-to-State stable.

Theorem 4.1. *If the system* (4.13) *admits a GISpS-Lyapunov function, then it is GISpS.*

Theorem 4.2. *Under assumption 4.1, the closed-loop* (4.13) *is GISpS.*

Before proving these theorems, we discuss the existence of the matrix Q_F. In the case where the system is scalar, *i.e.* when $n_x = n_u = 1$, we can show that such Q_F always exist. In the scalar case $A, B, \bar{Q}, \bar{R}$ belong to $\mathbb{R}$. We note, from Equation (4.8), that $\kappa = \begin{bmatrix} K & 0 \end{bmatrix}$, where $K \in \mathbb{R}$. We take Q_F on the form $\begin{bmatrix} \bar{Q}_F & 0 \\ 0 & \bar{R}_F \end{bmatrix}$ where $\bar{Q}_F$ and $\bar{R}_F$ belong to $\mathbb{R}$. Equation (4.15) yields:

$$\begin{aligned} (\Phi_{CL} - \Gamma_{CL}\kappa)^\top Q_F (\Phi_{CL} - \Gamma_{CL}\kappa) - Q_F + Q_{CL} + \kappa^\top R_{CL}\kappa &\preceq 0 \\ \begin{bmatrix} A - BK & -K \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{Q}_F & 0 \\ 0 & \bar{R}_F \end{bmatrix} \begin{bmatrix} A - BK & 0 \\ -K & 0 \end{bmatrix} - \begin{bmatrix} \bar{Q}_F & 0 \\ 0 & \bar{R}_F \end{bmatrix} + \begin{bmatrix} \bar{Q} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \bar{R}K^2 & 0 \\ 0 & 0 \end{bmatrix} &\preceq 0 \\ \begin{bmatrix} \bar{Q}_F (A - BK)^2 + \bar{R}_F K^2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -\bar{Q}_F + \bar{Q} + \bar{R}K^2 & 0 \\ 0 & -\bar{R}_F \end{bmatrix} &\preceq 0 \\ \begin{cases} \bar{Q}_F (A - BK)^2 + \bar{R}_F K^2 - \bar{Q}_F + \bar{Q} + \bar{R}K^2 \leq 0 \\ -\bar{R}_F \leq 0 \end{cases} \end{aligned}$$

Finding suitable $\bar{Q}_F$ and K in the previous statement is equivalent to find $\bar{Q}_F \geq 0$ and K such that $|A - BK| \leq 1$ and $\bar{Q}_F \leq \mathtt{f}(K)$, where the function $\mathtt{f}$ is defined by:

$$\mathtt{f}(K) = \frac{-(\bar{R} + \bar{R}_F)K^2 - 1}{(A - BK)^2 - 1}.$$

It can be shown that there exists $\hat{Q}$ such that for any $\bar{Q}_F \geq \hat{Q}$, one can find $\bar{K}_0 \in \left(\frac{A-1}{B}, \frac{A+1}{B}\right)$ (if $B > 0$, $\left(\frac{A+1}{B}, \frac{A-1}{B}\right)$ otherwise) such that $\bar{Q}_F \geq \mathtt{f}(K_0)$. And $\hat{Q}$ is given by:

$$\hat{Q} = \begin{cases} -\sigma + \sqrt{\sigma^2 + \dfrac{1}{\bar{R} + \bar{R}_F}} & \text{if } AB > 0 \\ -\sigma - \sqrt{\sigma^2 + \dfrac{1}{\bar{R} + \bar{R}_F}} & \text{if } AB < 0 \end{cases}$$
$$\sigma = \frac{B^2 - (\bar{R} + \bar{R}_F)(A^2 - 1)}{2(\bar{R} + \bar{R}_F)AB}$$

This result means that, in the scalar case, any $\bar{Q}_F \geq \hat{Q}$ satisfies Assumption 4.1.

When the system is not scalar, there is no such systematic computation to prove the existence of suitable Q_F. In the general case, one can use numerical tools to find a solution of the corresponding Linear Matrix Inequality (LMI) with the numerical values of the parameters Φ_{CL}, Γ_{CL}, Q_{CL}, R_{CL} and Q_F.

4.4.2 Proof of the practical stability

The proof of the stability is based on the framework of ISS, and relies on the works [Raimondo et al., 2009, Lazar et al., 2008] and [Jiang and Wang, 2001]. ISS is used to prove the stability of a system despite the presence of disturbance. Even though we don't want to consider any disturbances at first, the ISS framework is again of great help to prove stability in our case. The authors in [Raimondo et al., 2009] address the problem of regional Input-to-State practical Stability (ISpS). The notion of practical stability (defined by Definition 4.1) means that the state of the system is not converging asymptotically to a point, but to a set, and once this set is reached, the system remains in this set. The notion of regional stability refers to a system having constraints. Typically, there exists constraints on the disturbance, or on the state space or the control space. The authors in [Jiang and Wang, 2001] on the other hand consider *global* ISS, not taking constraint into account, and considering asymptotic stability to a point. Our contribution consists in considering ISS in the global and practical case, to prove stability of our system.

We will first prove Theorem 4.1, namely that if our system admits a GISpS-Lyapunov function, then it is stable in the sense of Definition 4.1. Then we prove Theorem 4.2, which shows that such GISpS-Lyapunov function exists in our case.

Proof of Theorem 4.1

This proof is based on the proofs of ISS and ISpS from [Jiang and Wang, 2001, Raimondo et al., 2009]. We assume that Equations (4.16)-(4.17) hold, *i.e.* that the system (4.13) admits an GISpS-Lyapunov function, denoted $V(z, m)$. Let's prove that the closed-loop system is GISpS in the sense of Definition 4.1. The proof is divided into three steps: first we prove that the closed-loop system admits an invariant set (as defined in Definition 4.3); then we prove that the invariant set is attractive; finally we prove that having an attractive invariant set is equivalent to the practical stability. But before proceeding, we give the definition of an invariant set.

Definition 4.3. *A set $\Omega \subset \mathbb{X}$ is called an invariant set for system (4.13) if any $(z, m) \in \Omega$ verifies $f_{\eta^*(z,m)}(z, \mu^*(z, m)) \in \Omega$.*

Step 1 First, we prove that the closed-loop system (4.13) admits an invariant set $\Omega \subset \mathbb{X}$.

We define $\overline{\alpha}_2(s) \triangleq \alpha_2(s) + s$, then, noting that $c_1 \geq 0$ and $\|z\| \geq 0$, (4.16) implies:

$$\begin{aligned} V(z,m) &\leq \alpha_2(\|z\| + c_1) + \|z\| + c_1 \\ &= \overline{\alpha}_2(\|z\| + c_1) \\ \Rightarrow \overline{\alpha}_2^{-1}(V(z,m)) &\leq \|z\| + c_1. \end{aligned} \tag{4.18}$$

Let $\xi(s)$ be a $\mathcal{K}_\infty$-function.

- If $c_1 \leq \|z\|$:

$$\begin{aligned} c_1 \leq \|z\| &\Leftrightarrow \frac{\|z\| + c_1}{2} \leq \|z\| \\ \Rightarrow \alpha_3\left(\frac{\|z\| + c_1}{2}\right) &\leq \alpha_3(\|z\|) \leq \alpha_3(\|z\|) + \xi(c_1). \end{aligned}$$

- If $c_1 > \|z\|$:

$$\begin{aligned} \|z\| < c_1 &\Leftrightarrow \frac{\|z\| + c_1}{2} < c_1 \\ \Rightarrow \xi\left(\frac{\|z\| + c_1}{2}\right) &\leq \xi(c_1) \leq \alpha_3(\|z\|) + \xi(c_1). \end{aligned}$$

Let's define $\underline{\alpha_3}(s) \triangleq \min\left\{\xi\left(\frac{s}{2}\right), \alpha_3\left(\frac{s}{2}\right)\right\}$. We notice that $\underline{\alpha_3} \in \mathcal{K}_\infty$, $\underline{\alpha_3}$ is then strictly increasing, which yields:

$$\underline{\alpha_3}(\|z\| + c_1) \leq \alpha_3(\|z\|) + \xi(c_1) \tag{4.19}$$

$$(4.18), (4.19) \Rightarrow \underline{\alpha_3}\left(\overline{\alpha}_2^{-1}(V(z,m))\right) \leq \alpha_3(\|z\|) + \xi(c_1).$$

Let's define $\alpha_4 \triangleq \underline{\alpha_3} \circ \overline{\alpha}_2^{-1}$, then:

$$\begin{aligned} \alpha_4(V(z,m)) &\leq \alpha_3(\|z\|) + \xi(c_1) \\ \Rightarrow -\alpha_3(\|z\|) - \xi(c_1) &\leq -\alpha_4(V(z,m)) \\ (4.17) \Rightarrow \Delta V(z,m) &\leq -\alpha_3(\|z\|) + c_2 - \xi(c_1) + \xi(c_1) \\ \Delta V(z,m) &\leq -\alpha_4(V(z,m)) + c_2 + \xi(c_1). \end{aligned} \tag{4.20}$$

Let ρ be a $\mathcal{K}_\infty$-function such that $(id - \rho)$ is also a $\mathcal{K}_\infty$-function. $\rho(s) = \frac{s}{2}$ is an example. We define $\Omega \subset \mathbb{X}$:

$$\Omega \triangleq \left\{(z,m) \in \mathbb{X} : V(z,m) \leq \omega(c_3)\right\}, \tag{4.21}$$

where $\omega \triangleq \alpha_4^{-1} \circ \rho^{-1}$ and $c_3 \triangleq c_2 + \xi(c_1)$.

We assume that $(id - \alpha_4)$ is a $\mathcal{K}_\infty$-function. Lemma B.1 in [Jiang and Wang, 2001] proves that if $(id - \alpha_4)$ is not a $\mathcal{K}_\infty$-function, there exists a $\mathcal{K}_\infty$-function $\hat{\alpha}_4$ such that $\hat{\alpha}_4(s) \leq \alpha_4(s)$ and $(id - \hat{\alpha}_4)$ is a $\mathcal{K}_\infty$-function that can be used hereafter to lead to the same result.

Let's now assume that $(z,m) \in \Omega$:

$$\begin{aligned} (4.20) \Rightarrow V(g_{v^*}(z,u^*),v^*) - V(z,m) &\leq -\alpha_4(V(z,m)) + c_3 \\ \Rightarrow V(g_{v^*}(z,u^*),v^*) &\leq (id - \alpha_4)(V(z,m)) + c_3 \\ &\leq (id - \alpha_4)\left(\omega(c_3)\right) + c_3 \\ &= \omega(c_3) - \alpha_4\left(\omega(c_3)\right) + c_3 \\ &= \omega(c_3) - \alpha_4\left(\omega(c_3)\right) + \rho \circ \alpha_4(\omega(c_3)) \\ &= \omega(c_3) - (id - \rho)(\alpha_4\left(\omega(c_3)\right) \end{aligned}$$

Indeed, from the definition of ω, it holds that $\rho \circ \alpha_4(\omega(s)) = s$, and since $(id - \rho)(s) \geq 0$ (being a $\mathcal{K}_\infty$-function), it yields:

$$V(g_{v^*}(z,u^*),v^*) \leq \omega(c_3). \tag{4.22}$$

Equation (4.22) with the assumption that z is in Ω implies that Ω is an invariant set for the policy (4.12).

Step 2 Let's now prove that the invariant set Ω is an attractive set, *i.e.* that for any $(z_k, m_k) \notin \Omega$, there exists a finite $\bar{k} \geq k$ such that, $(z_{\bar{k}}, m_{\bar{k}}) \in \Omega$.

We assume that, for any time t from 0 up to time k, $(z_t, m_t) \notin \Omega$. From Equation (4.21), and ignoring the time index, the following holds for any time up to k:

$$\begin{aligned} & V(z,m) > \omega(c_3) = \alpha_4^{-1} \circ \rho^{-1}(c_3) \\ \Rightarrow\ & \rho \circ \alpha_4(V(z,m)) > c_3 \\ \Leftrightarrow\ & \rho \circ \alpha_4(V(z,m)) - c_3 > 0. \end{aligned} \tag{4.23}$$

Moreover:

$$\begin{aligned} (4.20) \Rightarrow \Delta V(z,m) &\leq -\alpha_4(V(z,m)) + c_3 \\ &= -(id - \rho) \circ \alpha_4(V(z,m)) - \rho \circ \alpha_4(V(z,m)) + c_3 \\ (4.23) \Rightarrow \Delta V(z,m) &\leq -(id - \rho) \circ \alpha_4(V(z,m)). \end{aligned} \tag{4.24}$$

Let's define $\bar{k}$ as the first time index where the system enters Ω, for the initial condition (z_0, m_0):

$$\bar{k} \triangleq \min\left\{k \in \mathbb{Z}_{\geq 0} : z_k, m_k \in \Omega\right\} \leq \infty.$$

Remark 4.1. *We allow $\bar{k}$ to be infinite, which means that the trajectories never enter Ω. To prove that Ω is attractive, we need to prove that $\bar{k}$ is finite.*

Equation (4.24) holds for all (z_0, m_0) and for all z_k, m_k with $0 \leq k < \bar{k}$. According to [Jiang and Wang, 2002, Lemma 4.3], there exists a $\mathcal{KL}$-function $\hat{\gamma}(s,k)$ such that, $\forall\ (z_0, m_0) \in \mathbb{X}$:

$$V(z_k, m_k) \leq \hat{\gamma}(V(z_0, m_0), k),\ \forall k \in \{0, 1, \ldots, \bar{k}\}$$

It holds then that, for all $k \in \{0, 1, \ldots, \bar{k}\}$, the value function is decreasing and going to 0.
Assume that $\bar{k}$ is not finite, since $\omega(c_3) > 0$[1], there exists a finite time $\bar{t}$ such that $\hat{\gamma}(V(z_0, m_0), \bar{t}) \leq \omega(c_3)$, which contradicts the assumption that $\bar{k}$ is not finite. This proves that Ω is an attractive set.

Step 3 Finally, we want to prove that Equation (4.14) holds. We collect the results from the previous steps, $\forall (z_0, m_0) \in \mathbb{X}$, $\forall k \in \mathbb{Z}_{\geq 0}$:

- if $(z_k, m_k) \in \Omega$, $V(z_k, m_k) \leq \omega(c_3)$,
- if $(z_k, m_k) \notin \Omega$, $V(z_k, m_k) \leq \hat{\gamma}\left(V(z_0, m_0, k)\right)$.

Equation (4.16) implies that $\|z_k\| \leq \alpha^{-1}(V(z_k, m_k))$, we thus obtain:

- if $(z_k, m_k) \in \Omega$, $\|z_k\| \leq \alpha^{-1}\left(\omega(c_3)\right)$,
- if $(z_k, m_k) \notin \Omega$, $\|z_k\| \leq \alpha^{-1}\left(\hat{\gamma}\left(V(z_0, m_0), k\right)\right)$.

In any case, we have:

$$\|z_k\| \leq \alpha^{-1}\left(\hat{\gamma}\left(V(z_0, m_0), k\right)\right) + \alpha^{-1}\left(\omega(c_3)\right).$$

Equation (4.16) implies that $V(z_0, m_0) \leq \alpha_2(\|z_0\|) + c_1$, which implies:

$$\|z_k\| \leq \alpha^{-1}\left(\hat{\gamma}\left(\alpha_2(\|z_0\|) + c_1, k\right)\right) + \alpha^{-1}\left(\omega(c_3)\right). \tag{4.25}$$

Then, we notice that, for any function $\alpha(s)$ of class $\mathcal{K}_\infty$, $\forall (s_1, s_2) \in \mathbb{R}_{\geq 0}$, the following holds:

$$\begin{aligned} \alpha(s_1 + s_2) &\leq \begin{cases} \alpha(2s_1), \text{ if } s_1 \geq s_2 \\ \alpha(2s_2), \text{ if } s_1 \leq s_2 \end{cases} \\ \Rightarrow \alpha(s_1 + s_2) &\leq \alpha(2s_1) + \alpha(2s_2). \end{aligned}$$

[1] Note that $\omega(c_3) = \alpha_4^{-1} \circ \rho^{-1}(\xi(c_1) + c_2)$, and if $c_1 = 0$, $c_2 = 0$, $\xi(0) = 0$, $\rho^{-1}(0) = 0$ and $\alpha_4^{-1}(0) = 0$, then $\omega(c_3) = 0$, but in this situation since $c_1 = 0$ and $c_2 = 0$, we are back in the case studied in [Jiang and Wang, 2001].

Since, for a given k, $\alpha^{-1}\left(\hat{\gamma}\left(s,k\right)\right)$ is a function of class $\mathscr{K}_\infty$, we have:

$$\alpha^{-1}\left(\hat{\gamma}\left(\alpha_2(\|z_0\|)+c_1,k\right)\right) \leq \alpha^{-1}\left(\hat{\gamma}\left(2\alpha_2(\|z_0\|),k\right)\right)+\alpha^{-1}\left(\hat{\gamma}\left(2c_1,k\right)\right).$$

The function $\alpha^{-1}\left(\hat{\gamma}\left(2c_1,k\right)\right)$ is of class $\mathscr{K}_\infty$, and thus going to 0 as $k\to\infty$, which means that it attains its maximum for $k=0$:

$$\alpha^{-1}\left(\hat{\gamma}\left(2c_1,k\right)\right) \leq \alpha^{-1}\left(\hat{\gamma}\left(2c_1,0\right)\right),\ \forall\ k\in\mathbb{Z}_{\geq 0}. \tag{4.26}$$

Equations (4.25)-(4.26) imply:

$$\begin{aligned}&\|z(z_0,m_0,k)\| \leq \gamma(\|z_0\|,k)+c\\ &\text{with } \gamma(s,k)=\alpha^{-1}\left(\hat{\gamma}\left(2\alpha_2(s),k\right)\right)\\ &\qquad c=\alpha^{-1}\left(\omega(c_3)\right)+\alpha^{-1}\left(\hat{\gamma}\left(2c_1,0\right)\right).\end{aligned}$$

Proof of Theorem 4.2

Now we want to prove that the system (4.13) is GISpS. First we will prove that the system (4.13) admits a GISpS-Lyapunov function. This part is proven in three steps, one for each of the inequalities in Definition 4.2. In a last step, using Theorem 4.1 we conclude that our system is practically stable. We take the Value Function V_H from Equation (4.3) as a candidate.

Step 1 - Proving that $\alpha_1(|z|)\leq V_H(z,m)$: We recall the expression of V_H, Equation (4.3):

$$V_H(z_k,m_k)=\min_{(\Pi,\pi)\in\mathscr{P}_H}\left\{z_k^\top\Pi z_k+\pi_{m_k}\right\}$$

By noticing that $\theta_{i,j}\geq 0$ for all $(i,j)\in\mathbb{M}^2$, we have:

$$V_H(z_k,m_k)\geq\min_{(\Pi,\pi)\in\mathscr{P}_H}\left\{z_k^\top\Pi z_k\right\}.$$

The Π matrices in $\mathscr{P}_H$ are symmetric positive definite for any $H\geq 0$, which implies that there exists a symmetric positive definite matrix $\underline{\Pi}$ which belongs to the set $\mathscr{P}_H$ (with an appropriate vector $\underline{\pi}$), such that:

$$z_k^\top\underline{\Pi}z_k\leq\min_{(\Pi,\pi)\in\mathscr{P}_H}\left\{z_k^\top\Pi z_k\right\}.$$

Then $\alpha_1(|z|)\leq V_H(z,m)$ holds with:

$$\alpha_1(|z|)=z^\top\underline{\Pi}z.$$

Step 2 - Proving that $V_H(z,m)\leq\alpha_2(|z|)+c_1$: While V_H defined by Equation (4.3) is the solution of the optimisation problem defined by Equation (4.2), we want to consider the solution of the same problem with the following constraint:

$$\mathscr{V}=\hat{\mathscr{V}}=(1,1,\ldots,1), \tag{4.27}$$

i.e. the constraint of transmitting at each sampling period. It is straightforward to see that the transition cost in the cost function results in a constant (depending on the initial radio mode). Considering the optimisation problem (4.2) with the constraint (4.27) is equivalent to considering the minimisation of following problem:

$$\begin{aligned}\hat{J}_{\hat{\mathscr{U}}}(z_k,m_k)&=z_{k+H}^\top Q_F z_{k+H}+\sum_{i=k}^{k+H-1}\hat{\ell}(z_i,m_i,u_i)\\ \hat{\ell}(z_k,u_k)&=z_k^\top Q_{CL}z_k+u_k^\top R_{CL}u_k.\end{aligned}$$

One can notice that the memory term $\tilde{u}_k$ is useless with this formulation, but we still consider it to keep consistency with the rest of our work. We denote $J^*|_{\hat{\mathscr{V}}}$ the optimal cost function of the problem

(4.2) under the constraint (4.27) and $\hat{J}^* = \min_{\hat{\mathscr{U}}} \hat{J}_{\hat{\mathscr{U}}}$. Then $J^*|_{\hat{\mathscr{V}}}(z_k, m_k) = \hat{J}^*(z_k, m_k) + \mathfrak{c}(m_k)$ where $\mathfrak{c}(m_k)$ is a constant depending on the radio mode at time k.

This problem lies in the field of standard Linear Quadratic (LQ) formulation, and the solution is well known to be (see [Bertsekas, 2005a, Section 4.1]):

$$\begin{aligned}
\hat{J}^*(z_k, m_k) &= z_k^\top \hat{\Pi}_H z_k \\
\hat{\Pi}_i &= \Phi_{CL}^\top \hat{\Pi}_{i-1} \Phi_{CL} - \hat{\kappa}_i^\top \hat{\Gamma}_{CL}^\top \Pi_{i-1} \Phi_{CL} + Q_{CL} \\
\hat{\Pi}_0 &= Q_F \\
\hat{\kappa}_i &= (\Gamma_{CL}^\top \hat{\Pi}_{i-1} \Gamma_{CL} + R_{CL})^{-1} \Gamma_{CL}^\top \hat{\Pi}_{i-1} \Phi_{CL} \\
\hat{u}^*_{k+i} &= -\hat{\kappa}_{H-i} z_{k+i} \\
\hat{\mathscr{U}}^* &= (\hat{u}^*_k, \hat{u}^*_{k+1}, \dots, \hat{u}^*_{k+H-1}).
\end{aligned}$$

We denote $\hat{\Pi}_H \triangleq \hat{\Pi}$. Now, one should notice that the set $\mathscr{P}_H$ considers all the possible combinations of $\mathscr{V}$, including the one which always transmit, *i.e.* $\hat{\mathscr{V}}$. Then, one can prove that there exists a vector $\hat{\boldsymbol{\pi}}$ such that $(\hat{\Pi}, \hat{\boldsymbol{\pi}}) \in \mathscr{P}_H$ and $J^*|_{\hat{\mathscr{V}}}(z_k, m_k) = \hat{J}^*(z_k, m_k) + \hat{\pi}_{m_k}$. This implies that, assuming that $H > 0$, $\hat{\Pi}$ is symmetric positive definite. Moreover, $\hat{\boldsymbol{\pi}}$ can easily be computed:

$$\hat{\boldsymbol{\pi}} = \begin{bmatrix} H\theta_{1,1} & \theta_{2,1} + (H-1)\theta_{1,1} & \dots & \theta_{N,1} + (H-1)\theta_{1,1} \end{bmatrix}.$$

We have shown the following:

$$J^*\Big|_{\hat{\mathscr{V}}}(z_k, m_k) = z_k^\top \hat{\Pi} z_k + \hat{\pi}_{m_k}. \tag{4.28}$$

$V_H = J^*$ being the minimum cost obtained with the optimal policy, any other policy gives a larger cost. This holds in particular with the policy $J^*|_{\hat{\mathscr{V}}}$:

$$\begin{aligned}
& V_H(z_k, m_k) \leq J^*\Big|_{\hat{\mathscr{V}}}(z_k, m_k) \\
\text{Equation (4.28)} \ & \Rightarrow V_H(z_k, m_k) \leq z_k^\top \hat{\Pi} z_k + \hat{\pi}_{m_k} \\
& \Rightarrow V_H(z_k, m_k) \leq z_k^\top \hat{\Pi} z_k + H\theta_{\max}
\end{aligned}$$

where $\theta_{\max} = \max_{(i,j)\in\mathbb{M}^2} \left\{\theta_{i,j}\right\}$. Then it holds that $V(z, m) \leq \alpha_2(|z|) + c_1$ with:

$$\begin{aligned}
\alpha_2(|z|) &= z^\top \hat{\Pi} z \\
c_1 &= H\theta_{\max}
\end{aligned}$$

Step 3 - Proving $\Delta V_H(z, m) \leq -\alpha_3(|z|) + c_2$: In this subsection, we will note $\mathscr{U} = \mathscr{U}_{k,H}$ to specify that the sequence start at the time index k over an horizon of length H (and identically for $\mathscr{V}_{k,H}$).
Let's consider the following policy:

$$\begin{aligned}
\mathscr{U}_{k+1,H} = \bar{\mathscr{U}}_{k+1,H} &\triangleq (u^*_{k+1}, u^*_{k+2}, \dots, u^*_{k+H-1}, \bar{u}_{k+H}) \\
\mathscr{V}_{k+1,H} = \bar{\mathscr{V}}_{k+1,H} &\triangleq (v^*_{k+1}, v^*_{k+2}, \dots, v^*_{k+H-1}, \bar{v}_{k+H}),
\end{aligned}$$

where the terms u_i^*, v_i^* are taken from the optimal policy from time k over an horizon H, *i.e.* :

$$\begin{aligned}
\mathscr{U}^*_{k,H} &= (u^*_k, u^*_{k+1}, \dots, u^*_{k+H-1}) \\
\mathscr{V}^*_{k,H} &= (v^*_k, v^*_{k+1}, \dots, v^*_{k+H-1}),
\end{aligned}$$

and $(\bar{u}_{k+H}, \bar{v}_{k+H})$ being any admissible policy that we choose on the form[2] $\bar{u}_{k+H} = -\bar{\kappa} z_{k+H} \triangleq -\begin{bmatrix} \bar{K} \\ \mathbf{0} \end{bmatrix} z_{k+H}$, $\bar{v}_{k+H} = 1$. The policy $(\bar{\mathscr{U}}_{k+1,H}, \bar{\mathscr{V}}_{k+1,H})$ over the horizon H may not be optimal, it holds

[2] It is easy to compute that, when choosing $\bar{v}_{k+H} \neq 1$, it is not possible to prove that $\Delta V(x) \leq -\alpha_3(\|z\|) + c_2$.

that:

$$\begin{aligned}
J^*\left(f_{v_k^*}(z_k,m_k,u_k^*)\right) &\leq J_{\bar{\mathcal{Y}}_{k+1,H},\bar{\mathcal{V}}_{k+1,H}}\left(f_{v_k^*}(z_k,m_k,u_k^*)\right)\\
J^*\left(f_{v_k^*}(z_k,m_k,u_k^*)\right) &\leq J^*\left(z_k,m_k\right) - \ell_{v_k^*}(z_k,m_k,u_k^*) - z_{k+H}^\top Q_F z_{k+H}\\
&\quad + \ell_{\bar{v}_{k+H}}(z_k,m_k,\bar{u}_{k+H}) + z_{k+H+1}^\top Q_F z_{k+H+1}\\
V_H\left(f_{v_k^*}(z_k,m_k,u_k^*)\right) - V_H\left(z_k,m_k\right) &\leq -\ell_{v_k^*}(z_k,m_k,u_k^*) - z_{k+H}^\top Q_F z_{k+H}\\
&\quad + \ell_{\bar{v}_{k+H}}(z_k,m_k,\bar{u}_{k+H}) + z_{k+H+1}^\top Q_F z_{k+H+1}\\
\Delta V_H(z_k,m_k) &\leq -\ell_{v_k^*}(z_k,m_k,u_k^*) - z_{k+H}^\top Q_F z_{k+H}\\
&\quad + \ell_{\bar{v}_{k+H}}(z_k,m_k,\bar{u}_{k+H}) + z_{k+H+1}^\top Q_F z_{k+H+1}\\
\Delta V_H(z_k,m_k) &\leq -\ell_{v_k^*}(z_k,m_k,u_k^*) - z_{k+H}^\top Q_F z_{k+H}\\
&\quad + z_{k+H}^\top Q_{CL} z_{k+H} + \bar{u}_{k+H}^\top R_{CL}\bar{u}_{k+H} + \theta_{m_{k+H},\bar{v}_{k+H}}\\
&\quad + z_{k+H}^\top\left((\Phi_{CL}-\Gamma_{CL}\bar{\kappa})^\top Q_F(\Phi_{CL}-\Gamma_{CL}\bar{\kappa})\right)z_{k+H},
\end{aligned}$$

since $z_{k+H+1} = (\Phi_{CL} - \Gamma_{CL}\bar{\kappa})z_{k+H}$. Assuming that assumption 4.1 holds, and using the definition of $Q'_F \succeq 0$ given in Equation (4.15), it yields:

$$\begin{aligned}
\Delta V_H(z_k,m_k) &\leq -\ell_{v_k^*}(z_k,m_k,u_k^*) - z_{k+H}^\top Q'_F z_{k+H} + \theta_{m_{k+H},\bar{v}_{k+H}}\\
&\leq -\ell_{v_k^*}(z_k,m_k,u_k^*) + \theta_{\max}\\
&\leq -z_k^\top Q_{v_k^*} z_k - u_k^{*\top} R_{v_k^*} u_k^* - \theta_{m_k,v_k^*} + \theta_{\max}\\
&\leq \begin{cases} -z_k^\top\left(Q_{CL} + \kappa^{*\top} R_{CL}\kappa^*\right)z_k - \theta_{\min} + \theta_{\max} & \text{if } v_k^* = 1\\ -z_k^\top Q_{OL} z_k - \theta_{\min} + \theta_{\max} & \text{if } v_k^* \neq 1 \end{cases}\\
&\leq -z_k^\top\left(Q_{CL} + Q_{OL} + \kappa^{*\top} R_{CL}\kappa^*\right)z_k - \theta_{\min} + \theta_{\max}\\
&\leq -z_k^\top\left(Q_{CL} + Q_{OL} + \kappa_{\min}^\top R_{CL}\kappa_{\min}\right)z_k - \theta_{\min} + \theta_{\max}
\end{aligned}$$

where $\kappa_{\min}$ verifies, $\forall(\Pi,\pi)\in\mathscr{P}_H$, $\forall z\in\mathbb{R}^{n+n_u}$:

$$\begin{aligned}
\kappa &= \left(R_{CL} + \Gamma_{CL}^\top \Pi \Gamma_{CL}\right)^{-1}\Gamma_{CL}^\top \Pi \Phi_{CL}\\
\text{and } z^\top \kappa_{\min}^\top R_{CL}\kappa_{\min} z &\leq z^\top \kappa^\top R_{CL}\kappa z.
\end{aligned}$$

We notice that $Q_{CL} + Q_{OL} + \kappa_{\min}^\top R_{CL}\kappa_{\min}$ is symmetric positive definite, then it holds that $\Delta V(z,m) \leq -\alpha_3(|z|) + c_2$ with:

$$\begin{aligned}
\alpha_3(|z|) &= z_k^\top\left(Q_{CL} + Q_{OL} + \kappa_{\min}^\top R_{CL}\kappa_{\min}\right)z_k\\
c_2 &= \theta_{\max} - \theta_{\min}
\end{aligned}$$

Step 4 We have proven that V_H is a GISpS-Lyapunov function for the system (4.13). Then according to Theorem 4.2, the system (4.13) is GISpS.

4.5 Finite horizon solution - General case

Now we focus on the general case, with additive Gaussian noise, a memoryless erasure channel and several transmitting modes, as described in Chapter 2.

4.5.1 Mathematical model for the closed-loop system

The mathematical model for the general case is the same as presented in Section 3.3.1. Only the optimisation problem differs since now we are considering a finite horizon.

Optimal problem

The optimisation problem, which is solved at each sampling time, consists in finding an optimal feedback sequence $\mathscr{U}^*_{k,H}$ and an optimal switching sequence $\mathscr{V}^*_{k,H}$ such that:

$$J^*(z_k, m_k) \triangleq J_{\mathscr{U}^*,\mathscr{V}^*}(z_k, m_k) = \min_{\mathscr{U},\mathscr{V}} J_{\mathscr{U},\mathscr{V}}(z_k, m_k), \tag{4.29}$$

where the cost function $J_{\mathscr{U},\mathscr{V}}(z_k, m_k)$ is given by:

$$J_{\mathscr{U},\mathscr{V}}(z_k, m_k) = \mathop{\mathsf{E}}_{\substack{\beta_k,\,\omega_k \\ k=0,1,\dots}} \left[z_{k+H}^\top Q_F z_{k+H} + \sum_{i=k}^{k+H-1} \ell_{\nu_i}(z_i, m_i, \hat{u}_i, \beta_i) \right].$$

We recall that $\mathscr{U} = \{\mu_0, \mu_1, \dots, \mu_{H-1}\}$, $\mathscr{V} = \{\eta_0, \eta_1, \dots, \eta_{H-1}\}$ and that $\hat{u}_k = \mu_k(z_k, m_k)$, $\nu_k = \eta_k(z_k, m_k)$, $z_{k+1} = f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k)$.

Similarly to the ideal case, it is worth noting that the problem is time invariant, which means that the control inputs and switching decisions depend only on the horizon step and the corresponding state. It is indeed sufficient to solve the problem with the state as a parameter.

4.5.2 Derivation of the optimal policy and implementation

Again, the joint optimal switching policy and feedback control law can be derived with Dynamic Programming. The Value Iteration method introduced in Section 2.5.2 provides the optimal cost function J^* and optimal sequences, $\mathscr{U}^*$, $\mathscr{V}^*$, which may not be unique. In the finite case, the Value Iteration method is just an application of Bellman's Principle of Optimality. It is given by the following recursion on the Value Function $V_i(z, m)$ over an horizon on length H:

$$\begin{aligned}
V_0(z, m) &= z^\top Q_F z \\
V_{i+1}(z, m) &= \min_{(\hat{u},\nu)\in\mathbb{U}(z)} \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_\nu(z, m, \hat{u}, \beta) + V_i(f_\nu(z, \hat{u}, \omega, \beta), \nu) \right] && i \in \{0, 1, \dots, H-1\} \\
(\mu_i^*(z, m), \eta_i^*(z, m)) &= \arg\min_{(\hat{u},\nu)\in\mathbb{U}(z)} \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_\nu(z, m, \hat{u}, \beta) + V_i(f_\nu(z, \hat{u}, \omega, \beta), \nu) \right] && i \in \{0, 1, \dots, H-1\} \\
J^*(z, m) &= V_H(z, m) && \forall (z, m) \in \mathbb{X} \\
\mathscr{U}^* &= \left\{ \mu_0^*, \mu_1^*, \dots, \mu_{H-1}^* \right\} \\
\mathscr{V}^* &= \left\{ \eta_0^*, \eta_1^*, \dots, \eta_{H-1}^* \right\}
\end{aligned}$$

In the general case the Value Function cannot be written in an explicit form as done in Equation (4.3) in the ideal case. Indeed, the stochastic formulation introduces expectations that complexify the computation. Then, the implementation of the algorithm relies on the discretisation scheme (using a grid), as explained in Section 3.2.3.

While the solution of the optimisation problem provides sequences on the switching decisions and the feedback control inputs over the horizon, the use of MPC implies that only the first elements of each sequence are applied to the system, and the procedure is repeated at the next sampling interval. This results in the optimal stationary joint policy (μ^*, η^*) given by:

$$\begin{cases} \mu^*(z_k, m_k) = \mu_0^*(z_k, m_k) \\ \eta^*(z_k, m_k) = \eta_0^*(z_k, m_k). \end{cases} \tag{4.30}$$

As done previously, the optimal strategy can be divided into an offline computation and an online computation. The optimal joint policy (μ^*, η^*) is computed offline. Then the following algorithm is run online on the smart node at each sampling time:
Compute the switching decision, i.e. *the next radio-mode,* $m_{k+1} = \nu_k^* = \eta^*(z_k, m_k)$ *and*

- *if* $\nu_k^* \in \mathbb{M}_1$*, compute the optimal control input* $u_k^* = \mu^*(z_k, m_k)$ *and send it to the actuator,*
- *if* $\nu_k^* \in \mathbb{M}_2$*, do nothing.*

This leads to an EBC law where the mode switching policy η triggers a control update based on the current state (z_k, m_k). Indeed, the space $\mathbb{X}$ is divided into regions whose crossing triggers an event, in a similar manner that threshold crossing triggers an event in works such that [Cogill, 2009, Heemels et al., 2013].

4.5.3 Use of the control sequence

The solution of the optimisation problem given in Equation (4.29) provides an optimal feedback sequence $\mathcal{U}^*_{k,H}$ and an optimal switching sequence $\mathcal{V}^*_{k,H}$. However, the MPC approach uses only the first elements of these sequences, as described by Equation (4.30). This is consistent with the use of a receding horizon. However, the purpose of the EBC scheme is to limit the amount of transmission, and then to run the system open-loop when the control performance is good enough in order to save energy. According to Equation (2.2), when a control update is not received by the actuator node (because it has been dropped by the erasure channel or because it has not been sent by the smart node), the actuator keeps applying the last control input.

Knowing that an optimal control sequence over an horizon H has been computed at the smart node side, one may argue that holding the control input at the actuator side is not the best choice when no update is received and when the current time subtracted to the time the last update was received is less than the horizon length. However, in order to use other elements of the control sequence at the actuator side, the smart node has to send the whole sequence, which implies a consequent energy consumption, which is precisely what we want to avoid. On the other hand, applying the other elements of the control sequence may results in a better open-loop performance, which may in turn extend the time spent without updating the control sequence. The benefits of such scheme will depend on the system.

Moreover the switched system as given in Equation (2.3) is not designed to take into account the use of the optimal control sequence when no update is received. To do so, the dimension of the switched system should be augmented to $H-1$ dimensions to keep the memory of the H control inputs in the sequence (note that the switched system is already augmented with the memory of the last control input). This increases the computation burden, which may be acceptable offline but not necessarily online.

4.6 Average cost horizon

In Section 4.1, we have motivated the use of a finite horizon. So far, we focused on the use of a receding horizon in order to apply the scheme on larger time interval than the horizon length. Another natural approach to meet the same goal is to consider the average of the cost function on the infinite horizon. However, this approach cannot be used with our system for the reasons explained in this section. The formulation of the cost function in the average case is given by:

$$J_{\mu,\eta}(z_0,m_0) = \lim_{H\to\infty} \frac{1}{H} \mathop{\mathsf{E}}_{\substack{\beta_k,\omega_k \\ k=0,1,\dots}} \left[\sum_{k=0}^{H-1} \ell_{v_k}(z_k, m_k, \hat{u}_k, \beta_k) \right],$$

where the evolution of the system is the same as previously. Note that the average cost does not consider neither final cost nor discount factor, and it relies on the infinite cost function.

4.6.1 Literature review

The average approach is discussed in [Bertsekas, 2007, Chap.4] in the case of a finite and countable state space. In the case that we are interested in (real vector state space), the author in [Bertsekas, 2007, Chap.4] says that the reference [Ros70] from [Bertsekas, 2007] gives counter examples where:

- there may not exist an optimal policy,
- there may exist an optimal non-stationary policy but not an optimal stationary policy.

Similarly to the discounted infinite case, and the finite case, one can establish the analog of Bellman's Equation for the average cost case. It is given in Proposition 4.2 taken from [Bertsekas, 2007, Proposition 4.2.1] which also provides a proof that we omit here.

Proposition 4.2. *Given a system $f(x,u)$ where $x \in \mathbb{X}$ and $u \in \mathbb{U}$, a policy μ such that $u = \mu(x)$, a cost function $J_\mu(x_0) = \lim_{H\to\infty} \frac{1}{H} \left\{ \sum_{k=0}^{H-1} \ell(x_k,u_k) \right\}$,*
if a scalar ρ^* and a function $h(x)$ satisfy:

$$\rho^* + h(x) = \min_{u\in\mathbb{U}} \{\ell(x,u) + h(f(x,u))\} \qquad \forall\, x \in \mathbb{X}, \tag{4.31}$$

then ρ^* is the optimal average cost per stage $J^*(x)$ for all x:

$$\rho^* = \min_{\mu} J_\mu(x) = J^*(x) \qquad \forall\ x \in \mathbb{X}$$

Furthermore, if $\mu^*(x)$ attains the minimum in Equation (4.31) for each x, the stationary policy μ^* is optimal, i.e. $J_{\mu^*}(x) = \rho^*$

$h(x)$ may have several interpretation, see *e.g.* [Bertsekas, 2005a, Chap 7.4] or [Bertsekas, 2007, Chap 4.1]. It can be seen as a relative cost, like the distance to ρ^* depending on (x).
Bellman's equation allows to prove that a given policy is optimal, however, not only it does not provide a way to compute an optimal policy, but also it does not ensure that such optimal policy exists. However, the Bellman's equation may inspire tools to derive the optimal policy, such as the Value Iteration method introduced in Section 2.5.
It has to be noticed that in Proposition 4.2, as stated in [Bertsekas, 2007], the state space $\mathbb{X}$ is finite and the system is deterministic. While our system is stochastic and considers real vector state space, our main difficulty is not to extend this proposition to our case, but to find a tool to derive the optimal policy with our setup.
Here follows several schemes to derive such optimal policy which fail in our case.

First, it can be noticed that the optimal policy exists and can be derived in the special case of LQ formulation as it is treated in [Bertsekas, 2007, Sec. 4.4]. However, the cost-to-go we consider, as defined in Equation (2.5), also accounts for the transmission energy in a non-quadratic fashion. This suffices to prevent the derivation of the optimal policy in our case as it is done in the LQ case.

The study of the average approach is mainly related to the discounted infinite case. [Bertsekas, 2007, Chap. 4.1] gives a relation between the discounted problem formulation and the average one. The idea is to look at the limit of the discounted case when the discount factor goes to 1. Then, a Backwell Optimal stationary policy is defined as a policy which is simultaneously optimal for all the λ-discounted problems with the discount factor λ in an interval $(\bar{\lambda}, 1)$ where $\bar{\lambda}$ is some scalar between 0 and 1. Proposition 4.2.2 in [Bertsekas, 2007] states that if a policy is Backwell Optimal, then it is also optimal for the average formulation problem.
However, in our case, an optimal policy for a given λ is not Backwell, since, when λ is changed, the corresponding optimal policy is changed as well.

The last scheme we discuss is a Value Iteration method that converges to the optimal average cost, as given in Theorem 4.3, which is taken from [Hernández-Lerma and Lasserre, 1996, Theorem 5.6.3].

Theorem 4.3. *Given a system $f(x,u)$ where $x \in \mathbb{X}$ and $u \in \mathbb{U}$, a policy μ such that $u = \mu(x)$, a cost function $J_\mu(x_0) = \lim_{H\to\infty} \frac{1}{H}\left\{\sum_{k=0}^{H-1} \ell(x_k, u_k)\right\}$,*
under given assumptions, the Value Iteration procedure, defined by:

$$\begin{cases} h_i(x) \triangleq J_i^*(x) - J_i^*(\tilde{x}) \\ \rho_i(x) \triangleq J_i^*(x) - J_{i-1}^*(x) \end{cases} \qquad \forall\ x \in \mathbb{X} \text{ and a fixed } (\tilde{x}) \in \mathbb{X}$$

where $J_i^*(x)$ is the optimal i-stage cost:

$$\begin{aligned} J_i^*(x) &= min_\mu J_{\mu,i} && \forall\ i \geq 1 \\ J_{\mu,i} &\triangleq \sum_{k=0}^{k=i} \ell_v(x) \\ J_{\mu,0} &\triangleq 0 && \forall\ (x) \in \mathbb{X} \end{aligned}$$

converges, that is, the following holds:

$$\begin{cases} \lim_{i\to\infty} h_i(x) = h(x) \\ \lim_{i\to\infty} \rho_i(x) = \rho^* \end{cases} \qquad \forall\ x \in \mathbb{X}$$

where ρ^* and $h(x)$ satisfy Proposition 4.2. Moreover the convergence is uniform on compact subsets of $\mathbb{X}$.

The idea of this procedure consists in computing the difference between a fixed point of the state space $\mathbb{X}$ and the optimal cost over an horizon i. By averaging around the fixed point at each iteration, the difference results in the function $h(z,m)$, the equivalent in the average strategy of the Value Function in the rest of this thesis.

We have omitted the assumptions needed to apply this theorem. We will not go through all of them as they appear in [Hernández-Lerma and Lasserre, 1996], but we will focus on the first one that prevents us from using this theorem (note that the other assumptions that are not met fail for the same kind of reason). Assumption 5.4.1b) from [Hernández-Lerma and Lasserre, 1996] is as follows:

Assumption 4.2. *There exist a constant $W \geq 0$ and a non negative measurable function $b(z,m)$ on $\mathbb{X}$ such that:*

$$-W \leq h_\lambda(z,m) \leq b(z,m) \qquad \forall x \in \mathbb{X} \text{ and } \lambda \in [\bar{\lambda}, 1)$$
$$\text{where} \qquad h_\lambda(z,m) \triangleq V_\lambda^*(z,m) - V_\lambda^*(\tilde{z},\tilde{m})$$

and $V_\lambda^*(z,m)$ is the optimal solution of the infinite horizon λ-discounted problem.

Note that the Value Function $V_\lambda^*(z,m)$ refers to the Value Function of the corresponding λ-discounted infinite horizon problem, *i.e.* $V_i(z,m)$ in Section 3.2.2 when i goes to infinity.
We do not satisfy this assumption because it relies on λ. Indeed, such constant $W \geq 0$ and non negative measurable function $b(z,m)$ can be found for a given value of λ, but not for a range of λ up to the upper bound 1.

This can be clearly seen when we consider our model in its deterministic formulation, namely when $w_k = \mathbf{0}$ and $\beta = 1$ for all $k \geq 0$. Indeed, under this condition, Section 3.2.3 proves that $V_\lambda^*(z,m)$ can be written on the form:

$$V_\lambda^*(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}^*} \left\{ z^\top \Pi z + \pi_m \right\},$$

as given in Equation (3.4) when $i \to \infty$. Then one can find $(\bar{\Pi}, \bar{\pi}) \in \mathscr{P}^*$ such that:

$$\frac{\theta_{\min}}{1-\lambda} \leq V_\lambda(z,m) \leq z^\top \bar{\Pi} z + \frac{\theta_{\max}}{1-\lambda} \qquad \forall (z,m) \in \mathbb{X}. \tag{4.32}$$

We denote $\theta_{\max} = \max_{(i,j)\in\mathbb{M}^2} \left\{\theta_{i,j}\right\}$ and $\theta_{\min} = \min_{(i,j)\in\mathbb{M}^2} \left\{\theta_{i,j}\right\}$. We notice that $\bar{\Pi}$ does not depends on λ. We omit the details on how to derive such $\bar{\Pi}$. Equation (4.32) yields:

$$\frac{\theta_{\min} - \theta_{\max}}{1-\lambda} \leq V_\lambda(z,m) - V_\lambda(\tilde{z},\tilde{m}) \leq z^\top \bar{\Pi} z + \frac{\theta_{\max} - \theta_{\min}}{1-\lambda}$$

The reason why Assumption 4.2 is not satisfied is that $\frac{\theta_{\max}-\theta_{\min}}{1-\lambda}$ diverges as $\lambda \to 1$. This prevents from finding a constant $W \geq 0$ and a non negative measurable function $b(z,m)$ that satisfy Assumption 4.2.

4.6.2 Further comments

The Value Iteration method fails because of the accumulation of transition costs. Indeed, this causes the Value Function to diverge along the infinite number of iterations. To tackle that, the Value Iteration method given by Theorem 4.3 subtract a scalar to the Value Function h_i, at each iteration. However in our case, this scalar cannot compensate for the accumulations related to the different costs for the different radio modes. In other words, for this scheme to converge, one would have to subtract a scalar depending on the radio mode. However this suppresses the whole interest of our scheme, since it results in compensating for the different transition costs. And having different costs for the different states of the radio is precisely what we want to account for.

Another variation of the scheme proposed in Theorem 4.3 could be to change the Value Iteration method in order to average the influence of the accumulated costs instead of subtracting a given quantity, at each iteration. One way to average is to divide by the number of iterations, but the iteration update should not rely on the iteration index directly. Another possibility is to use a discount factor on the update part of the accumulated cost, for instance:

$$\text{accumulated cost}_{i+1} = \lambda \text{accumulated cost }_i + \text{cost depending on the state}$$

with $0 < \lambda < 1$ and i the iteration index.
However, the problem formulated that way is the same as the infinite horizon discounted case, it only differs by a constant. Moreover, because the proof of convergence of the Value Iteration method in the average problem in [Hernández-Lerma and Lasserre, 1996] uses the results from the discounted case, by changing the update law of the Value Iteration, one would have to not only prove the convergence for the reformulated average case, but also for the reformulated discounted case.

As a last comment, it can be noticed that translating all the costs so that $\theta_{\min} = 0$ does not help.

4.7 Simulations

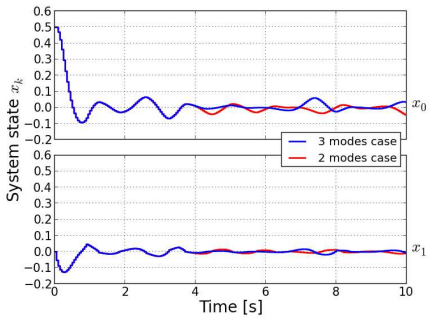
4.7.1 Ideal case

We simulate the control and radio-mode switching scheme on a second order plant. We focus on an unstable plant because energy-efficiency is more critical in the case where the system is not converging naturally. We consider the following parameters:

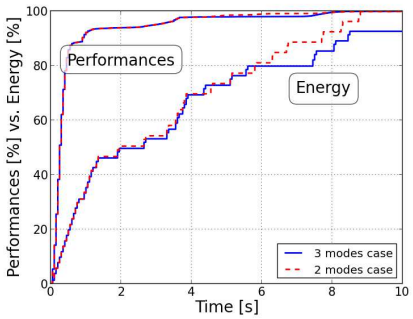
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$
$$\bar{Q} = 0.1\mathbf{I}\ ; \quad \bar{R} = 0.1\ ; \quad N = 3\ ; \quad H = 10$$
$$Q_F = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}\ ; \Theta = \begin{bmatrix} 2.9 & 1.8 & 1.9 \\ 3.2 & 1.4 & 60 \cdot 10^{-6} \\ 3.5 & 3.75 \cdot 10^{-3} & 60 \cdot 10^{-6} \end{bmatrix}$$

where the elements of the matrix Θ are the transition costs $\theta_{i,j}$, given in [mJ] and computed from the datasheet of the radio chip Texas Instrument CC1100 [Chipcon Products].

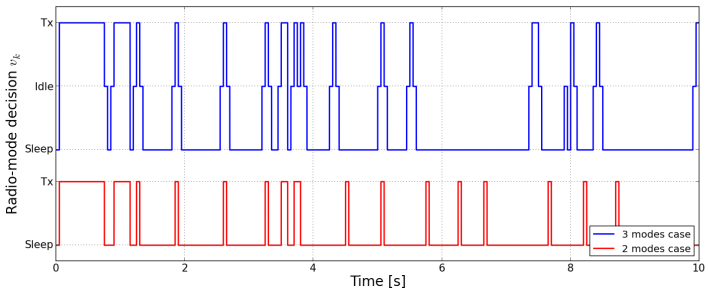
The plant is discretised with the sampling period $T_s = 0.05$s. We want to observe the energy savings obtained by the introduction of the low consuming mode `Idle` in between the `Tx` and `Sleep` modes.



(a) Output of the system x_k. The first component of each case is plotted in the top and the second in the bottom of the figure.



(b) Feedback performances $\sum_{i=0}^{k}(x_i^\top \bar{Q} x_i + u_i^\top \bar{R} u_i)$ (upper curve) and the accumulated energy consumption E_k. Curves are normalised to the largest value at the end of the simulation.



(c) Switching decision v_k.

Figure 4.1 *– Online simulations comparing the cases with 2 and 3 radio modes. The 3 modes case is plotted in blue and the 2 modes case in red.*

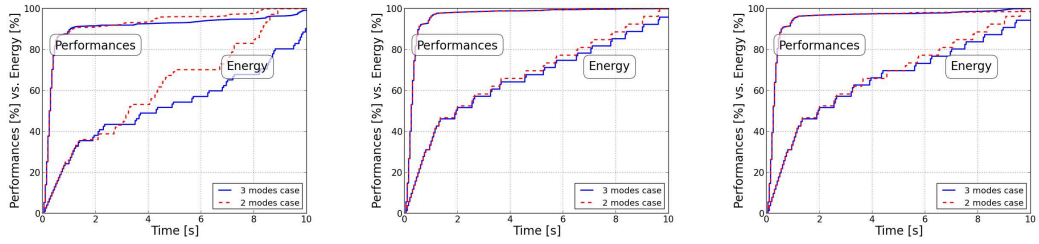
To this end, we compare the system described above where $N = 3$ with the case where $N = 2$ in an online simulation. We use the same parameters in the two cases expect for the transition cost matrix, which for the controller governing only 2 modes is given by:

$$\Theta' = \begin{bmatrix} 2.9 & 1.9 \\ 3.5 & 60 \cdot 10^{-6} \end{bmatrix}$$

The results are given in Figure 4.1. The system is stabilised in both cases, as depicted in Figure 4.1(a), with a similar control performance, as we can see that the upper curves of Figure 4.1(b) are almost the same.

It is important to note that the amount of energy consumed by the proposed controller governing the 3 modes is notably less than when only 2 modes are considered only after the system has reached the invariant set around its equilibrium point, as we can see on the lower curves of Figure 4.1(b). Finally Figure 4.1(c) gives the switching decisions for both cases. We noticed that the switching decisions consist in transmitting in order to reach the invariant set and then to keep the state within some boundaries given by the choices of the weighting factors of the cost function.

We also study the influence of the horizon length H over the performance and the energy consumption in both the 3 and 2 modes cases on Figure 4.2. It appears that the 3 modes case always consumes less energy than the 2 modes case, for performances that are similar. Nevertheless, the amount of energy saved is more important for lower horizon length.



(a) Horizon $H = 8$ (b) Horizon $H = 13$ (c) Horizon $H = 15$

Figure 4.2 *– Illustration of the influence of the horizon length H over the performance and energy consumption for both the 3 and 2 modes cases.*

4.7.2 General case

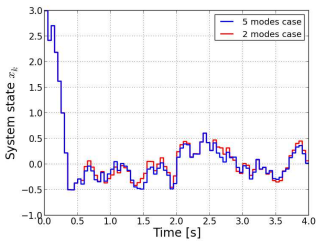
In the general case, we consider that the channel is not perfect, and control input updates may be dropped with a probability which depends on the transmission power level. In this subsection, we illustrate the use of several transmitting modes ($N_1 > 1$). We simulate our multi-mode EBC scheme on a first order unstable system. We consider three transmitting modes (denoted $\texttt{Tx}_1$ to $\texttt{Tx}_3$), and two low consuming modes (called `Idle` and `Sleep`):

$$x_{k+1} = 1.074x_k - 1.4808u_k + w_k; \quad T_s = 0.05\text{s};$$
$$\bar{W} = 0.02; \quad \bar{Q} = 0.01; \quad \bar{R} = 0.1; \quad Q_F = 1.5;$$
$$\Theta = \begin{bmatrix} 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.78 & 2.81 & 1.85 & 0.24 & 6\cdot 10^{-5} \\ 3.98 & 3.02 & 2.05 & 0.20 & 6\cdot 10^{-5} \end{bmatrix}; \quad \epsilon = \begin{bmatrix} 0.35 \\ 0.4 \\ 0.5 \end{bmatrix}$$

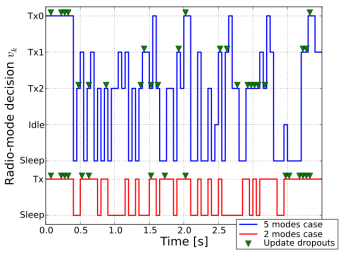
where the elements of the matrix Θ are the transition costs $\theta_{i,j}$, given in [mJ] and computed from the datasheet of the radio chip Texas Instrument CC1100, and where the elements of the ϵ vector are the dropout probabilities for each radio transmitting mode.

Figures 4.3(a)-4.3(c) give the state of the system x_k, the switching decision ν_k and the energy consumed and performance of our EBC scheme in the case where 5 radio-modes are used and in the case where the radio-modes are limited to $\texttt{Tx}_2$ and `Sleep`. We can see that the two state trajectories are very similar. The amount of transmissions and dropouts are similar but the amount of energy consumed by the 5 modes case is clearly less than the energy consumed in the 2 modes case. The use of several transmitting and low consuming modes allows to adapt the transmission decisions to

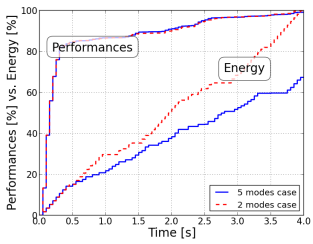
the actual need of the closed-loop performance. Critical transmissions use more power to reduce the failure probability while less needed updates are ignored or sent with smaller power level.



(a) State of the system x_k.



(b) Switching decision v_k.



(c) Feedback performances $\sum_{i=0}^{k}(x_i^\top \bar{Q} x_i + u_i^\top \bar{R} u_i)$ (upper curve) and the accumulated energy consumption E_k. Curves are normalised to the largest value at the end of the simulation.

Figure 4.3 *– A temporal simulation comparing our EBC scheme in the case where 5 radio modes are consider to the case with only 2 modes. The 5 modes case is plotted in blue and the 2 modes case in red.*

Conclusion

Networked Control Systems (NCSs) constitute a topic that has drawn much attention over the past ten years. Moreover, the use of wireless communication is particularly appealing in many contexts where it is difficult or constraining to use wires. Thanks to the reduction of the cost of electronic devices, one can now embed computation, communication and sensing units in a single piece of hardware, that we call *smart sensor* in this thesis. These little pieces of hardware are easy and cheap to produce, and they have opened the way to many challenging and interesting problems in a very wide range of applications, such as traffic control, remote surgery or industrial automation.

Energy saving and robustness to data drop became major challenges in wireless systems. These challenges have been addressed by the Control Community and the Communication Community. Although the interests of these two communities overlap, their approaches generally differ and the contributions in one field barely benefit to the other field. Indeed, the Communication Community is mainly interested in open-loop systems (monitoring) with a low-level description (generally the packet level) while the Control Community holds a strong theory taking into account perfect channel communication and considers the application level. As it is presented in the introduction, this thesis focus on the co-design of control and communication in the framework of NCS. It gives a particular focus on energy savings.

The purpose of Chapter 1 is precisely to understand how the energy is used in a NCS. It reviews some of the main schemes used to save energy in wireless nodes. This chapter covers both the control and communication communities. The review is organised in a multi-layer architecture. Indeed, it is natural to model the communication protocols using a stack of layers which correspond to different levels of abstraction. We use a four layer stack composed of the Physical layer, the Data Link (or Media Access Control (MAC)) layer, the Network (or routing) layer and the Application layer.
This chapter highlights that the efforts of the Communication Community concentrate on the low-level layers (Physical to Network layers) while the Control Community generally considers co-design with only two layers: the Application layer (in which the Control Theory lies) plus either the Physical or the Data Link layer, but never the Network layer.
It has to be noticed that there does not exist communication protocol dedicated to NCS. The notion of control and communication co-design implies that the parameters of the communication can be adapted by the control scheme. However, most of the protocols do not expose such parameters to the Application layer. While some contributions consist in designing such protocols, it is difficult to apply them in general control setups.
Finally, an important conclusion of Chapter 1 is that the use of control and communication co-design is essential to save more energy in a node, but it is not trivial.

After understanding how energy can be saved, the rest of the thesis focuses on a novel strategy which consists in a co-design between the Physical, Data Link and Application layer. It is commonly accepted that the transmission process is responsible for the largest energy consumption in a wireless node. Then, our strategy considers to turn off the radio at some time instants to save energy but this makes the system running open loop. Our strategy also exploits the capability of the radio chip to switch to low-consuming radio-modes. Indeed, when not transmitting, turning off all the components of the radio saves a lot of energy, but time and energy are needed to turn them back on in order to transmit again. A low-consuming radio-mode is a state of the radio where only some of the components are turned off. Less energy is saved when some components are still active, but less time and energy are needed to *wake up* the radio. This radio-mode management is usually addressed at the Data Link layer. Moreover, when transmitting, the radio can use different power levels to influence the success probability of the transmission over an erasure channel. Our strategy also considers to change

the power level of the radio, usually addressed at the Physical layer. The different power levels are considered as states of the radio, and treated as transmitting radio-modes.

The Chapter 2 is a descriptive chapter which introduces our strategy, the setup that we consider, and gives a general introduction on the procedures that are used in Chapters 3 and 4 to derive our strategy. Indeed, we focus on a setup with two nodes. The smart sensor node is in charge of computing the optimal radio-mode switching policy and the optimal feedback law that is only sent to the second node, the actuator node, when the performance of the control application is not good enough. A cost function is derived in order to trade-off the transmission energy and the control performance.

Our strategy, is a joint radio-mode switching policy and feedback control law. It is derived in the framework of optimal control with the use of a cost function. The cost function is considered over an horizon which is infinite in Chapter 3 and finite in Chapter 4. In both cases, the optimisation problem is solved using the framework of Dynamic Programming, and more precisely the Value Iteration method. In the infinite horizon case (treated in Chapter 3), the cost function must consider a discount factor in order to have an optimisation problem well posed. In this chapter, we first solve the optimisation problem in a deterministic case. While the standard arguments of Dynamic Programming allow to use the Value Iteration method as given in [Bertsekas, 2007], we derive a recursion to compute an explicit form of the Value Function. Indeed, the computation of this Value Function is the major difficulty of the Value Iteration method, and when no explicit form can be derived, one have to compute it along the points on a grid of a portion of the state space.
Then, we derive the Value Iteration method in the general case (namely, the stochastic case). In this case, we prove the convergence of the method, which is an important contribution of this thesis.
Chapter 3 is concluded with simulations showing that our strategy outperforms standard periodic approaches. We can also see that our strategy results in a stable closed-loop. Indeed, while the Value Iteration method ensures to provide an optimal solution, because of the use of a discount factor, the solution cannot be proved to be stable. However, stability is verified in simulation with an *a posteriori* analysis.

Finally, Chapter 4 addresses the same optimisation problem over a finite horizon. In order to use the solution in situations where the closed-loop system is run over time period larger than the length of the horizon, we consider the framework of Model Predictive Control (MPC), which uses a receding horizon. Again, the standard arguments of Dynamic Programming allow to use the Value Iteration method as given in [Bertsekas, 2005a]. As in the infinite horizon case, we are able to derive an explicit form of the Value Function in order to compute it exactly in the deterministic case. In the stochastic case, we are forced to use the discretisation scheme (using a grid).
An important contribution of this chapter is that we are able to prove the practical stability of our strategy in the deterministic case. Under the framework of Input-to-State Stability (ISS), we can prove that the system we considered closed with our joint switching policy and feedback law converges to a invariant set. Indeed, because our scheme consists in running open loop to save energy when the control performance is good enough, then when the system is open-loop unstable, it diverges when no transmission is scheduled. However, we prove that it will not leave a given ball around the origin.
Chapter 4 also discusses why the approach consisting in averaging the infinite horizon cannot be used in our context.
Finally, the chapter is concluded with simulations that show the energy saving that can be achieved with our strategy. It highlights that there exists a trade-off between control performance and energy saving.

To sum up, this thesis provides a review of the energy saving schemes that exist in the control and communication literature. It draws the important conclusion that co-design is essential to save important amount of energy. Then it provides a novel control strategy that benefits from control and communication advances. With the use of radio-modes, our strategy leads to an advanced Event-Based Control (EBC) scheme. The strategy can be applied on both infinite and finite horizon cases and the stability of the resulting closed-loop is proved in the finite deterministic case, and observed *a posteriori* in the other cases. However, many issues are still open, and some possible extensions are presented in the following.

Concerning the improvements and extensions of this work, first we can focus on the limitations of the results obtained in this thesis. As it has been discussed in Sections 3.3.4 and 4.5.2, when considering the stochastic model, no explicit form of the Value Function has been found. This forces to use the discretisation scheme (with a grid, as detailed in Section 2.5.4), which gives a solution on

the grid points only, with an important computation burden. Even though we have failed to extended to the stochastic case the explicit form that holds in the deterministic case, a possible improvement is to approximate the Value Function in a smarter way than the discretisation scheme. For instance the author in [Lincoln, 2003] is interested in Approximate Dynamic Programming, and gives interesting ideas to converge to a suboptimal cost function. Also, when considering the deterministic case and the explicit Value Function (in Sections 3.2.3 and 4.3.2), the computation burden of the explicit form is still heavy. Another improvement consists in deriving branching algorithms to speed up the computation, even at the price of obtaining a suboptimal solution.

Another direct improvement of the work in this thesis is to consider energy harvesting. Indeed, if the node is able to recover energy, it may change its behaviour depending on the amount of energy available. Actually, there are different ways to consider energy harvesting:

- If the device that recovers energy must have an exclusive access to the battery, then the node must choose between consuming the energy from the battery, or loading the battery. In this case, energy harvesting can be considered as another radio-mode, with appropriate transition costs and a negative cost to stay in this special mode (negative cost means that energy is recovered).
- If the node is able to gain energy while using the battery, one can think to add a penalty in the cost function on the transmission energy term (as defined in Equation (2.5)) which can be adapted as a function of the state of the battery. This would make all the transmission costs cheaper when a lot of energy is available, resulting in adapting the trade-off between the energy saving and the control performance to the state of the battery.
- If the node can also control the energy harvesting device, that adds another control variable which may be included in the cost function.

Among all the possible improvements that we could consider, we must notice that obtaining stability results in the infinite horizon case and in the stochastic finite horizon case is very important. However, in the infinite horizon case, as mentioned in Section 3.4.1, this is still an open issue in the field of Dynamic Programming. One of the main future directions is to consider Mean Square Stability to extend the practical stability to the stochastic finite horizon case.

We have used the framework of ISS to obtain our stability result. Originally, this framework is used not only to get stability but also to prove robustness. Indeed, ISS usually considers disturbance, and it ensures that as long as the disturbance stays bounded, the output of the system also stays bounded. It is then natural to extend our result to the disturbed case, obtaining a robustness result.

As discussed in Section 2.3.3, a very interesting extension concerns the use of quantisation to transmit the data. This allows to consider a more realistic channel, while ours is able to transmit real-valued messages. In addition to quantisation, one can also consider source coding to limit further the amount of data transmitted.

Another extension discussed in Section 2.3.3 is to consider self-triggered control. While EBC necessitates to sense the system periodically, self-triggered control let the system run open loop for a time period computed before opening the loop. It allows the node to turn off not only the radio but also the computation and the sensing units. Of course this makes the system more sensible to disturbances.

Finally one important extension, but maybe the most difficult, consists in considering several sensor nodes. This implies a lot of difficulties, *e.g.* waking up a sleeping node to relay a message in a multi-hop setup, or exchanging information between the sensors themselves. The radio-mode switching policies may used different approaches. Basically, either each node decides on its own radio state or a centralised controller tells each node when to sleep. The main drawback of the former case is that each sensor may not have sufficient information to take this decision in a way that is optimal for the whole system. In the latter case, the main problem consists in defining when the sensors should wake up. The centralised controller cannot base its decision on the current state of the system when part of the nodes are asleep, because it cannot retrieve the measurement from a sleeping node.

Appendix

This appendix contains a French summarisation of this thesis. It is only a preview of the content of the thesis, important topics that are not essential for understanding were intentionally omitted. The reader is referred to the rest of this document for a comprehensive discussion of the topic.

Cette annexe contient un résumé en Français de cette thèse. Il ne s'agit que d'un aperçu du contenu de la thèse, des sujets importants mais non essentiels pour la compréhension ont été volontairement omis. Le lecteur est renvoyé vers le reste de ce document pour une discussion complète du sujet abordé.

Contents

Introduction

Contexte et motivations

Les systèmes commandés en réseau (*Networked Control Systems*) sont des systèmes de contrôle où la communication entre les capteurs, les contrôleurs et les actionneurs se produit à travers un réseau, voir la Figure 1 et [Hespanha et al., 2007]. Le réseau peut être filaire ou sans fil. Ce type de configuration est une évolution naturelle de la commande discrète à distance, et est principalement destiné à décrire soit un système de contrôle où la boucle est bouclée par un réseau de communication ou un système composé de plusieurs capteurs et actionneurs qui partagent un réseau. Les Networked Control Systems (NCSs) peuvent être rencontrés dans un large éventail de domaines, tels que les systèmes de contrôle du trafic routier, les véhicules sous-marins autonomes, la chirurgie à distance, la production industrielle, les réseaux de capteurs, la distribution d'eau, ou les bâtiments intelligents.

Alors que l'automatique considère à l'origine les systèmes continus, la percée du numérique a apporté un grand intérêt pour les systèmes en temps discret. Cependant, bien que les systèmes discrets soient mieux adaptés aux problèmes de contrôle numériques réalistes, le domaine de la commande en temps discret est basé sur le paradigme que l'information exacte est instantanément transmise des capteurs au contrôleur et du contrôleur aux actionneurs. Les effets de diverses imperfections pratiques, tels que les retards, les perturbations, le bruit additif ou les saturations, ont été étudiés pour améliorer la qualité de la boucle fermée dans les systèmes réels. Toutefois, les imperfections causées par un réseau sont généralement beaucoup plus restrictives que celles envisagées dans le cas de systèmes dédiés filaires. Lorsque l'on considère un réseau, non seulement l'information peut souffrir de pertes de signal variables, mais aussi elle est quantifiée et généralement divisée en paquets de taille finie qui nécessitent une reconstruction côté récepteur. Ces restrictions sévères ne peuvent être abordées par les approches en temps discret existantes. Il est nécessaire d'entrer dans l'étude des problèmes de communication liés au réseau qui fait des systèmes commandés en réseau la fusion de la Théorie de l'Information avec la Théorie de la Commande, et apporte de nouveaux problèmes complexes.

Les NCSs ont gagné en intérêt au cours des dix dernières années pour les raisons suivantes :

- Les nœuds capteurs[1] sont des systèmes électroniques simples, produits en grande quantité, avec un coût de revient faible.
- L'utilisation d'un réseau (par opposition à un cablage dédié) facilite l'installation, la reconfiguration et la maintenance du système.
- Le système est plus flexible, il peut par exemple continuer de fonctionner en mode dégradé même si l'un des nœuds tombe en panne, ou on peut imaginer des configurations où le calcul de la loi de commande est déporté ou distribué.

Cependant, l'utilisation d'un réseau dans un système de contrôle présente également des questions généralement omises dans la Théorie de la Commande :

- La période d'échantillonnage est variable, car elle dépend des conditions du réseau qui peut souffir d'interférences, de congestions ou de pannes.
- Les transmissions peuvent subir des pertes de paquets.
- Les données sont très généralement quantifiées, et la granularité de cette quantification est liée à la bande passante offerte par le réseau, elle-même liée à l'énergie nécessaire à la transmission.

[1] On appelle un nœud capteur un nœud du réseau qui, en plus de ses capacités de mesure, est doté d'une unité de traitement (processeur et mémoires), d'une radio, et d'une alimentation.

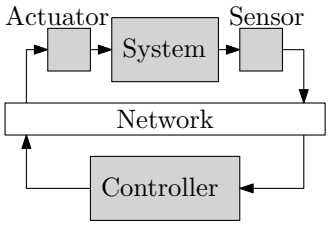


***Figure 1** – Schéma de principe typique d'un système commandé en réseau : le capteur envoie sa mesure au contrôleur sur le réseau, qui envoie la commande de l'actionneur sur le même réseau.*

- La Communauté de la Commande s'est intéressée à l'économie d'énergie au niveau de l'actionneur, alors que dans un nœud capteur sans fil alimenté par une batterie, l'énergie est limitée pour toutes les tâches (mesure, calcul, communication), pas seulement pour l'actionneur. L'économie d'énergie devient un enjeu majeur pour maximiser la durée de vie du système.

Dans cette thèse, nous nous concentrons sur la question de l'efficacité énergétique dans les NCSs sans fil. Même en présence de dispositifs de récupération d'énergie, l'utilisation de l'énergie doit être optimisée afin de maximiser la durée de vie des nœuds du réseau. L'objectif de cette thèse est d'identifier comment économiser l'énergie dans un nœud capteur sans fil et de proposer des lois de commande pour économiser l'énergie du nœud capteur.

Il est admis que l'énergie consommée par la radio représente 50% à 80% de la consommation globale d'un nœud capteur, voir par exemple [Joshi et al., 2007, Kimura and Latifi, 2005, Wen et al., 2007]. Une règle souvent citée est que l'exécution de 3 millions d'instructions exécutées par un processeur est équivalent à la transmission de 1000 bits à une distance de 100 mètres en termes d'énergie dépensée [Mann et al., 2007]. Il est donc intéressant d'optimiser l'utilisation de la radio, même si une petite quantité d'effort de calcul doit être ajoutée. L'objectif de la première partie de cette thèse est de comprendre en détails comment l'énergie est consommée par la transmission d'une radio, afin de proposer des méthodes pour réduire la consommation. Éteindre la radio n'est pas la seule façon d'économiser de l'énergie : comme on le verra après, on peut changer la modulation de la radio, la puissance de transmission, ou d'autres paramètres. Mais ces techniques influencent le système en boucle fermée et la gestion de la radio doit donc être prise en compte au niveau de la commande en faisant une co-conception, comme proposé dans cette thèse.

Contribution de cette thèse

Ce document est divisé en deux parties principales. L'objectif de la première partie (Chapitre 1) est de comprendre comment l'énergie est consommée dans un NCS et de présenter des techniques de contrôle de l'énergie du point de vue de la communication. La revue de la littérature est organisée selon une architecture de communication en couches couvrant, de bas en haut, les couches physique, liaison, réseau, et application.

La deuxième partie (Chapitres 2-4) étudie la commande événementielle économe avec la gestion de mode radio. En effet, d'une part les puces radio peuvent utiliser différentes puissances d'émission pour adapter la probabilité de réussite de la transmission. D'autre part elles offrent des modes de basse consommation qui permettent de limiter leur consommation sans pour autant complètement éteindre la puce. La Communauté de la Commande a déjà montré un grand intérêt pour le sujet de commande intermittente ou événementielle, qui permet d'éteindre la radio des nœuds sur des intervalles de temps plus longs que dans le cas périodique. Alors que la littérature existante ne traite que les politiques utilisant deux modes de radio (`Tx` – Transmission – et `Sleep` – veille –), cette thèse prend en compte plusieurs modes de transmission radio et plusieurs modes intermédiaires de faible consommation.
Nous dérivons un modèle linéaire commuté qui représente à la fois le système en boucle fermée et les modes radio. Nous définissons un critère de coût qui prend en compte à la fois la performance du contrôle et de la consommation d'énergie. Puis, nous concevons une loi commune de commande événementielle en boucle fermée et de gestion des modes radio pour contrôler efficacement le système et économiser de l'énergie du côté du nœud capteur.

Le Chapitre 2 introduit le système qui sera utilisé dans le reste de la thèse. Il fournit des commentaires généraux sur l'approche qui sera étudiée dans les chapitres suivants.

Le Chapitre 3 se concentre sur le problème à horizon infini. La loi commune est obtenue dans le cadre de la Commande Optimale, en utilisant la Programmation Dynamique.

Le Chapitre 4 considère le problème similaire avec un horizon fini. La solution est obtenue en utilisant le formalisme de Model Predictive Control (MPC) et encore la Programmation Dynamique.

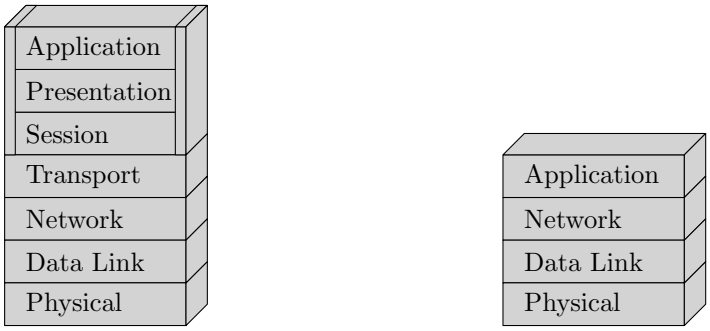
Dans les deux cas, infini et fini, la stabilité est discutée (mais seulement prouvée dans le cas fini avec l'utilisation de Input-to-State Stability (ISS) et les résultats sont illustrés par des simulations.

1 Architecture multi-couches dans les systèmes commandés en réseau sans fil

1.1 La pile à quatre couches

La consommation d'énergie peut être envisagée à différents niveaux du réseau. Au niveau du nœud, les paramètres de communication peuvent être réglés pour économiser l'énergie. Au niveau du réseau, les techniques d'ordonnancement des flux de communication permettent de maximiser la durée de vie de l'ensemble du réseau. Les réglages des nœuds et du réseau ont des implications majeures sur la commande, où certaines contraintes doivent être respectées pour garantir la stabilité et la performance du contrôle. Une approche multi-dimensionnelle est présentée dans cette section afin de déterminer comment l'énergie est utilisée dans un réseau sans fil et quelle est l'influence des techniques d'économie d'énergie existantes sur la boucle de régulation.

Un modèle en couches est couramment utilisé dans la Théorie de la Communication pour décrire les processus qui entrent en jeu dans une communication en réseau. Le standard *Open System Interconnection* (OSI) est composé de sept couches, comme représenté sur la Figure 2(a). Dans le cadre des NCSs, une pile comprenant seulement quatre couches est plus pertinente, et nommée ici la pile NCS. Elle est donnée dans la Figure 2(b) et détaillée ci-après.



(a) La pile OSI est la pile la plus complète, comprenant sept couches. Elle est utilisée par exemple pour décrire la communication en réseau entre ordinateurs.

(b) La pile NCS se limite à quatre couche, elle est mieux adaptée pour décrire la communication entre nœuds d'un NCS.

***Figure 2** – Les différentes étapes qui entrent en jeu dans une communication en réseau sont communémment abstraites avec une approche par couches, regroupées dans une pile, où chaque couche a une fonction spécifique. Le nombre de couche dans la pile dépend du type de communication qu'elle décrit.*

Le but de ce chapitre est d'examiner l'architecture multi-couches des NCSs à la lumière de l'utilisation de l'énergie dans chaque couche, et de souligner des contributions majeures dans le domaine des lois de gestion énergétique, couche par couche.

1. **La couche physique**[2]**.** Cette couche est en charge de la modulation radio des données numériques. Il s'agit des processus d'émission et de réception au niveau de la radio.
2. **La couche liaison.** Dans les communications sans fil, le support de transmission est partagé entre tous les nœuds, et il n'y a pas de voie dédiée entre une source et une destination. Cela implique que plusieurs nœuds du réseau émettant en même temps interfèrent les uns avec les autres. La couche liaison définit comment utiliser et partager le support de transmission.
3. **La couche réseau.** Dans les réseaux avec plusieurs nœuds, il peut y avoir plusieurs possibilités d'atteindre une destination à partir d'une source. Le rôle principal de la couche réseau est d'acheminer les données vers la destination d'une manière économe en énergie afin de maximiser la durée de vie de l'ensemble du réseau.
4. **La couche application.** Cette couche concerne l'encodage et le décodage de la source, et l'application finale, c'est-à-dire dans notre cas le calcul de la loi de commande.

[2]Nous rappelons que nous nous concentrons sur les réseaux sans fil.

1.2 La couche Physique

La couche physique concerne le matériel, c'est-à-dire la puce radio, utilisé pour transmettre un flux de bits sur le réseau. Elle spécifie les paramètres de bas niveau de la transmission, tels que le schéma de modulation, la fréquence, la période et la phase du signal ou la puissance d'émission. Il n'y a aucune considération à propos des paquets, de la correction d'erreurs ou de fiabilité à cette couche. Deux paramètres principaux influencent la consommation d'énergie : la puissance d'émission et le débit binaire, utilisés pour moduler les données dans les ondes radio ; ces deux paramètres peuvent être contrôlés.

Le contrôle de la puissance

La puissance d'émission est généralement liée à la fiabilité de la communication. L'augmentation du niveau d'énergie diminue la probabilité d'effacement, jusqu'à un certain niveau où les interférences doivent être prises en compte. La modification de la puissance de transmission a deux objectifs principaux : le premier est de faire face à différentes conditions de canal, le second est d'augmenter la probabilité de réussite des transmissions de données critiques. Dans le premier cas, plus d'énergie est consommée afin de maintenir la même qualité de la liaison alors que le débit du canal s'affaiblit (en raison d'obstacles en mouvement ou de perturbations ou de déplacement des nœuds). Dans le deuxième cas, plus d'énergie est consommée dans le but d'assurer la transmission de données critiques dans les mêmes conditions de transmission. Cependant, une puissance arbitrairement grande va induire des interférences avec d'autres liens de communication et une consommation d'énergie non modérée. La consommation d'énergie doit être équilibrée pour prolonger la vie de la batterie tout en assurant des transmissions correctes. Dans ce but, le contrôle de la puissance peut s'appuyer sur les observations du récepteur qui sont renvoyés à l'émetteur, ou sur les conditions du canal mesurées par l'émetteur.

Le contrôle du débit binaire

Le contrôle de la puissance est principalement utilisé pour réguler la qualité de transmission en cas de perte de paquets et d'interférences du signal. Le contrôle du débit binaire, parfois appelée Dynamic Modulation Scaling (DMS) en Anglais, est une alternative à la régulation de la fiabilité de la transmission sans augmenter la puissance d'émission, au prix de l'augmentation de la latence de transmission. DMS est une méthode qui consiste à commuter les caractéristiques de modulation (à puissance constante), pour améliorer la réception du signal.

1.3 La couche Liaison

La couche liaison (Media Access Control (MAC)) définit comment accéder au support de transmission, partagé entre tous les nœuds. Elle réglemente la communication en fournissant un moyen d'éviter les transmissions simultanées ou de se remettre des collisions de transmission. Elle est en charge de l'adressage physique et elle est capable de détecter et de corriger les erreurs et d'envoyer des accusés de réception. Cette couche est également chargée de contrôler l'état de la puce radio.

Les deux notions de base pour économiser l'énergie à cette couche sont soit d'utiliser un gestionnaire des modes d'activité, soit de contrôler le service de livraison de paquets via le réglage des paramètres du protocole MAC (comme par exemple la taille des *slots* de communication, les temps de sommeil et de réveil, ou le nombre et la durée des *backoffs*).

La gestion des modes d'activité

Les modes d'activité sont les états d'un nœud (ou de la puce radio dans le nœud). Alors que les deux modes de base sont `On` (marche) et `Off` (arrêt), on peut considérer des modes intermédiaires où seules des parties du nœud (ou de la puce radio) sont actives. La commutation entre les modes consomme à la fois du temps et de l'énergie (voir la Figure 3). Un compromis doit être fait entre la conscience du nœud et la consommation d'énergie.

On peut opposer ici les approches des deux Communités de la Communication et de la Commande. La Communauté de la Communication s'intéresse en priorité à la puce radio (laissant les autres parties du nœud, comme le processeur, gérer elles-mêmes leur énergie). Elle utilise des méthodes d'ordonnancement pour déterminer quand, combien de temps et à quelle profondeur la puce radio doit dormir. Ces méthodes sont en général basées sur le nombre de paquets qui doivent être envoyés sur le réseau

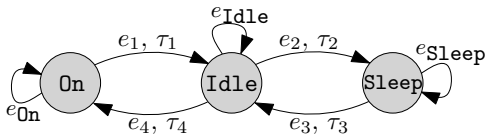


***Figure 3** – Illustration d'un automate qui décrit les commutations entre trois modes d'activité. Un coût énergétique est consommé pour commuter entre les modes (e_i, $i \in \{1,2,3,4\}$) ou pour rester dans un mode donné ($e_{\texttt{ON}}$, $e_{\texttt{Idle}}$, $e_{\texttt{Sleep}}$). Les coûts de transition temporel (τ_i, $i \in \{1,2,3,4\}$) sont seulement nécessaires pour commuter dans des modes différents du mode actuel.*

ou la probabilité qu'un événement (par exemple une perturbation) se produise plutôt que sur les exigences du système de contrôle. Les auteurs dans [Brownfield et al., 2010] envisagent plusieurs modes de faible consommation : `Idle`, `Power Down` et `Power Off`, en plus des modes de transmission et de réception, `Transmit` et `Receive`, comme détaillés dans le Tableau 1. Cela permet d'équilibrer les économies d'énergie et la réactivité lorsque les périodes de sommeil sont trop courtes pour éteindre complètement le nœud.

	Synthétiseur de fréquence	**Oscillateur**	**Régulateur de tension**
Transmit	Allumé	Allumé	Allumé
Receive	Allumé	Allumé	Allumé
Idle	Eteint	Allumé	Allumé
Power Down	Eteint	Eteint	Allumé
Power Off	Eteint	Eteint	Eteint

***Tableau 1** – Détail des modes radio de [Brownfield et al., 2010]. Les modes radio ne concernent que la puce radio d'un nœud.*

Concernant la Communité de la Commande, bien qu'elle prenne en compte le fait que la radio peut être éteinte alors que le nœud continue à relever des mesures, elle ne considère que deux modes d'activité, actif ou inactif. Dans le cas où c'est le nœud entier qui s'éteint, il décide, avant de s'éteindre le temps pendant lequel il sera indisponible. On parle alors de *self-triggered control*, voir par exemple [Anta and Tabuada, 2010]. Dans le cas où c'est juste la radio qui est éteinte, il décide de la rallumer lorsqu'il détecte un événement qui mérite d'être envoyé au contrôleur, et on parle ici de *event-triggered control*, voir par exemple [Cogill, 2009].

Ajustement du protocol MAC

Les paramètres du protocole MAC peuvent être adaptés pour équilibrer la fiabilité et la latence afin de répondre aux exigences de contrôle et de minimiser la consommation d'énergie. Les protocoles pris en compte dans cette section couvrent à la fois la couche physique et la couche liaison, et parfois aussi la couche réseau (routage). Comme les auteurs dans [Liu and Goldsmith, 2003] le concluent, il n'y a pas de protocole standard dédié aux NCSs. Les efforts faits jusqu'ici ont consisté soit à utiliser des protocoles standards (tels que 802.15.4, 802.11, ZigBee, Bluetooth, B-MAC, X-MAC) pour exploiter les paramètres réglables qu'ils ont à offrir [Fischione et al., 2009a], soit à développer de nouveaux protocoles dédiés aux NCSs, comme Breath [Park et al., 2008] ou TrEND [Di Marco et al., 2010].

1.4 La couche Réseau

La couche réseau (routage) est chargée de choisir un chemin (c'est-à-dire une série de nœuds relais) dans le réseau pour transmettre des paquets entre l'émetteur et le récepteur, d'une manière économe en énergie.

Routage économe en énergie

La littérature regorge de protocoles de routage pour les Wireless Sensor Networks (WSNs). Une étude complète peut être trouvée dans [Al-Karaki and Kamal, 2004] où les auteurs distinguent les protocoles basés sur la structure du réseau de ceux fondés sur le fonctionnement même du protocole. En termes généraux, les protocoles basés sur la structure du réseau utilisent la topologie du réseau et la quantité d'énergie disponible à chaque nœud pour choisir un itinéraire. Cependant ces protocoles ne proposent

pas de paramètres ajustables aux couches supérieures et donc ne sont pas intéressants dans le point de vue des NCSs.

Codage réseau

Dans les protocoles de routage discutés précédemment, les nœuds relaient juste les paquets qu'ils reçoivent. Dans le cadre du paradigme de codage réseau, les nœuds peuvent effectuer certaines opérations sur les données des paquets qu'ils reçoivent avant de les transmettre. Ceci est généralement illustré par l'exemple du réseau papillon [Yeung, 2008].

Le but du codage réseau est de maximiser le débit du réseau. Ceci est obtenu en ignorant les informations redondantes, ou en mettant en œuvre des calculs sur les données reçues à partir des nœuds voisins. Les calculs utilisés peuvent être une moyenne, un minimum ou maximum ou des combinaisons linéaires de ces derniers. L'agrégation de données (ou la fusion de données) se réfère à des réseaux où certains nœuds effectuent ces calculs tout en collectant des données à partir des nœuds voisins.

1.5 La couche Application

La couche application comprend la partie exécutive de la détection et de la commande, qui peut être divisée en deux catégories ; le nœud capteur est en charge de la détection et du traitement du signal, et le nœud contrôleur décode l'information reçue, et calcule l'action de commande. La première partie de cette section présente des travaux en rapport avec le codage source économe en énergie du coté du nœud capteur, et la seconde partie aborde la couche application du coté du nœud contrôleur.

Codage source

Alors que le codage réseau vise à maximiser le débit, le codage source a deux objectifs principaux. Le premier consiste à convertir la mesure analogique du capteur en signal numérique (sauf si le capteur fournit déjà un signal numérique). Le second consiste à compresser les informations, afin de réduire le taux d'occupation du canal de transmission. Le but des procédés de compression est principalement de réduire la quantité totale de bits et, dans certains cas, le coût par bit associé à la transmission.

Travaux fondamentaux Les limites théoriques du codage source ont été données dans le célèbre ouvrage de Shannon, [Shannon, 1948]. Dans le cas particulier des NCSs, nous ne sommes pas seulement intéressés par le nombre minimum de bits qui devraient être considérés pour décrire la source, mais aussi pour stabiliser le système en boucle fermée, ou pour atteindre d'autres indices de performance de contrôle. Une vue d'ensemble des procédés de contrôle sous contraintes de débit est donnée dans [Nair et al., 2007].

Codage source distribué Dans le cas où plusieurs sources fournissent de l'information, d'autres approches peuvent être considérées. Par exemple, le théorème Slepian-Wolf [Slepian and Wolf, 1973] permet de réduire la quantité de données circulant sur le réseau (et donc d'économiser de l'énergie). C'est le but du codage source distribué, qui exploite le fait que les mesures effectuées par des nœuds voisins sont corrélées.

Codage source uniforme La stratégie de codage dépend de la répartition des coûts de transmission. Le codage uniforme (en énergie) suppose que les coûts de transmission sont répartis uniformément entre tous les mots de code, ou de façon équivalente que les bits "0" et "1" sont associés à un coût de transmission d'énergie égal. Dans ce contexte, la stratégie de codage consiste à tenter de réduire le nombre total de bits que les mots de code vont utiliser sans aucune autre considération sur la nature de la valeur des bits eux-mêmes, voir par exemple [Canudas De Wit et al., 2009], [Jaglin et al., 2008].

Codage source non-uniforme Le codage source non-uniforme (en énergie) suppose que les coûts de transmission ne sont pas répartis uniformément entre tous les mots de code, c'est-à-dire que certains mots de code (ou bits) sont plus coûteux que d'autres. Ces différences peuvent être décrites en supposant une distribution de probabilité particulière, ou en changeant la distribution via le procédé bien connu de *transform coding*, voir [Goyal, 2001]. Des travaux dans cette direction peuvent être trouvés dans [Canudas De Wit et al., 2007a], [Canudas De Wit et al., 2007b] ou [Canudas De Wit and Jaglin, 2009].

Procédés de contrôle en réseau

La nature distribuée des nœuds capteurs et de contrôle dans le cadre d'un NCS fait que chacun de ces composants fonctionne avec sa propre horloge et réagit de manière asynchrone aux événements. Cela oblige les nœuds du réseau, soit à effectuer une synchronisation, soit à opérer de manière asynchrone. Le problème de synchronisation d'horloge peut être abordé, par exemple, avec un consensus [Carli et al., 2011]. Avoir des nœuds synchronisés permet d'utiliser les modèles standards de contrôle synchrone, même si ces méthodes doivent être adaptées aux limitations du réseau (tels que le retard, la bande passante limitée, la perte de paquets).

Approche du délai variable Une approche naturelle dans les NCSs est de considérer que les mesures sont reçues avec un retard variant dans le temps. Cette approche est une technique qui a reçu une grande attention pour régler le problème de la communication asynchrone dans un réseau. Elle est utilisée pour modéliser les arrivées asynchrones, au moment t_k, des mesures au niveau du nœud de contrôle [Fridman et al., 2004]. Le délai variant dans le temps est alors modélisé comme suit :

$$\tau(t) = t - t_k, \quad 0 \leq \tau(t) \leq h$$

où h est l'intervalle de temps maximal entre deux mesures. L'arrivée asynchrone de l'information obtenue peut donc être modélisée par : $x(t_k) = x(t - \tau(t))$.

Envoyer zéro, maintenir ou compenser Une autre approche naturelle pour faire face aux arrivées de mesures asynchrones du coté du contrôleur est de mettre en œuvre un système de contrôle périodique qui utilise les mesures quand elles sont disponibles et qui utilise un autre comportement quand aucune nouvelle mesure n'est reçue. Parmi les stratégies en boucle ouverte possibles, celles les plus rencontrées sont les suivants : envoyer zéro (appliquer une commande de valeur zéro au système), maintenir la dernière commande, ou compenser (par exemple en utilisant un prédicteur de l'état du système). Même si on peut penser que maintenir la commande donne de meilleures performances en boucle fermée que d'envoyer zéro, l'auteur dans [Schenato, 2009] montre que ce n'est pas toujours vrai.
Des exemples de stratégies de compensateur sont donnés dans [Bernardini and Bemporad, 2008], [Gommans et al., 2012].

Paradigme asynchrone - Le contrôle basé sur les événements Le contrôle basé sur les événements (en Anglais *Event Based Control* (EBC) ou aussi appelé *Event Triggered Control*) est un nouveau paradigme dans la Théorie de la Commande qui apporte de nombreux avantages et des défis. Il permet de gérer les arrivées de mesures asynchrones et réduit la bande passante sur le réseau.

L'analyse et la conception de système de contrôle, historiquement en temps continu, se sont étendus au temps discret à cause de leur utilisation pratique dans un monde numérique. Mais le contrôle en temps discret suppose des capteurs et un contrôleur parfaitement synchrones, ce qui est une contrainte difficile à satisfaire en présence d'un réseau. Le contrôle basé sur les événements propose de discrétiser les signaux continus, non pas dans le domaine temporel, mais dans le domaine des valeurs, comme le montre la Figure 4.

Cette approche est bien adaptée à des configurations où les tâches de détection et de contrôle sont effectuées dans les nœuds séparés. Le contrôle asynchrone à haut rendement énergétique repose alors sur la réduction de la communication nécessaire pour mettre à jour la loi de commande. Des systèmes de détection efficace en énergie doivent être mis en œuvre dans les nœuds capteurs afin de décider quand envoyer de nouvelles mesures pour assurer la stabilité en boucle fermée et une performance de contrôle donnée. Ces stratégies sont souvent basées sur des événements spécifiques à l'application. Le contrôleur doit être conçu pour gérer des arrivées de

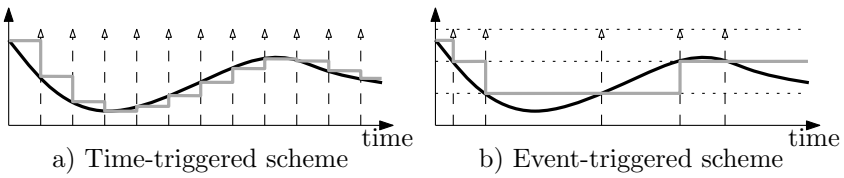


***Figure 4** – Comparaison du contrôle basé sur les coups d'horloge et sur les événements ; les flèches blanches indiquent les instants d'échantillonnage.*

mesures asynchrones et des accusés de reception peuvent être nécessaires dans le cas de canal avec pertes.

Les travaux dans [Åström and Bernhardsson, 2002] et [Otanez et al., 2002] donnent les bases du contrôle basé sur les événements, aussi appelé dans le deuxième papier *deadband control*. Le contrôle basé sur les événements est appliqué à une commande par retour d'état dans [Cogill, 2009], ou à une commande Proportionnelle Intégrale (PI) dans [Rabi and Johansson, 2008].

2 Formulation du problème et approche de commande

2.1 Modèle mathématique pour le système en boucle fermée

Nous considérons le problème de l'efficacité énergétique dans un système contrôlé en réseau sans fil (NCS). Nous limitons notre attention sur une configuration composée de deux nœuds, comme le montre la Figure 5 et décrit ci-après. Une configuration à deux nœuds saisit les enjeux de l'efficacité énergétique, sans introduire les difficultés qui apparaissent dans une configuration multi-nœuds (tels que le contrôle d'accès au média ou le routage).
Le premier nœud, appelé le nœud capteur intelligent, a des capacités de détection et de calcul. Il est chargé de détecter la sortie du système, de calculer de la loi de commande et de décider d'envoyer ou non la commande vers le deuxième nœud, chargé d'appliquer la loi de commande sur l'actionneur. On suppose que le nœud récepteur est co-localisé avec l'actionneur, et donc il a accès à une source d'énergie illimitée. Cette configuration, appelée aussi *one-channel feedback NCS*, est très répandu dans la littérature [Hespanha et al., 2007]. Ainsi, nous sommes seulement intéressés par les économies d'énergie du côté du nœud intelligent.

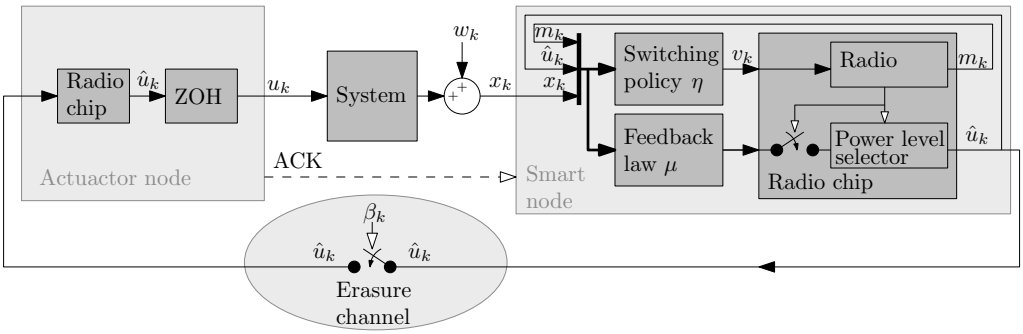


***Figure 5** – Diagramme de blocs du problème. Le nœud intelligent mesure l'état bruité du système, x_k, il calcule la commande $\hat{u}_k$ et décide de l'envoyer ou non au nœud actionneur. $\hat{u}_k$ peut être $\varnothing$ si aucune transmission n'est prévue ou lorsque le message transmis est perdu. Le récepteur est capable de déterminer si la commande a été reçue ou non. β_k est égal à 1 lorsque la transmission est réussie ou 0 en cas de perte.*

A cette fin, nous combinons les deux techniques. Tout d'abord, les communications du capteur intelligent vers le nœud récepteur sont basées sur les événements (*Event-Based Control* EBC). Cela signifie que le capteur intelligent peut décider de laisser le système fonctionner en boucle ouverte lorsque la performance du contrôle est assez bonne. Nous définissons dans la Section 2.3 une fonction de coût qui est utilisée par le capteur intelligent pour déterminer quelle est la qualité de la performance de commande par rapport à l'énergie nécessaire pour une transmission. Lorsque le nœud de réception ne reçoit pas de nouvelle mise à jour à partir du capteur intelligent, il continue d'appliquer la mémoire de la dernière commande reçue. La mémoire est mise en œuvre dans un dispositif de bloqueur d'ordre zéro dans la Figure 5. La mise à jour de la commande calculée du côté capteur intelligent, notée $\hat{u}_k$, peut être différente de la commande réellement appliquée au système, notée u_k. Les deux grandeurs $\hat{u}_k$ et u_k prennent des valeurs dans $\mathbb{R}^{n_u}$.
Ce comportement basé sur les événements est considéré sur la base d'une surveillance en temps discret. Cela signifie que le capteur intelligent surveille le système à une période d'échantillonnage donnée. Ensuite, en fonction de l'état du système, il décide si oui ou non une transmission se fait à l'instant courant. Ce dernier point est important, comme le soulignent les auteurs dans [Heemels et al., 2013], car il s'affranchit d'une contrainte largement imposée dans le contrôle basé sur des événements qui

suppose que les événements sont surveillés en temps continu, alors que seule une surveillance en temps discret peut être mise en œuvre en pratique.

Ensuite, la deuxième technique considérée conjointement avec l'approche EBC est la gestion des modes radio. En effet, non seulement le nœud intelligent décide d'utiliser ou non la radio, mais il décide aussi du mode utilisé par la radio pour la transmission ou l'oisiveté. En effet, lorsque la radio est active, un niveau de puissance d'émission est choisi pour faire une transmission. Quand elle n'est pas active, elle passe à l'un des nombreux modes radio à faible consommation. La radio peut utiliser différents niveaux de puissance pour changer la probabilité de réussite de la transmission dans le cas de conditions de canal inchangées. En cas de canal variable, la puissance d'émission peut être augmentée pour faire face à de mauvaises conditions de canal et pour maintenir constante la probabilité de réussite de la transmission. Nous définissons N_1 comme le nombre de niveaux de transmission. Les modes à faible consommation (par exemple `Idle`, `Sleep`) permettent d'économiser l'énergie en désactivant certains éléments de la puce radio. Cependant, un certain temps et une certaine énergie sont nécessaires pour basculer entre les modes actifs et inactifs. Cela implique que le passage à un mode de faible consommation, en plus de faire fonctionner le système en boucle ouverte, peut causer plus de gaspillage d'énergie que de rester dans un mode de transmission. Nous définissons N_2 comme le nombre de modes à faible consommation. Le nombre total de modes radio est donnée par $N = N_1 + N_2$.

Ci-après, *loi de contrôle* fait référence à la loi d'asservissement qui est utilisée pour calculer la commande $\hat{u}_k$ éventuellement envoyée au nœud récepteur et appliquée au système. *Politique de commutation* fait référence à la décision du nœud de passer à un mode donné. La décision de commutation est notée $\nu_k \in \{1, 2, \cdots, N\}$, où $\nu_k = i$ signifie que le mode radio choisi au temps k est le mode i.

Système

Le système à contrôler est un système linéaire à temps discret soumis à un bruit blanc additif Gaussien de moyenne nulle, décrit par l'Equation (1) :

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1}$$

où x_k est l'état au temps k, u_k la commande, prenant leurs valeurs dans $\mathbb{R}^{n_x}$ et $\mathbb{R}^{n_u}$ respectivement, et $w_k \sim \mathcal{N}(\mathbf{0}, W)$ est le bruit de mesure. A et B ont les dimensions appropriées, le système est controllable et il peut être instable.

Modélisation du composant radio

L'état du composant radio est le mode radio au temps k, m_k :

$$m_k \in \mathbb{M} \triangleq \mathbb{M}_1 \cup \mathbb{M}_2$$

avec $\mathbb{M}_1 \triangleq \{1, 2, \cdots, N_1\}$ l'ensemble des modes de transmissions et $\mathbb{M}_2 \triangleq \{N_1 + 1, N_1 + 2, \cdots, N\}$ l'ensemble des modes de basse consommation.

Le mode radio est mis à jour par la décision de commutation : $m_{k+1} = \nu_k$. Cela signifie que choisir $\nu_k \in \mathbb{M}_1$ implique la décision de transmettre. La consommation du composant radio pendant une période d'échantillonnage, appelé le coût de transmission, et dénoté $\theta_{m_k, m_{k+1}}$, dépends du mode radio m_k et de la décision de commutation ν_k.

La quantité d'énergie E consommée depuis la mise en marche du système (où $E_0 = 0$) peut être calculée comme suit :

$$E_{k+1} = E_k + \theta_{m_k, \nu_k} = E_k + \theta_{m_k, m_{k+1}}.$$

Modélisation du canal de communication

Comme c'est le cas dans [Imer et al., 2006], le canal est modélisé par un canal à effacement sans mémoire simplifié où le message $\hat{u}_k$ est perdu avec une probabilité $\epsilon(m_k)$ pour $m_k \in \mathbb{M}_1$, et sinon correctement reçu. Nous supposons que le processus de gain de canal est i.i.d. (indépendant et identiquement distribué).

Les probabilités de perte dépendent de la puissance d'émission utilisée par la puce radio, c'est-à-dire le mode de transmission. La puissance de transmission la plus élevée produit la probabilité de succès la plus grande, soit $\epsilon(1) < \epsilon(2) < \ldots < \epsilon(N_1)$. Nous considérons un modèle où les pertes concernent le

message à valeur réelle $\hat{u}_k$, et non pas des bits ou des paquets individuels. Les succès de transmission sont modélisés par la variable aléatoire de Bernoulli β_k, où 1 désigne le succès et

$$\mathbb{P}\{\beta_k = 0 | m_k = m\} = \epsilon(m),$$
$$\mathbb{P}\{\beta_k = 1 | m_k = m\} = 1 - \epsilon(m).$$

Étant donné le mode m_k, β_k est conditionnellement indépendant des valeurs précédentes de $\{\beta_h\}_{h<k}$ et $\{w_k\}_{h<k}$; les valeurs des probabilités de perte par mode, $\epsilon(m)$, $m \in \mathbb{M}_1$, sont supposées connues. Des accusés de réception sont envoyés par le nœud récepteur pour confirmer au capteur intelligent qu'une nouvelle mise à jour de la commande a été appliquée au système. Ces accusés de réception sont supposés fiables, c'est-à-dire toujours correctement reçus par le nœud intelligent. Cette hypothèse est raisonnable puisque le nœud actionneur a accès à une quantité infinie d'énergie pour les envoyer avec assez de puissance. Par ailleurs, un accusé de récepteur est un message court (éventuellement réduit à un seul bit), et il est donc possible de l'encoder avec une très grande redondance pour la correction d'erreur.

Politique de commutation et loi de commande

Le nœud capteur intègre une politique de commutation η (dont la co-conception avec la loi de commande μ, décrite ci-après, est l'objectif de cette thèse) pour attribuer le mode radio. La décision de basculer entre les modes est basée sur la sortie du système x_k, la commande précédente u_{k-1} et le mode radio en cours, noté m_k. En introduisant la notation $\tilde{u}_k = u_{k-1}$, la mémoire de la dernière commande, la décision de commutation est donnée par $\nu_k = \eta(x_k, \tilde{u}_k, m_k)$. L'accusé de réception envoyé par le nœud actionneur lorsque la transmission est réussie laisse le capteur intelligent avoir une connaissance parfaite de la dernière commande appliquée au système.

La commande appliquée au système, notée u_k, dépend de l'arrivée de la mise à jour de la commande $\hat{u}_k$, qui dépend elle-même de la réussite de la transmission et de la décision d'envoyer une mise à jour, comme décrit dans l'Equation (2). Si une mise à jour est reçue, alors la loi de commande est la loi optimale calculée par le capteur intelligent, notée $\hat{u}_k = \mu(x_k, \tilde{u}_k, m_k)$ et calculée dans les prochaines sections. Sinon, la commande est maintenue à sa valeur précédente tant qu'aucune mise à jour n'est reçue par le nœud intelligent :

$$u_k = \begin{cases} \beta_k \hat{u}_k + (1-\beta_k) u_{k-1}, & \text{en cas de transmission, c-a-d si} \nu_k \in \mathbb{M}_1, \\ u_{k-1},\ \forall \beta & \text{sinon.} \end{cases} \tag{2}$$

Modélisation des coûts de transmission

Un composant radio peut être modélisé par un automate où :

- les états décrivent l'état de la radio (par exemple `Transmission`, `Idle`, `Sleep`),
- et les transitions modélisent des événements, comme les signaux exogènes envoyés par le microprocesseur (les demandes de commutation de mode radio) ou les débordements de pile interne.

Certaines transitions ont une durée minimale, ce qui implique qu'un temps donné est nécessaire pour passer d'un état à un autre. Chacun des états est caractérisé par une consommation de courant, tout comme les événements de transition. Un exemple de contrainte temporelle issue de la documentation technique [Chipcon Products] est que $721\mu s$ sont nécessaires pour calibrer le synthétiseur de fréquence lors du passage d'un mode de faible consommation à un mode de transmission.

Le nombre d'états de l'automate est exactement le nombre de modes radio, N. Les modes qui permettent une transmission sont numérotés de 1 à N_1 et ceux qui ne transmettent pas et économisent l'énergie sont numérotés de $N_1 + 1$ à $N = N_1 + N_2$, comme expliqué auparavant. La Figure 6 donne un example de l'automate que nous utilisons dans le cas où $N = 5$ et $N_1 = 3$.

Les coûts de transition modélisés dans notre automate sont différents des valeurs brutes fournies par un constructeur de composants radio pour la raison suivante. Nous considérons que l'état de l'automate ne change qu'à l'instant d'échantillonnage. En d'autres termes, la puce radio reste dans un état

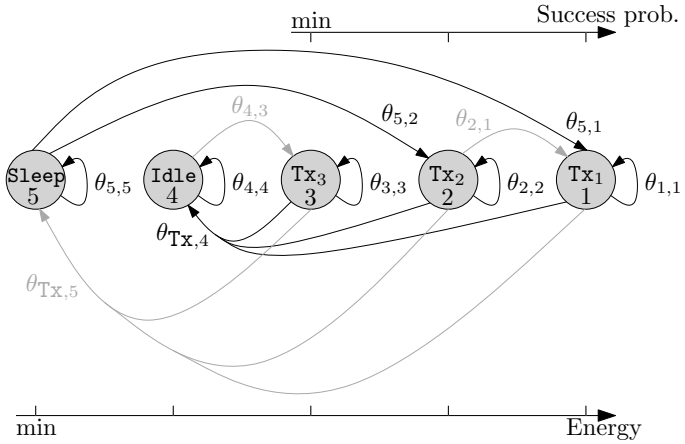


***Figure 6** – Illustration de l'automate qui modélise un composant radio avec $N_1 = 3$ et $N_2 = 2$.* `Idle` *est un mode intermédiaire entre les modes de transmission et le mode* `Sleep`*. Les modes sont ordonnés suivant leur consommation en énergie, par exemple $\theta_{5,5} < \theta_{4,4}$ où $\theta_{5,5}$ est le coût énergétique pour rester dans le mode* `Sleep` *et $\theta_{4,4}$ dans le mode* `Idle`*. Plus un mode de transmission consomme de l'énergie et plus sa probabilité de transmettre une donnée avec succès est élevée. Les flèches représentent les coûts de transition entre les états. Une figure plus détaillée aurait fait apparaitre toutes les transitions $\theta_{i,j}$ entre n'importe quels modes i et j.*

donné pendant l'ensemble de la période d'échantillonnage et ne peut commuter son état que périodiquement. Les coûts de transition de la Figure 6 peuvent être synthétisés dans la matrice Θ :

$$\Theta = \begin{bmatrix} \theta_{\texttt{L1}} & \theta_{\texttt{L2}} & \theta_{\texttt{L3}} & \theta_{\texttt{L1,Idle}} & \theta_{\texttt{L1,Sleep}} \\ \theta_{\texttt{L1}} & \theta_{\texttt{L2}} & \theta_{\texttt{L3}} & \theta_{\texttt{L1,Idle}} & \theta_{\texttt{L1,Sleep}} \\ \theta_{\texttt{L1}} & \theta_{\texttt{L2}} & \theta_{\texttt{L3}} & \theta_{\texttt{L1,Idle}} & \theta_{\texttt{L1,Sleep}} \\ \theta_{\texttt{Idle,L1}} & \theta_{\texttt{Idle,L2}} & \theta_{\texttt{Idle,L3}} & \theta_{\texttt{Idle}} & \theta_{\texttt{Idle,Sleep}} \\ \theta_{\texttt{Sleep,L1}} & \theta_{\texttt{Sleep,L2}} & \theta_{\texttt{Sleep,L3}} & \theta_{\texttt{Sleep,Idle}} & \theta_{\texttt{Sleep}} \end{bmatrix}$$

où $\theta_{\texttt{mode}}$ est une notation raccourcie de $\theta_{\texttt{mode,mode}}$ qui représente l'énergie nécessaire pour rester une période d'échantillonnage dans le mode `mode`, $\theta_{\texttt{mode1,mode2}}$ est l'énergie nécessaire pour commuter du mode `mode1` vers le mode `mode2`, et $\theta_{\texttt{Li}}$ est l'énergie nécessaire pour envoyer une commande avec le niveau de puissance i.

Scénarios dans lequels appliquer notre technique Notre approche prend en compte plusieurs modes de faible consommation afin d'économiser plus d'énergie qu'un simple modèle `On/Off` et afin de fournir une grille de résolution plus élevée pour équilibrer les coûts énergétiques et la qualité de l'information envoyée. Cependant, comme le modèle de transition que nous proposons ici est vu du point de vue applicatif, l'utilisation de modes de basse consommation intermédiaires ne sera bénéfique que sous les suppositions émises ci-dessous.

Supposition 1. *$\theta_{i,i} \geq \theta_{j,j} > 0$, $\forall i \in \mathbb{M}_1$ et $j \in \mathbb{M}_2$. Cela signifie que les modes de transmission consomment plus que les modes de faible consommation et que le coût pour rester dans un mode (quelque soit ce mode) est toujours strictement positif.*

Nous introduisons la convention de numéroter les modes en fonction de leurs coûts énergétiques, à savoir $\theta_{1,1} \geq \theta_{2,2} \geq \ldots \geq \theta_{N,N}$. La quantité d'énergie qui peut être économisée avec l'utilisation des modes de faible consommation est directement liée à la différence entre $\theta_{1,1}$ and $\theta_{N,N}$.

Supposition 2. *Pour tout $j_1 \in \mathbb{M}_2$ et $j_2 \in \mathbb{M}_2$, si $j_1 < j_2$ alors $0 < \theta_{\texttt{i},\texttt{j}_1} \leq \theta_{\texttt{i},\texttt{j}_2}$, $\forall i \in \mathbb{M}_1$. Cela signifie que le coût de transition d'un mode de transmission à un mode de faible consommation est plus grand pour un mode à faible consommation plus économe. La condition symétrique (depuis un mode de faible consommation vers un mode de transmission) est également supposée, c'est-à-dire que pour tout $j_1 \in \mathbb{M}_2$ et $j_2 \in \mathbb{M}_2$, si $j_1 < j_2$ alors $0 < \theta_{\texttt{j}_1,\texttt{i}} \leq \theta_{\texttt{j}_2,\texttt{i}}$, $\forall i \in \mathbb{M}_1$.*

2.2 Modèle commuté

L'évolution du système avec les différents choix de mode radio est ici formulé dans le formalisme des systèmes linéaires commutés, avec autant de systèmes que le nombre de modes N. Du point de vue de la commande, les différents modes sont en fait réduits à deux cas : lorsque se produit une transmission réussie (c'est-à-dire que la boucle de commande est fermée) et lorsque le système fonctionne en boucle ouverte. Les différents modes influencent la consommation d'énergie et aussi la probabilité de succès pour les modes permettant la transmission.

Le choix de la politique de commutation au temps k est équivalent à choisir le mode radio. L'évolution du système commuté dépend de x_k, l'état du système, de $\tilde{u}_k = u_{k-1}$, la mémoire de la dernière commande appliquée, et de m_k le mode de la puce radio. Nous définissons z_k comme l'état du système augmenté avec la mémoire de commande :

$$z_k = \begin{bmatrix} x_k \\ \tilde{u}_k \end{bmatrix} \in \mathbb{R}^{n_x+n_u}, \text{ et aussi } \omega_k = \begin{bmatrix} w_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_x+n_u}.$$

Le vecteur ω_k est le bruit w_k augmenté pour s'adapter à la dimension de z_k.

L'état du système commuté est alors donné par :

$$(z_k, m_k),$$

de telle manière que l'espace d'état du système commuté est $\mathbb{X} = \mathbb{R}^{n_x+n_u} \times \mathbb{M}$. L'évolution du système donné par l'Equation (1) avec la loi de commande $\mu(x_k, \tilde{u}_k, m_k)$ décrite dans l'Equation (2), ainsi que la politique de commutation $\eta(x_k, \tilde{u}_k, m_k)$, donne lieu au système commuté suivant :

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ \hat{u}_k = \mu(z_k, m_k), \end{cases} \tag{3}$$

où la fonction f_{ν_k} est définie par

$$f_{\nu_k}(z_k, \hat{u}_k, \beta_k, \omega_k) = \Phi_{\nu_k}(\beta_k) z_k + \Gamma_{\nu_k}(\beta_k)\hat{u}_k + \omega_k, \tag{4}$$

et les matrices $\Phi_{\nu_k}(\beta_k), \Gamma_{\nu_k}(\beta_k)$, pour $\nu_k \in \mathbb{M}$, sont comme suit :

1. si $\nu_k \in \mathbb{M}_1$ (s'il y a une transmission), alors

$$\Phi_{\nu_k}(\beta_k) = \begin{bmatrix} A & (1-\beta_k)B \\ \mathbf{0} & (1-\beta_k)\mathbf{I} \end{bmatrix} = \begin{cases} \Phi_{CL} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{si } \beta_k = 1 \\ \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \text{si } \beta_k = 0. \end{cases}$$

$$\Gamma_{\nu_k}(\beta_k) = \beta_k \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} = \begin{cases} \Gamma_{CL} = \begin{bmatrix} B \\ \mathbf{I} \end{bmatrix} & \text{si } \beta_k = 1 \\ \Gamma_{OL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \text{si } \beta_k = 0. \end{cases}$$

2. si $\nu_k \in \mathbb{M}_2$ (s'il n'y a pas de transmission), alors

$$\Phi_{\nu_k}(\beta_k) = \Phi_{OL}, \ \forall \beta_k,$$
$$\Gamma_{\nu_k}(\beta_k) = \Gamma_{OL}, \ \forall \beta_k.$$

2.3 Problème d'optimisation

Le but d'une co-conception contrôle et communication est d'équilibrer les performances de commande et l'énergie consommée par le composant radio. Nous utilisons le formalisme de la Commande Optimale pour résoudre un problème optimal basé sur une fonction de coût. Nous définissons d'abord le coût unitaire, noté ℓ, qui prend en compte les performances de commande et l'énergie sur une seule période d'échantillonnage :

$$\ell_{\nu_k}(x_k, m_k, u_k) = \underbrace{x_k^\top \bar{Q} x_k}_{\text{performances}} + \underbrace{u_k^\top \bar{R} u_k}_{\text{énergie de commande}} + \underbrace{\theta_{m_k,\nu_k}}_{\text{énergie de transition}}. \tag{5}$$

$\bar{Q}$ et $\bar{R}$ sont des matrices symétriques définies positives qui peuvent être ajustées pour équilibrer les performances et l'énergie. En utilisant les notations définies plus tôt, nous pouvons formuler ce coût unitaire sous forme commutée :

$$\ell_{v_k}(z_k, m_k, \hat{u}_k, \beta_k) = z_k^\top Q_{v_k}(\beta_k) z_k + \hat{u}_k^\top R_{v_k}(\beta_k)\hat{u}_k + \theta_{m_k, v_k}, \tag{6}$$

où les matrices $Q_{v_k}(\beta_k)$ et $R_{v_k}(\beta_k)$, pour $v_k \in \mathbb{M}$, sont définies comme suit :

1. si $v_k \in \mathbb{M}_1$, (si $u_k = \beta_k \hat{u}_k + (1-\beta_k)\tilde{u}_k$), alors

$$Q_{v_k}(\beta_k) = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & (1-\beta_k)\bar{R} \end{bmatrix} = \begin{cases} Q_{CL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{si } \beta_k = 1 \\ Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} & \text{si } \beta_k = 0, \end{cases}$$

$$R_{v_k}(\beta_k) = \beta_k \bar{R} = \begin{cases} R_{CL} = \bar{R} & \text{si } \beta_k = 1 \\ R_{OL} = \mathbf{0} & \text{si } \beta_k = 0. \end{cases}$$

2. si $v_k \in \mathbb{M}_2$, (si $u_k = \tilde{u}_k$), alors

$$Q_{v_k}(\beta_k) = Q_{OL},\ \forall \beta_k,$$
$$R_{v_k}(\beta_k) = R_{OL},\ \forall \beta_k.$$

Nous définissons maintenant la fonction de coût $J_{\mathscr{U},\mathscr{V}}$ qui considère l'espérance du coût que le système doit payer pour contrôler le système (1) avec la séquence de commande $\mathscr{U} = \{\mu_0, \mu_1, \ldots, \mu_{H-1}\}$ et la séquence de commutation $\mathscr{V} = \{\eta_0, \eta_1, \ldots, \eta_{H-1}\}$, sur un horizon de temps H et les conditions initiales z_0, m_0 :

$$J_{\mathscr{U},\mathscr{V}}(z_0, m_0) = \mathop{\mathsf{E}}_{\substack{\beta_k, \omega_k \\ k=0,1,\ldots}} \left[\ell_F(z_H, m_H) + \sum_{k=0}^{H-1} \lambda^k \ell_{v_k}(z_k, m_k, \hat{u}_k, \beta_k) \right], \tag{7}$$

avec $\hat{u}_k = \mu_k(z_k, m_k)$, $v_k = \eta_k(z_k, m_k)$, $z_{k+1} = f_{v_k}(z_k, \hat{u}_k, \beta_k, \omega_k)$, et $\ell_F(z_H, m_H)$ étant le coût unitaire final, et $\lambda \in [0, 1)$ étant un facteur d'oubli.

Le problème d'optimisation considéré pour le reste de cette thèse est de trouver une séquence de commande $\mathscr{U}^*$ et une séquence de commutation $\mathscr{V}^*$ telles que :

$$J^*(z_0, m_0) \triangleq J_{\mathscr{U}^*,\mathscr{V}^*}(z_0, m_0) = \min_{\mathscr{U},\mathscr{V}} J_{\mathscr{U},\mathscr{V}}(z_0, m_0). \tag{8}$$

Enfin, nous définissons l'espace de contrôle $\mathbb{U}(\tilde{u})$ comme suit :

Définition 1.

$$\mathbb{U}(\tilde{u}) \triangleq \{(u, v) : u \in \mathbb{R}^{n_u},\ v \in \mathbb{M}_1\} \cup \{(\tilde{u}, v) : v \in \mathbb{M}_2\}.$$

Pour simplifier la notation, nous utiliserons $\mathbb{U}(z)$ à la place de $\mathbb{U}(\tilde{u})$. La loi conjointe (μ, η) prend ses valeurs dans $\mathbb{U}(z)$.

3 Optimisation à horizon infini pour la commande économe

3.1 Problème d'optimisation dans le cas infini

Nous avons défini dans le cas général le problème d'optimisation que nous voulons résoudre avec l'Equation (8). Dans le cas infini, la somme des coûts unitaires présentée dans l'Equation (7) doit être considérée avec un horizon infini, soit $H = \infty$. Ceci implique que le coût unitaire final ℓ_F n'est plus considéré et que les séquences de contrôle et de commutation, $\mathscr{U}$ et $\mathscr{V}$, sont de dimension infinie. Pour des raisons évidentes d'implémentation, nous nous limitons donc à chercher une solution optimale invariante, en d'autres termes, nous cherchons les lois $\mu^*(x, \tilde{u}, m)$ et $\eta^*(x, \tilde{u}, m)$ telles que $\mathscr{U}^* = (\mu^*, \mu^*, \mu^*, \ldots)$ et $\mathscr{V}^* = (\eta^*, \eta^*, \eta^*, \ldots)$.

La fonction de coût dans le cas infini se réécrit alors de la manière suivante :

$$J_{\mu,\eta}(z_0, m_0) = \underset{\substack{\beta_k, \omega_k \\ k=0,1,\dots}}{\mathsf{E}} \left[\sum_{k=0}^{\infty} \lambda^k \ell_{\eta(z_k,m_k)}(z_k, m_k, \mu(z_k, m_k), \beta_k) \right],$$

Le problème d'optimisation dans le cas infini est alors défini comme suit :

$$J^*(z_0, m_0) \triangleq J_{\mu^*,\eta^*}(z_0, m_0) = \min_{\mu,\eta} J_{\mu,\eta}(z_0, m_0). \tag{9}$$

Le Principe de Bellman, qui sert de fondement à la Programmation Dynamique, nous démontre que le problème d'optimisation (9) vérifie la récurrence suivante :

$$J^*_{H-i}(z, m) = \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \underset{\beta,\omega}{\mathsf{E}} \left[\lambda^{H-i} \ell_v(z, m, \hat{u}, \beta) + J^*_{H-i+1}(f_v(z, \hat{u}, \omega, \beta), v) \right] \right\}.$$

Enfin, la Programmation Dynamique (voir [Bertsekas, 2007]) nous permet de montrer que la solution optimale au problème précédent peut être obtenue en itérant la Value Function, qui est donnée, $\forall (z, m) \in \mathbb{X}$, par :

$$\begin{aligned} V_0(z, m) &= 0 \\ V_{i+1}(z, m) &= \min_{(\hat{u},v)\in\mathbb{U}(z)} \left\{ \underset{\beta,\omega}{\mathsf{E}} \left[\ell_v(z, m, \hat{u}, \beta) + \lambda V_i(f_v(x, \hat{u}, \omega, \beta), v) \right] \right\} \\ \lim_{i\to\infty} V_i(z, m) &= J^*(z, m). \end{aligned} \tag{10}$$

3.2 Solution dans le cas idéal

Si nous nous plaçons maintenant dans un cas idéal, c'est-à-dire le cas déterministique où le système n'est pas soumis aux bruits de mesure ni aux pertes de paquets, alors nous pouvons prouver que la Value Function peut être écrite sous une forme explicite. Ceci est le but de cette sous-section. Notons que ne pas considérer les pertes de paquets revient à ne considérer qu'un seul mode de transmission. En effet, les différentes puissances d'émission servent à augmenter ou diminuer la probabilité de réception des données, alors que dans le cas idéal, la probabilité de bonne réception est toujours de 100%.

Nous commençons par présenter dans le cas idéal le modèle mathématique que nous avons introduit précédemment dans le cas général. Ensuite nous donnerons la forme explicite de la Value Function et nous terminerons cette sous-section par la présentation de l'implémentation de notre solution. Pour la preuve de notre résultat, et pour l'étude du cas général, nous renvoyons le lecteur vers la partie complète en Anglais de cette thèse.

Modèle mathématique dans le cas idéal

Modèle du système — Le système est identique au cas général sans bruit additif :

$$x_{k+1} = Ax_k + Bu_k.$$

Modèle du cannal — Quand une transmission est prévue, la commande calculée par le nœud intelligent $\hat{u}_k$ est toujours reçue par l'actionneur, c'est-à-dire $\beta_k = 1,\ \forall k$.

Politique de commutation — La politique de commutation est encore donnée par $\nu_k = \eta(x_k, u_{k-1}, m_k)$.

Loi de contrôle — La loi de contrôle que nous considérons ici est un retour d'état statique : $\hat{u}_k = -Kx_k = -\begin{bmatrix} K & \mathbf{0} \end{bmatrix} z_k$. Cette commande est appliquée au système seulement si une transmission est prévue par le nœud intelligent :

$$u_k = \begin{cases} \hat{u}_k, & \text{en cas de transmission,} \\ u_{k-1}, & \text{sinon.} \end{cases}$$

Modèle de la puce radio — Il est similaire au cas général avec $N_1 = 1$, c'est-à-dire qu'on ne considère qu'un seul mode de transmission. Les coûts de transition entre tous les modes sont synthétisés dans la matrices Θ. L'état de la radio est le mode au temps k, $m_k \in \mathbb{M}$.

Modèle commuté

Le modèle commuté est ici déterministique. Nous rappelons que z_k est l'état augmenté avec la mémoire de la commande $\tilde{u}_k = u_{k-1}$, $z_k = \begin{bmatrix} x_k^\top & \tilde{u}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x+n_u}$. Le modèle commuté est donné par :

$$\begin{cases} z_{k+1} = f_{v_k}(z_k) \\ m_{k+1} = v_k = \eta(z_k, m_k) \\ \hat{u}_k = \mu(z_k, m_k), \end{cases}$$

où la fonction f_{v_k} est définie par

$$f_{v_k}(z_k) = \Phi_{v_k} z_k,$$

et les matrices Φ_{v_k}, pour $v_k \in \mathbb{M}$, sont comme suit :

1. si $v_k = 1$, (en cas de transmission), alors

$$\Phi_{v_k} = \Phi_{CL} = \begin{bmatrix} A - BK & \mathbf{0} \\ -K & \mathbf{0} \end{bmatrix},$$

2. si $v_k \in \mathbb{M}_2$, (pas de transmission), alors

$$\Phi_{v_k} = \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Le coût unitaire est défini par :

$$\ell_{v_k}(z_k, m_k) = z_k^\top Q_{v_k} z_k + \theta_{m_k, v_k},$$

où les matrices Q_{v_k}, pour $v_k \in \mathbb{M}$, sont comme suit :

1. si $v_k = 1$, (en cas de transmission), alors

$$Q_{v_k} = Q_{CL} = \begin{bmatrix} \bar{Q} + K^\top \bar{R} K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

2. si $v_k \in \mathbb{M}_2$, (pas de transmission), alors

$$Q_{v_k} = Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix}.$$

Problème optimal

L'optimisation consiste à trouver la loi conjointe déterministique et stationnaire $\eta^*(z_k, m_k)$ tel que

$$J^*(z_0, m_0) \triangleq J_{\eta^*}(z_0, m_0) = \min_{\eta} J_\eta(z_0, m_0),$$

où la fonction de coût $J_\eta(z_0, m_0)$ est définie par :

$$J_\eta(z_0, m_0) = \lim_{H\to\infty} \sum_{k=0}^{H-1} \lambda^k \ell_{v_k}(z_k, m_k) = \sum_{k=0}^{\infty} \lambda^k \ell_{v_k}(z_k, m_k),$$

où $v_k = \eta(z_k, m_k),\ \forall k \geq 0$.

Forme explicite de la Value Function

Alors que de manière générale la Value Function ne trouve pas de forme analytique et oblige l'utilisation d'une grille de points pour la calculer, dans le cas déterministique, nous trouvons une forme explicite, comme détaillé dans cette sous-section. L'idée est d'exploiter une structure qui réapparait au fil des itérations. Par exemple, dans le cas bien connu LQ, la Value Function prend la forme suivante : $V_i(z) = z^\top \Pi_i z$.

Dans notre cas, la structure est un peu plus compliquée, il s'agit du minimum sur un ensemble fini de fonctions quadratiques de la forme $z^\top \Pi z + \pi_m$.

Si $V_0(z, m) \equiv 0,\ \forall (z, m)$, alors les itérations (10) donnent des Value Functions $V_i(z, m)$ telles que

$$V_i(z, m) = \min_{(\Pi, \pi) \in \mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}, \tag{11}$$

où l'ensemble $\mathscr{P}_i$ est composé d'éléments (Π, π), où Π est une matrice symétrique et $\pi = \begin{bmatrix} \pi_1, \pi_2, \ldots, \pi_N \end{bmatrix} \in \mathbb{R}^N$ est un vecteur de nombres scalaires positifs ou nuls, et π_m représente la $m^{\text{ième}}$ composante de π.

Nous prouvons le résultat ci-dessous avec un raisonnement par récurrence.

a) Itération de base :

Par définition, $V_0(z,m) = 0$ pour tout $(z,m) \in \mathbb{X}$. $V_1(z,m)$ peut être calculé en utilisant l'algorithme itératif (10) :

$$\begin{aligned} V_1(z,m) &= \min_{v\in\mathbb{M}} \left\{\ell_v(z,m) + \lambda V_0(f_v(z),v)\right\} = \min_{v\in\mathbb{M}} \left\{\ell_v(z,m)\right\} = \min_{v\in\mathbb{M}} \left\{z^\top Q_v z + \theta_{m,v}\right\} \\ &= \min\left\{z^\top Q_{CL} z + \theta_{m,1}; \min_{v\in\mathbb{M}_2} \left\{z^\top Q_{OL} z + \theta_{m,v}\right\}\right\} \\ &= \min\left\{z^\top Q_{CL} z + \theta_{m,1}; z^\top Q_{OL} z + \min_{v\in\mathbb{M}_2} \left\{\theta_{m,v}\right\}\right\} \\ &= \min_{(\Pi,\pi)\in\mathscr{P}_1} \left\{z^\top \Pi z + \pi_m\right\}, \end{aligned}$$

où

$$\mathscr{P}_1 = \left\{ \left(Q_{CL}, \begin{bmatrix} \theta_{1,1} \\ \theta_{2,1} \\ \vdots \\ \theta_{N,1} \end{bmatrix}^\top \right), \left(Q_{OL}, \begin{bmatrix} \min_{v\in\mathbb{M}_2}\{\theta_{1,v}\} \\ \min_{v\in\mathbb{M}_2}\{\theta_{2,v}\} \\ \vdots \\ \min_{v\in\mathbb{M}_2}\{\theta_{N,v}\} \end{bmatrix}^\top \right) \right\}$$

b) Récurrence :

Supposons que la Value Function à l'itération i est donnée par l'Equation (11) :

$$V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{z^\top \Pi z + \pi_m\right\}.$$

La Value Function à l'itération suivante peut être calculée en utilisant l'algorithme (10) :

$$\begin{aligned} V_{i+1}(z,m) &= \min_{v\in\mathbb{M}} \left\{\ell_v(z,m) + \lambda V_i(f_v(z),v)\right\} \\ &= \min_{v\in\mathbb{M}} \left\{z^\top Q_v z + \theta_{m,v} + \lambda \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{z^\top \Phi_v^\top \Pi \Phi_v z + \pi_v\right\}\right\} \\ &= \min_{v\in\mathbb{M},\,(\Pi,\pi)\in\mathscr{P}_i} \left\{z^\top \left(\lambda \Phi_v^\top \Pi \Phi_v + Q_v\right) z + \lambda\pi_v + \theta_{m,v}\right\} \\ &= \min\left\{ \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{z^\top \left(\lambda \Phi_{CL}^\top \Pi \Phi_{CL} + Q_{CL}\right) z + \lambda\pi_1 + \theta_{m,1}\right\}; \right. \\ &\qquad \left. \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{z^\top \left(\lambda \Phi_{OL}^\top \Pi \Phi_{OL} + Q_{OL}\right) z + \lambda \min_{v\in\mathbb{M}} \left\{\pi_v + \theta_{m,v}\right\}\right\}\right\} \\ &= \min_{(\Pi',\pi')\in\mathscr{P}_{i+1}} \left\{z^\top \Pi' z + \pi'_m\right\}, \end{aligned}$$

où

$$\mathscr{P}_{i+1} = \mathscr{P}_{i+1}^{(1)} \cup \mathscr{P}_{i+1}^{(2)} \tag{12}$$

$$\mathscr{P}_{i+1}^{(1)} \triangleq \left\{ \left(\lambda \Phi_{CL}^\top \Pi \Phi_{CL} + Q_{CL}, \begin{bmatrix} \lambda\pi_1 + \theta_{1,1} \\ \lambda\pi_1 + \theta_{2,1} \\ \vdots \\ \lambda\pi_1 + \theta_{N,1} \end{bmatrix}^\top \right) \text{ tel que } (\Pi,\pi) \in \mathscr{P}_i \right\}$$

$$\mathscr{P}_{i+1}^{(2)} \triangleq \left\{ \left(\lambda \Phi_{OL}^\top \Pi \Phi_{OL} + Q_{OL}, \begin{bmatrix} \min_{v\in\mathbb{M}_2}\{\lambda\pi_v + \theta_{1,v}\} \\ \min_{v\in\mathbb{M}_2}\{\lambda\pi_v + \theta_{2,v}\} \\ \vdots \\ \min_{v\in\mathbb{M}_2}\{\lambda\pi_v + \theta_{N,v}\} \end{bmatrix}^\top \right) \text{ tel que } (\Pi,\pi) \in \mathscr{P}_i \right\}$$

Implémentation de notre stratégie

La solution optimale est obtenue avec un processus de calcul en deux étapes.

Calculs hors-ligne

La première étape, calculée hors-ligne, consiste à calculer la Value Function $V_i(z,m)$ donnée par :

$$V_i(z,m) = \min_{(\Pi,\pi)\in\mathscr{P}_i} \left\{ z^\top \Pi z + \pi_m \right\}.$$

où le calcul des ensembles $\mathscr{P}_i$ est fait par récurrence comme expliqué dans la sous-section précédente. La Value Function converge vers la fonction de coût optimale quand i va à l'infini. En pratique, nous arrêtons la récurrence après un nombre suffisant d'itérations, noté I, pour observer la convergence. Par abus de notation, dans la suite, nous appelons $\mathscr{P}_\infty$ l'ensemble $\mathscr{P}_I$.

Calculs en ligne

La seconde étape est faite en ligne. Elle consiste à calculer la décision de commutation optimale au temps k, v_k^* comme une fonction de l'état courant de système (z_k, m_k), telle que donnée par la politique de commutation optimale : $v_k^* = \eta^*(z_k, m_k)$. Afin de calculer v_k^*, on doit d'abord calculer la Value Function pour l'état courant du système, ou plus précisemment déterminer le couple (Π_k, π_k) qui résulte de la minimisation de la Value Function :

$$(\Pi_k, \pi_k) \triangleq \arg\min_{(\Pi,\pi)\in\mathscr{P}_\infty} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\}. \tag{13}$$

Alors, la décision de commutation optimale est donnée par :

$$v_k^* = \eta^*(z_k, m_k) \triangleq \begin{cases} 1 & \text{if } (\Pi_k, \pi_k) \in \mathscr{P}_\infty^{(1)}, \\ \arg\min_{v\in\mathbb{M}_2} \left\{ \lambda\ \pi_k\big|_v + \theta_{m_k,v} \right\} & \text{if } (\Pi_k, \pi_k) \in \mathscr{P}_\infty^{(2)}. \end{cases}$$

La notation $\pi_k\big|_v$ fait référence au $v^{\text{ième}}$ élément du vecteur π_k.

On notera que, par construction, $\mathscr{P}_\infty$ est l'union de 2 sous-ensembles, $\mathscr{P}_\infty^{(1)}$ et $\mathscr{P}_\infty^{(2)}$. La définition de η^* est obtenue naturallement de la récursion (12).

3.3 Simulations

Pour illustrer la méthode proposée, nous présentons un exemple d'un système instable du premier ordre, avec les paramètres suivants :

$$\begin{aligned} &x_{k+1} = 1.074 x_k - 1.4808 u_k + w_k\,; \quad T_s = 0.05\text{s}\,; \\ &\bar{W} = 0.02\,; \quad K = -0.23\,; \quad \bar{Q} = 0.01\,; \quad \bar{R} = 0.1\,; \\ &\Theta = \begin{bmatrix} 2.85 & 1.8 & 1.9 \\ 3.2 & 1.4 & 6{\cdot}10^{-5} \\ 3.5 & 3.7{\cdot}10^{-3} & 6{\cdot}10^{-5} \end{bmatrix}; \quad \begin{matrix} N = 3 \\ \lambda = 0.8 \\ \epsilon = 0.3 \end{matrix} \end{aligned}$$

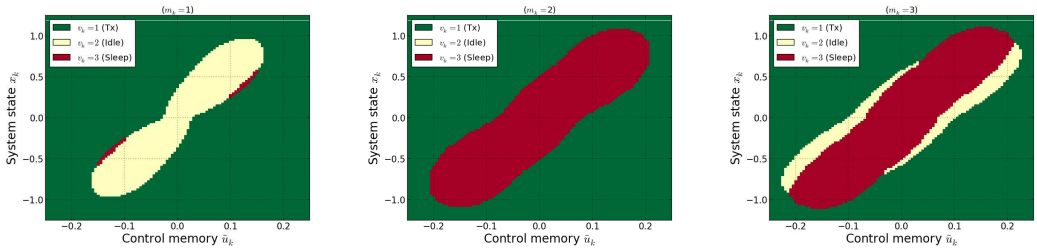
Nous considérons 3 modes radio (`Tx`, `Idle`, `Sleep`), les valeurs des coûts de transition sont données dans la matrice Θ en [mJ].

Simulations hors-ligne

Les calculs hors-ligne fournissent la politique de commutation $v = \eta^*(x, \tilde{u}, m)$. Ceci signifie qu'à l'issue des calculs hors-ligne, nous pouvons déterminer quelle est la politique de commutation v pour chaque point de l'espace d'état $(x, \tilde{u}, m)$. C'est ce qui est représenté dans la Figure 7.

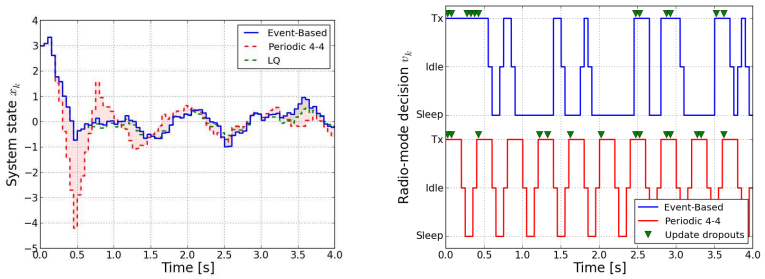
Simulations en ligne

Nous décidons de comparer dans des simulations en ligne (temporelles) le comportement de notre stratégie à la stratégie LQ bien connue, et des stratégies périodiques. Nous appelons une stratégie périodique i-j une loi commune qui est composée d'une loi de contrôle en retour d'état $u = -Kx$ et une politique de commutation périodique qui alterne i périodes d'échantillonnage en transmission et j périodes d'échantillonnage en mode de faible consommation. Du point de vue de la commande, le

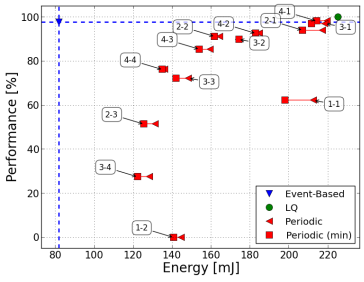


(a) Le mode actif est `Tx` ($m_k = 1$) (b) Le mode actif est `Idle` ($m_k = 2$) (c) Le mode actif est `Sleep` ($m_k = 3$)

***Figure 7** – Politique de commutation optimale résultant des calculs hors-ligne, vert ⟺ commuter vers* `Tx` *($v_k = 1$), jaune ⟺ commuter vers* `Idle` *($v_k = 2$) et rouge ⟺ commuter vers* `Sleep` *($v_k = 3$).*



(a) Etat du système x_k. (b) Décision de commutation v_k.



(c) Les performances face à l'énergie consommée E_{Tmax} pour plusieurs politiques périodiques.

***Figure 8** – Simulations en ligne pour comparer notre stratégie avec des politiques périodes (en considérant un bruit additif Gaussien de moyenne nulle et des pertes de paquets). Les triangles verts dans la Figure (b) indiquent les temps d'échantillonnage où une transmission était prévue mais une perte est survenue.*

choix du mode de faible consommation ne change rien. Mais d'un point de vue énergétique, nous faisons la différence entre une politique de commutation qui n'utilise que le mode le plus économe, et une politique qui utilise les modes qui donnent la plus faible consommation sur une période, et que nous notons "min".

Les Figures 8(a) et 8(b) montrent les résultats d'une simulation du modèle du premier ordre introduit précédemment dans les cas de notre stratégie déclenchée sur événement (Event Based), le cas périodique 4-4 ainsi que le cas bien connu LQ. Etant donné que le cas LQ consiste à toujours transmettre la commande, il a été omis de la Figure 8(b). L'état est soumis à un bruit de mesure, et la transmission de la commande est soumise à des pertes de paquets.
La Figure 8(c) compare les consommations énergétiques et les performances de ces stratégies ainsi que d'autres stratégies périodiques. Chacun des points a été obtenu en moyennant 100 simulations qui utilisent différentes séquences aléatoires de bruit et de pertes de paquets. La performance correspond à la fonction de coût calculée sans le terme énergétique lié à la consommation de la radio. Nous l'avons normalisée afin que la meilleure performance (c'est-à-dire le coût le plus faible) corresponde à 100%. Nous voyons très clairement que notre stratégie offre le meilleur compromis performance/énergie.

4 Optimisation à horizon fini pour la commande économe

4.1 Problème d'optimisation dans le cas fini

Dans cette sous-section, non seulement nous considérons l'horizon fini ($H < \infty$) mais en plus nous le considérons glissant. Cela signifie qu'à chaque période d'échantillonnage t, le problème d'optimisation est considéré sur l'horizon de temps de k vers $k+H-1$, quelque soit k. Si on applique directement la définition de la fonction de coût comme définie dans l'Equation (7), on obtient des séquences de contrôle et de commutation qui dépendent du temps k : $\mathscr{U}_k = \{\mu_k, \mu_{k+1}, \ldots, \mu_{k+H-1}\}$, $\mathscr{V}_k = \{\eta_k, \eta_{k+1}, \ldots, \eta_{k+H-1}\}$. Mais le système étant stationnaire dans le temps, le temps k lui même ne rentre pas en compte dans la solution optimale, et on peut paramétrer la fonction de coût en fonction de l'état (z_k, m_k) plutôt que du temps k. Ceci nous permet d'établir la fonction de coût suivante :

$$J_{\mathscr{U},\mathscr{V}}(z_k, m_k) = \mathop{\mathsf{E}}_{\substack{\beta_k,\omega_k \\ k=0,1,\ldots}} \left[z_{k+H}^\top Q_F z_{k+H} + \sum_{i=k}^{k+H-1} \ell_{v_i}(z_i, m_i, \hat{u}_i, \beta_i) \right].$$

Le facteur d'oubli λ a été pris égal à 1 et nous avons choisi un coût unitaire final quadratique :

$$\ell_F(z_H, m_H) = z_{k+H}^\top Q_F z_{k+H}. \tag{14}$$

La loi conjointe de commutation et de commade peut être calculée avec la Programmation Dynamique. La méthode de la Value Iteration introduite dans le début de ce résumé fournit la fonction de coût optimale J^* et les séquences optimales, $\mathscr{U}^*$, $\mathscr{V}^*$, qui ne sont pas garanties d'être uniques. Ici, la méthode Value Iteration est une application directe du Principe d'Optimalité de Bellman. Cette méthode est donnée par la récurrence de la Value Function $V_i(z,m)$ sur l'horizon de longueur H :

$$\begin{aligned}
V_0(z,m) &= z^\top Q_F z \\
V_{i+1}(z,m) &= \min_{(\hat{u},v)\in\mathbb{U}(z)} \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + V_i(f_v(z,\hat{u},\omega,\beta),v)\right] && i \in \{0,1,\ldots,H-1\} \\
(\mu_i^*(z,m), \eta_i^*(z,m)) &= \arg\min_{(\hat{u},v)\in\mathbb{U}(z)} \mathop{\mathsf{E}}_{\beta,\omega} \left[\ell_v(z,m,\hat{u},\beta) + V_i(f_v(z,\hat{u},\omega,\beta),v)\right] && i \in \{0,1,\ldots,H-1\} \\
J^*(z,m) &= V_H(z,m) && \forall (z,m) \in \mathbb{X} \\
\mathscr{U}^* &= \left\{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\right\} \\
\mathscr{V}^* &= \left\{\eta_0^*, \eta_1^*, \ldots, \eta_{N-1}^*\right\}
\end{aligned}$$

Dans le cas général, la Value Function ne peut pas être écrite sous une forme explicite comme nous l'avons montré dans le cas idéal sur un horizon infini. En effet, la formulation stochastique introduit une espérance à la fonction de coût qui complexifie les calculs. L'implémentation de notre algorithme repose donc sur un processus de discrétisation utilisant un grille de points sur une portion de l'espace d'état.

Bien que la solution du problème d'optimisation fournisse des séquences de commutation et de commande sur l'horizon H, l'utilisation du formalisme MPC (*Model Predictive Control* implique que seuls les premiers éléments de chacune des séquences sont appliqués au système, et la procédure est répétée au prochain intervalle d'échantillonnage. Il en résulte une politique conjointe optimale stationnaire (μ^*, η^*) donnée par :

$$\begin{cases} \mu^*(z_k, m_k) = \mu_0^*(z_k, m_k) \\ \eta^*(z_k, m_k) = \eta_0^*(z_k, m_k). \end{cases} \tag{15}$$

Comme cela a été fait précédemment, la stratégie optimale peut être divisée en un calcul hors-ligne et un calcul en ligne. La politique conjointe optimale (μ^*, η^*) est calculée hors-ligne. Ensuite, en ligne, l'algorithme suivant est exécuté par le nœud intelligent à chaque période d'échantillonnage :
Déterminer la décision de commutation, c'est-à-dire le prochain mode radio, $m_{k+1} = v_k^* = \eta^*(z_k, m_k)$ et

- *si $v_k^* \in \mathbb{M}_1$, déterminer la commande optimale $u_k^* = \mu^*(z_k, m_k)$ et l'envoyer à l'actionneur,*
- *si $v_k^* \in \mathbb{M}_2$, ne rien faire.*

Ceci conduit à une stratégie de contrôle déclenchée par événements, où la politique de commutation du mode η déclenche des mises à jour de la commande basées sur l'état actuel du système (z_k, m_k).

4.2 Stabilité ISS

Avant de discuter la stabilité dans le formalisme *Input-to-State Stability* (ISS), nous devons introduire quelques notations.

- Une fonction $\alpha(s) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ est appelée une $\mathcal{K}$-fonction, ou de classe $\mathcal{K}$, si elle est continue, strictement croissante et $\alpha(0) = 0$.
- Une fonction $\alpha(s) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ est appelée une $\mathcal{K}_\infty$-fonction, ou de classe $\mathcal{K}_\infty$, si elle est de classe $\mathcal{K}$ et $\alpha(s) \to \infty$ quand $s \to \infty$.
- Une fonction $\gamma(s,k) : \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ est appelée une $\mathcal{KL}$-fonction, ou de classe $\mathcal{KL}$, si pour chaque $k \geq 0$ fixé, $\gamma(\cdot, k)$ est de classe $\mathcal{K}$, pour chaque $s \geq 0$ fixé, $\gamma(s, \cdot)$ est décroissante, et $\gamma(s,k) \to 0$ quand $k \to \infty$.
- $\alpha^{-1}(s)$ définit, elle elle existe, l'inverse de la fonction $\alpha(s)$.
- $\alpha_1 \circ \alpha_2(s)$ définit la composition des fonctions $\alpha_1(s)$ et $\alpha_2(s)$ telle que $\alpha_1 \circ \alpha_2(s) = \alpha_1(\alpha_2(s))$.
- $\mathrm{id}(s)$ est la fonction identité, $\mathrm{id}(s) = s$.
- $\|s\|$ définit la norme Euclidienne de $s \in \mathbb{R}^n$.

Définitions et résultat de stabilité

Nous rappelons que le système commuté est donné par :

$$\begin{cases} z_{k+1} = f_{\nu_k}(z_k, u_k) \\ m_{k+1} = \nu_k = \eta(z_k, m_k) \\ u_k = \mu(z_k, m_k), \end{cases} \tag{16}$$

Nous sommes intéressés par la stabilité en boucle fermée de ce système avec la loi conjointe (15) et les conditions initiales (z_0, m_0), que nous notons $z_k(z_0, m_0)$, et qui évoluent comme suit :

$$\begin{cases} z_{k+1}(z_0, m_0) = f_{\nu_k^*}(z_k(z_0, m_0), u_k^*) \\ m_{k+1} = \nu_k^* = \eta^*(z_k, m_k) \\ u_k^* = \mu^*(z_k, m_k). \end{cases} \tag{17}$$

L'analyse de stabilité repose sur le formalisme ISS. Nous considérons la définition de stabilité pratique suivante.

Définition 2. *Le système bouclé* (17) *est appelé Globally Input-to-State practically Stable (GISpS) si il existe une fonction de classe $\mathcal{KL}$ γ, et une constante $c \geq 0$, telles que, pour tout $(z_0, m_0) \in \mathbb{X}$:*

$$\|z_k(z_0, m_0)\| \leq \gamma(\|z_0\|, k) + c, \quad k \in \mathbb{Z}_{\geq 0}. \tag{18}$$

La définition 2 implique que le système bouclé vérifie la proposition suivante :

Proposition 1. *Si le système bouclé* (17) *est GISpS, comme défini dans la Définition 2, alors pour tout scalaire positif ϵ et pour toute condition initiale (z_0, m_0), il existe un temps fini $\bar{k}(z_0, m_0, \epsilon)$ tel que :*

$$\|z_k\| \leq c + \epsilon, \ \forall k \geq \bar{k}(z_0, m_0, \epsilon).$$

Démonstration. La preuve suit immédiatement de l'Equation (18) et du fait que $\gamma(\|z_0\|, k) \to 0$ quand $k \to \infty$. □

Notre résultat de stabilité est sujet à une condition sur le facteur de poids final Q_F dans (14), qui est décrite dans la supposition suivante.

Supposition 3. *Le facteur de poids Q_F dans l'Equation* (14) *est tel qu'il existe $\kappa \in \mathbb{R}^{n_u \times (n_x + n_u)}$ tel que :*

$$\begin{aligned} Q_F &\succ 0 \\ \underbrace{(\Phi_{CL} - \Gamma_{CL}\kappa)^\top Q_F (\Phi_{CL} - \Gamma_{CL}\kappa) - Q_F + Q_{CL} + \kappa^\top R_{CL}\kappa}_{\triangleq -Q'_F} &\preceq 0 \\ \textit{et} \quad \max\{|\operatorname{eigs}(\Phi_{CL} - \Gamma_{CL}\kappa)|\} &\leq 1. \end{aligned} \tag{19}$$

L'existence d'un tel Q_F est discutée dans la partie en Anglais. On peut noter que l'existence du gain κ n'est utilisée que pour prouver la Supposition 3, ce gain n'est pas utilisé dans la suite comme le gain de retour d'état.

Pour prouver la stabilité pratique de la boucle fermée, nous utilisons une fonction GISpS-Lyapunov V, qui est définie dans la définition qui suit.

Définition 3. *$V : \mathbb{X} \to \mathbb{R}_{\geq 0}$ est appelée une fonction GISpS-Lyapunov pour le système (17) si*

- *il existe une paire de fonction de classe $\mathscr{K}_\infty$, α_1 et α_2, ainsi qu'une constante $c_1 \geq 0$ telles que, pour tout $(z, m) \in \mathbb{X}$:*

$$\alpha_1(|z|) \leq V(z, m) \leq \alpha_2(|z|) + c_1 \tag{20}$$

- *il existe une fonction de classe $\mathscr{K}_\infty$, α_3, et une constante $c_2 \geq 0$ telle que, pour tout $(z, m) \in \mathbb{X}$:*

$$\begin{aligned} \Delta V(z, m) &\triangleq V(f_{\nu^*}(z, m, u^*)) - V(z, m) \\ &\leq -\alpha_3(|z|) + c_2 \end{aligned} \tag{21}$$

Maintenant, nous donnons les deux théorèmes principaux qui prouvent que le système en boucle fermée est stable au sens ISS.

Théorème 1. *Si le système (17) admet une fonction GISpS-Lyapunov, alors il est stable au sens GISpS.*

Théorème 2. *Si la Supposition 3 est vérifiée, alors le système en boucle fermée (17) est GISpS.*

La preuve des résultats de stabilité est basée sur le formalisme ISS et repose sur les travaux de [Raimondo et al., 2009, Lazar et al., 2008] et [Jiang and Wang, 2001].

Les auteurs dans [Raimondo et al., 2009] considèrent le problème de stabilité ISpS régionale. La notion de stabilité pratique signifie que l'état du système ne converge pas asymptotiquement vers un point, mais vers un ensemble, et une fois cet ensemble atteint, le système y reste. La notion de stabilité régionale se réfère aux systèmes ayant des contraintes. En général, il existe des contraintes sur la perturbation, ou sur l'espace d'état ou l'espace de commande. Les auteurs de [Jiang and Wang, 2001] d'autre part considérent la stabilité ISS globale, ne prenant pas en compte de contrainte. Notre contribution consiste à considérer la stabilité ISS dans le cas global et pratique.

Sketch de la preuve du Théorème 1

Cette preuve est basée sur les preuves de ISS et ISPS dans [Jiang and Wang, 2001, Raimondo et al., 2009]. Nous supposons que les Equations (20)-(21) sont vérifiée, c'est-à-dire que le système (17) admet un fonction GISpS-Lyapunov, notée $V(z, m)$. Montrons alors que le système en boucle fermée est GISpS dans le sens de la Définition 2. La preuve est divisée en trois étapes : d'abord, nous montrons que le système en boucle fermée admet un ensemble invariant (tel que défini dans la Définition 4), puis nous montrons que l'ensemble invariant est attractif ; enfin nous prouvons que d'avoir un ensemble invariant attractif est équivalent à la stabilité pratique. Mais avant de poursuivre, nous donnons la définition d'un ensemble invariant.

Définition 4. *Un ensemble $\Omega \subset \mathbb{X}$ est appelé un ensemble invariant pour le système (17) si tout $(z, m) \in \Omega$ vérifie $f_{\eta^*(z,m)}(z, \mu^*(z, m)) \in \Omega$.*

Etape 1 La première étape consiste à montrer que le système en boucle fermée (17) admet un ensemble invariant $\Omega \subset \mathbb{X}$. Cet ensemble est défini par :

$$\Omega \triangleq \left\{(z, m) \in \mathbb{X} : V(z, m) \leq \omega(c_3)\right\}, \tag{22}$$

où les définitions de ω et c_3 sont omises ici.

Etape 2 La deuxième étape consiste à prouver que l'ensemble invariant Ω est attractif, c'est-à-dire que pour tout $(z_k, m_k) \notin \Omega$, il existe un scalaire fini $\bar{k} \geq k$ tel que, $(z_{\bar{k}}, m_{\bar{k}}) \in \Omega$. Dans cette étape, nous prouvons que $\forall k \in \{0, 1, \ldots, \bar{k}\}$, nous avons :

$$V(z_k, m_k) \leq \hat{\gamma}(V(z_0, m_0), k)$$

où la définition de $\hat{\gamma}$ est omise ici.

Etape 3 La dernière étape consiste à prouver que l'Equation (18) est vérifiée. Nous reprenons les résultats des étapes précédentes , $\forall (z_0, m_0) \in \mathbb{X}$, $\forall k \in \mathbb{Z}_{\geq 0}$:

- si $(z_k, m_k) \in \Omega,\ V(z_k, m_k) \leq \omega(c_3)$,
- if $(z_k, m_k) \notin \Omega,\ V(z_k, m_k) \leq \hat{\gamma}\left(V(z_0, m_0, k)\right)$.

L'Equation (20) implique que $\|z_k\| \leq \alpha^{-1}(V(z_k, m_k))$, et donc nous obtenons :

- si $(z_k, m_k) \in \Omega,\ \|z_k\| \leq \alpha^{-1}\left(\omega(c_3)\right)$,
- si $(z_k, m_k) \notin \Omega,\ \|z_k\| \leq \alpha^{-1}\left(\hat{\gamma}\left(V(z_0, m_0), k\right)\right)$.

Dans tout les cas, nous avons :

$$\|z_k\| \leq \alpha^{-1}\left(\hat{\gamma}\left(V(z_0, m_0), k\right)\right) + \alpha^{-1}\left(\omega(c_3)\right).$$

La vérification de l'Equation (18) suit quelques manipulations de cette dernière assertion.

Sketch de la preuve du Théorème 2

Nous présentons ici la preuve que le système (17) admet une fonction GISpS-Lyapunov. Cette partie est prouvée en quatre étapes, une pour chacune des inégalités dans la Définition 3, plus une dernière étape utilisant le Théorème 1 pour conclure que notre système est stable. Nous prenons la Value Function V_H de l'Equation (23) comme candidate.

$$J^*(z_k, m_k) = V_H(z_k, m_k) = \min_{(\Pi, \pi) \in \mathscr{P}_H} \left\{ z_k^\top \Pi z_k + \pi_{m_k} \right\} \tag{23}$$

Etape 1 L'inéquation $\alpha_1(|z|) \leq V_H(z, m)$ peut être vérifiée avec :

$$\alpha_1(|z|) = z^\top \underline{\Pi} z,$$

où :

$$z_k^\top \underline{\Pi} z_k \leq \min_{(\Pi, \pi) \in \mathscr{P}_H} \left\{ z_k^\top \Pi z_k \right\}.$$

Etape 2 L'inéquation $V(z, m) \leq \alpha_2(|z|) + c_1$ peut être vérifiée avec :

$$\begin{aligned} \alpha_2(|z|) &= z^\top \hat{\Pi} z \\ c_1 &= H\theta_{\max} \end{aligned}$$

où $\theta_{\max}$ est le coût de transition le plus élevé, et $\hat{\Pi}$ est obtenu en considérant une loi conjointe qui consiste à toujours transmettre, c'est-à-dire le cas LQ classique.

Etape 3 L'inéquation $\Delta V(z, m) \leq -\alpha_3(|z|) + c_2$ peut être vérifiée avec :

$$\begin{aligned} \alpha_3(|z|) &= z_k^\top \left(Q_{CL} + Q_{OL} + \kappa_{\min}^\top R_{CL} \kappa_{\min} \right) z_k \\ c_2 &= \theta_{\max} - \theta_{\min} \end{aligned}$$

où $\theta_{\min}$ est le coût de transition le plus faible, et la définition de $\kappa_{\min}$ est omise ici. C'est cette étape qui fait entrer en jeu la Supposition 3

Etape 4 Nous avons prouvé que V_H est une fonction GISpS-Lyapunov pour le système (17). Donc, d'après le Théorème 2, le système (17) est GISpS.

4.3 Simulations

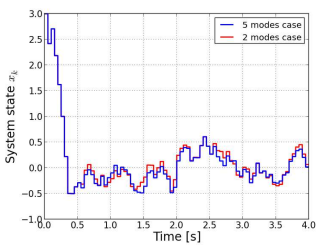
Nous ne présentons dans cette sous section que les résultat de simulation dans le cas général, en considérant des pertes de paquets et en ajoutant du bruit de mesure dans les simulations en ligne. Le but de ces simulations est de montrer l'avantage d'utiliser plusieurs mode radio au lieu de simplement deux modes, un de transmission et l'autre d'économie d'énergie en boucle ouverte. Nous considérons

donc ici 3 modes de transmissions (notés de $\texttt{Tx}_1$ à $\texttt{Tx}_3$), et deux modes de faible consommation (appelés `Idle` et `Sleep`). Le système est du premier ordre, instable en boucle ouverte :

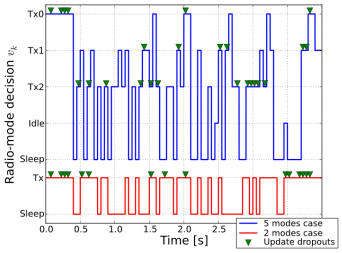
$$x_{k+1} = 1.074x_k - 1.4808u_k + w_k\,; \quad T_s = 0.05\text{s}\,;$$
$$\bar{W} = 0.02\,; \quad \bar{Q} = 0.01\,; \quad \bar{R} = 0.1\,; \quad Q_F = 1.5\,;$$
$$\Theta = \begin{bmatrix} 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.73 & 2.76 & 1.80 & 0.24 & 6\cdot 10^{-5} \\ 3.78 & 2.81 & 1.85 & 0.24 & 6\cdot 10^{-5} \\ 3.98 & 3.02 & 2.05 & 0.20 & 6\cdot 10^{-5} \end{bmatrix}\,; \quad \epsilon = \begin{bmatrix} 0.35 \\ 0.4 \\ 0.5 \end{bmatrix}$$

où les élements de la matrice Θ sont les coûts de transition $\theta_{i,j}$, donnés en [mJ], et les éléments du vecteur ϵ représentent les probabilités de perte d'information pour chacun des modes de transmission.

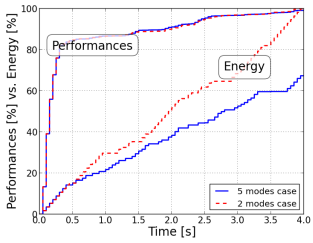
Les Figures 9(a)-9(c) donnent l'état du système x_k, la décision de commutation ν_k et la performance et l'énergie consommée de notre loi conjointe dans le cas où 5 modes radio sont utilisés et dans le cas où les modes sont limités à Tx_2 et `Sleep`. Nous pouvons voir que les deux trajectoires de l'état sont très similaires. La quantité de transmissions et les pertes sont similaires mais la quantité d'énergie consommée par le cas avec 5 modes est nettement inférieure à l'énergie consommée dans le cas avec 2 modes. L'utilisation de plusieurs modes d'émission et de faible consommation permet d'adapter les décisions de transmission au besoin réel des performances en boucle fermée. Les transmissions critiques consomment plus d'énergie afin de réduire la probabilité de perte d'information tandis que les mises à jour moins nécessaires sont ignorées ou utilisent des niveaux de puissance plus faibles.



(a) Etat du système x_k.



(b) Décision de commutation ν_k.



(c) Performances $\sum_{i=0}^{k}(x_i^\top \tilde{Q} x_i + u_i^\top \tilde{R} u_i)$ (courbe du haut) et l'énergie cumulée E_k. Les courbes sont normalisées à la plus grande valeur à la fin de la simulation.

***Figure 9** – Une simulation temporelle qui comparent nos stratégies EBC dans les cas de 5 et 2 modes radio (en bleu et rouge, respectivement).*

Conclusion

Les systèmes contrôlés en réseau constituent un sujet qui a attiré beaucoup d'attention au cours des dix dernières années. En outre, l'utilisation de la communication sans fil est particulièrement intéressante dans de nombreux contextes où il est difficile ou contraignant d'utiliser des fils. Grâce à la réduction des coûts de production, on peut désormais intégrer des unités de calcul, de communication et de détection dans un seul appareil, que nous appelons *capteur intelligent* dans cette thèse. Ces appareils sont peu chers à produire, et ils ont ouvert la voie à de nombreux problèmes difficiles et intéressants dans un très large éventail d'applications, telles que le contrôle du trafic, la chirurgie à distance ou l'automatisation industrielle.

Les économies d'énergie et la robustesse face aux pertes de données sont devenues des défis majeurs dans les systèmes sans fil. Ces défis ont été abordés par la Communauté de la Commande et la Communauté de la Communication. Bien que les intérêts de ces deux communautés se chevauchent, leurs approches diffèrent en général, et les contributions dans un domaine bénéficient assez peu à l'autre domaine. En effet, la Communauté de la Communication est principalement intéressée par les systèmes en boucle ouverte (pour la surveillance) avec une description de bas niveau (généralement le niveau des paquets), tandis que la Communauté de la Commande détient une théorie forte en prenant

en compte un canal de communication parfait et elle se limite souvent au niveau de la couche application. Comme présenté dans l'introduction, cette thèse se focalise sur la co-conception du contrôle et de la communication dans le cadre des systèmes contrôlés en réseau sans fil avec un accent particulier porté sur les économies d'énergie.

Le but du Chapitre 1 est précisément de comprendre comment l'énergie est utilisée dans un système contrôlé en réseau. Il passe en revue quelques unes des principales méthodes utilisées pour économiser de l'énergie dans les nœuds sans fil. Ce chapitre couvre à la fois les Communautés de la Commande et de la Communication. La revue est organisée dans une architecture multi-couches. En effet, il est naturel de modéliser les protocoles de communication en utilisant un empilement de couches correspondant à différents niveaux d'abstraction. Nous utilisons une pile à quatre couches composée de la couche physique, la couche liaison de données, la couche réseau et la couche d'application.

Après avoir compris comment l'énergie peut être sauvée, le reste de la thèse porte sur une nouvelle stratégie qui consiste en une co-conception entre les couches physique, liaison de données et application. Le Chapitre 2 est un chapitre descriptif qui présente notre stratégie, la configuration que nous considérons, et donne une introduction générale sur les procédures qui sont utilisées dans les chapitres suivants pour calculer notre stratégie. Elle est calculée dans le cadre de la commande optimale avec l'utilisation d'une fonction de coût. La fonction de coût est considérée sur un horizon qui est infini dans le Chapitre 3 et fini dans le Chapitre 4. Dans les deux cas, le problème d'optimisation est résolu en utilisant le formalisme de la Programmation Dynamique, et plus précisément la méthode de la Value Iteration.

Malgré les résultats intéressants obtenus dans cette thèse, de nombreuses questions restent encore ouvertes. Les plus importantes étant les preuves de stabilité qui n'ont pas pu être démontrées dans cette thèse, bien que le cas en horizon infini avec un facteur d'oubli soit un problème ouvert dans la Communauté de la Programmation Dynamique depuis très longtemps. Les extensions possibles des travaux dans cette thèse sont la prise en compte de la récupération d'énergie, le développement de méthodes de calcul numérique rapides pour accélérer l'obtention des solutions hors-ligne et en ligne, la considération de valeur quantifiées, l'utilisation de contrôle à déclenchement autonome (*self-triggered control*) ou encore la prise en compte de plusieurs nœuds capteurs.

Bibliography

T. Abdelzaher, T. He, and J. Stankovic. Feedback control of data aggregation in sensor networks. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 1490–1495, Dec. 2004.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002. ISSN 1389-1286.

J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, Dec. 2004.

M. Alamir. A framework for real-time implementation of low-dimensional parameterized NMPC. *Automatica*, 48(1):198–204, 2012. ISSN 0005-1098.

A. Alessio and A. Bemporad. A survey on Explicit Model Predictive Control. In *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 345–369. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-01093-4.

P. Alriksson and A. Rantzer. Sub-optimal sensor scheduling with error bounds. In *Procedings of the 16th IFAC World Congress*, volume 16, July 2005.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, Sept. 2010. ISSN 0018-9286.

D. Antunes, W.P.M.H. Heemels, J.P. Hespanha, and C. Silvestre. Scheduling measurements and controls over networks - Part II: Rollout strategies for simultaneous protocol and controller design. In *American Control Conference*, pages 2036–2041, 2012a.

D. Antunes, J.P. Hespanha, and C. Silvestre. Volterra integral approach to impulsive renewal systems: Application to networked control. *IEEE Transactions on Automatic Control*, 57:607–619, Mar. 2012b.

D. Antunes, J.P. Hespanha, and C. Silvestre. Stability of networked control systems with asynchronous renewal links: An impulsive systems approach. *Automatica*, 49(2):402–413, 2013. ISSN 0005-1098.

K.-E. Årzén. A simple event-based PID controller. In *14th World Congress of IFAC*, volume 18, pages 423–428, Beijing, P.R. China, Jan. 1999.

K.J. Åström and B.M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2011–2016, Dec. 2002.

L. Bao, M. Skoglund, C. Fischione, and K.H. Johansson. Optimized rate allocation for state feedback control over noisy channels. In *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference*, pages 573–578, 2009.

K. C. Barr and K. Asanović. Energy-aware lossless data compression. *ACM Trans. Comput. Syst.*, 24(3): 250–291, 2006. ISSN 0734-2071.

G.A. Bekey and R. Tomovic. Sensitivity of discrete systems to variation of sampling interval. *IEEE Transactions on Automatic Control*, 11(2):284–287, 1966. ISSN 0018-9286.

R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *37th IEEE Conference on Decision and Control*, volume 2, pages 2133–2138, 1998.

A. Bemporad, F. Borrelli, and M. Morari. Model Predictive Control Based on Linear Programming - The Explicit Solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, Dec. 2002a.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002b. ISSN 0005-1098.

D. Bernardini and A. Bemporad. Energy-aware robust Model Predictive Control based on wireless sensor feedback. In *47th IEEE Conference on Decision and Control*, pages 3342–3347, Dec. 2008.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. 2005a.

D. P. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. In *European Journal of Control, Fundamental Issues in Control*, volume 11, 2005b.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. 2007.

D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*. Athena Scientific, 1978.

M. I. Brownfield, T. Nelson, S. Midkiff, and N. J. Davis. Wireless sensor network radio power management and simulation models. *The Open Electrical & Electronic Engineering Journal*, 4:21–31, 2010.

C. Canudas De Wit and J. Jaglin. Energy-aware and entropy coding for networked controlled linear systems. *International Journal of Robust and Nonlinear Control*, 19:1851–1870, 2009.

C. Canudas De Wit, K. Crisanto Vega, and J. Jaglin. Non uniform sampling coding in networked controlled linear systems. In *Proceedings of the European Control Conference*, July 2007a.

C. Canudas De Wit, J. Jaglin, and C. Siclet. Energy-aware 3-level coding and control co-design for sensor network systems. *16th IEEE International Conference on Control Applications*, 2007b.

C. Canudas De Wit, F. Gomez Estern, and F. Rubio. Delta-Modulation Coding Redesign for Feedback-Controlled Systems. *IEEE Transactions on Industrial Electronics*, 56(7):1–20, June 2009.

R. Carli, E. D'Elia, and S. Zampieri. A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 7512–7517, Dec. 2011.

A. Chaillet and A. Bicchi. Delay compensation in packet-switching networked controlled systems. In *47th IEEE Conference on Decision and Control*, pages 3620–3625, 2008.

Y.P. Chen, A.L. Liestman, and J. Liu. A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 55(3):789–796, May 2006. ISSN 0018-9545.

C.-F. Chiasserini and E. Magli. Energy-efficient coding and error control for wireless video-surveillance networks. *Telecommunication Systems*, 26(2-4):369–387, June 2004. ISSN 1018-4864 (Print) 1572-9451 (Online).

Chipcon Products. *CC1100 Low-Power Sub- 1 GHz RF Transceiver*. Texas Instruments.

D. Ciscato and L. Martiani. On increasing sampling efficiency by adaptive sampling. *IEEE Transactions on Automatic Control*, 12(3):318–318, 1967. ISSN 0018-9286.

R. Cogill. Event-based control using quadratic approximate value functions. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 5883–5888, Dec. 2009.

R. Cogill, S. Lall, and J.P. Hespanha. A constant factor approximation algorithm for event-based sampling. In *American Control Conference*, pages 305–311, 2007.

J. Colandairaj, G.W. Irwin, and W.G. Scanlon. Wireless networked control systems with QoS-based sampling. *Control Theory Applications, IET*, 1(1):430–438, Jan. 2007. ISSN 1751-8644.

L. H.A. Correia, D. F. Macedo, A. L. dos Santos, A. A.F. Loureiro, and J. M. S. Nogueira. Transmission power control techniques for wireless sensor networks. *Computer Networks*, 51(17):4765–4779, 2007. ISSN 1389-1286.

D. B. Dačić and D. Nešić. Quadratic stabilization of linear networked control systems via simultaneous protocol and controller design. *Automatica*, 43(7):1145 – 1155, 2007. ISSN 0005-1098.

P. Di Marco, P. G. Park, C. Fischione, and K. H. Johansson. TREnD: a timely, reliable, energy-efficient dynamic wsn protocol for control application. In *IEEE International Conference on Communications*, 2010.

O.L. Do Valle Costa, M.D. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer, 2005. ISBN 9781852337612.

M.C.F. Donkers, W.P.M.H. Heemels, N. van de Wouw, and L. Hetel. Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9):2101–2115, 2011. ISSN 0018-9286.

M.C.F. Donkers, W.P.M.H. Heemels, D. Bernardini, A. Bemporad, and V. Shneer. Stability analysis of stochastic networked control systems. *Automatica*, 48(5):917–925, 2012. ISSN 0005-1098.

C. Erin and H.H. Asada. Energy optimal codes for wireless communications. *Decision and Control*, 5: 4446–4453, 1999.

C. Fischione, K. H. Johansson, F. Graziosi, and F. Santucci. Distributed cooperative processing and control over wireless sensor networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 1311–1316, 2006. ISBN 1-59593-306-9.

C. Fischione, S. Coleri Ergen, P. Park, K. H. Johansson, and A. Sangiovanni-Vincentelli. Medium access control analytical modeling and optimization in unslotted IEEE 802.15.4 wireless sensor networks. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 440–448, 2009a. ISBN 978-1-4244-2907-3.

C. Fischione, A. Speranzon, K. H. Johansson, and A. Sangiovanni-Vincentelli. Peer-to-peer estimation over wireless sensor networks via Lipschitz optimization. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 241–252, 2009b. ISBN 978-1-4244-5108-1.

E. Fridman, A. Seuret, and J.-P. Richard. Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica*, 40(8):1441–1446, 2004. ISSN 0005-1098.

A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1992. ISBN 9780792391814.

A.J. Goldsmith and S.B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Communications*, 9(4):8–27, Aug. 2002. ISSN 1536-1284.

F. Gomez-Estern, C. Canudas-de Wit, F.R. Rubio, and J. Fornes. Adaptive Delta-modulation coding for networked controlled systems. In *American Control Conference*, pages 4911–4916, July 2007.

T. M. P. Gommans, W. P. M. H. Heemels, N. W. Bauer, and N. van de Wouw. Compensation-based control for lossy communication networks. In *American Control Conference*, 2012.

D. Görges, M. Izak, and S. Liu. Optimal control and scheduling of switched systems. *IEEE Transactions on Automatic Control*, 56(1):135–140, Jan. 2011. ISSN 0018-9286.

V.K. Goyal. Theoretical foundations of transform coding. *IEEE Transactions on Signal Processing*, 18(5): 9–21, Sept. 2001. ISSN 1053-5888.

L. Grüne and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer, 2011.

S. Gupta. Increasing the sampling efficiency for a control system. *IEEE Transactions on Automatic Control*, 8(3):263–264, 1963. ISSN 0018-9286.

V. Gupta, T. Chung, B. Hassibi, and R. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, Feb. 2006. ISSN 00051098.

W. P M H Heemels, A.R. Teel, N. van de Wouw, and D. Nešić. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 55(8):1781–1796, 2010. ISSN 0018-9286.

W.P.M.H. Heemels and N. van de Wouw. Stability and stabilization of networked control systems. *Networked Control Systems, Lecture notes in control and information sciences*, 406:203–253, 2010.

W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2008.

W.P.M.H. Heemels, M.C.F. Donkers, and A.R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013. ISSN 0018-9286.

T. Henningsson, E. Johannesson, and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, 2008.

Onésimo Hernández-Lerma and Jean Bernard Lasserre. *Discrete-Time Markov Control Process, Basic Optimality Criteria*, volume 1. Springer, 1996.

J.P. Hespanha, P. Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, Jan. 2007. ISSN 0018-9219.

W. Hu, G. Liu, and D. Rees. Event-driven networked predictive control. *IEEE Transactions on Industrial Electronics*, 54(3):1603–1613, 2007. ISSN 0278-0046.

M.F. Huber. Optimal pruning for multi-step sensor scheduling. *IEEE Transactions on Automatic Control*, 57(5):1338–1343, May 2012. ISSN 0018-9286.

O. C. Imer, S. Yüksel, and T. Başar. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42:1429–1439, Sept. 2006. ISSN 0005-1098.

O.C. Imer and T. Basar. To measure or to control: optimal control with scheduled measurements and controls. June 2006.

O.C. Imer and T. Basar. Optimal estimation with limited measurements. *Int. J. Syst., Control Commun.*, 2:5–29, Jan. 2010. ISSN 1755-9340.

H. Ishii and T. Basar. Remote control of LTI systems over networks with state quantization. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 1, pages 830–835, Dec. 2002.

J. Jaglin, C. Canudas De Wit, and C. Siclet. Delta Modulation for Multivariable Centralized Linear Networked Controlled Systems. In *47th IEEE Conference on Decision and Control*, page 8, 2008.

Z.-P. Jiang and Y. Wang. Input-to-State Stability for discrete-time nonlinear systems. *Automatica*, 37 (6):857–869, 2001. ISSN 0005-1098.

Z.-P. Jiang and Y. Wang. A converse Lyapunov theorem for discrete-time systems with disturbances. *Systems & Control Letters*, 45(1):49–58, Jan. 2002.

M. Johansson, E. Bjornemo, and A. Ahlen. Fixed link margins outperform power control in energy-limited wireless sensor networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages I513–516, April 2007.

G. Joshi, S. Jardosh, and P. Ranjan. Bounds on dynamic modulation scaling for wireless sensor networks. In *3rd International Conference on Wireless Communication and Sensor Networks*, pages 13–16, Dec. 2007.

R. Jurdak, A.G. Ruzzelli, and G. O'Hare. Adaptive radio modes in sensor networks: How deep to sleep? In *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 386–394, June 2008.

J. Kim and J.G. Andrews. An energy efficient source coding and modulation scheme for wireless sensor networks. *Signal Processing Advances in Wireless Communications*, pages 710–714, June 2005.

N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2:8–13, 2005.

K. Lahiri, S. Dey, and A. Raghunathan. Communication-based power management. *IEEE Design & Test of Computers*, 19(4):118–130, July 2002. ISSN 0740-7475.

M. Lazar, D. Muñoz de la Peña, W.P.M.H. Heemels, and T. Alamo. On input-to-state stability of min–max nonlinear Model Predictive Control. *Systems & Control Letters*, 57(1):39–48, 2008. ISSN 0167-6911.

F. Li. NASC: Network-Aware Source Coding for wireless broadcast channels with multiple sources. In *IEEE 64th Vehicular Technology Conference*, pages 1–5, Sept. 2006.

S. Li and A. Ramamoorthy. Networked distributed source coding. In *Theoretical Aspects of Distributed Computing in Sensor Networks*, Monographs in Theoretical Computer Science. An EATCS Series, pages 191–224. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-14848-4.

A. Liff and J.K. Wolf. On the optimum sampling rate for discrete-time modeling of continuous-time systems. *IEEE Transactions on Automatic Control*, 11(2):288–290, 1966. ISSN 0018-9286.

C. Lin, Y.-X. He, and N. Xiong. An energy-efficient dynamic power management in wireless sensor networks. *International Symposium on Parallel and Distributed Computing*, pages 148–154, 2006.

B. Lincoln. *Dynamic Programming and Time-Varying Delay Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, May 2003.

X. Liu and A. Goldsmith. Wireless Medium Access Control in distributed control systems. In *Allerton Conference on Communications, Control, and Computing*, 2003.

X. Liu and A. Goldsmith. Wireless network design for distributed control. In *43rd IEEE Conference on Decision and Control*, volume 3, pages 2823–2829, Dec. 2004.

C. R. Mann, R. O. Baldwin, and B. E. Mullins. Wireless sensor networks: guidelines for design and a survey of current research. In *SpringSim*, pages 41–50, 2007.

G.N. Nair, F. Fagnani, S. Zampieri, and R.J. Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, Jan. 2007. ISSN 0018-9219.

D. Nešić and A.R. Teel. Input-output stability properties of networked control systems. *IEEE Transactions on Automatic Control*, 49(10):1650–1667, 2004. ISSN 0018-9286.

P.G. Otanez, J.R. Moyne, and D.M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proceedings of the 2002 American Control Conference*, volume 4, pages 3015–3020, 2002.

P.G. Park, C. Fischione, A. Bonivento, K.H. Johansson, and A. Sangiovanni-Vincentelli. Breath: A self-adapting protocol for wireless sensor networks in control and automation. pages 323–331, June 2008.

R.M. Passos, J.A. Nacif, R.A.F. Mini, A.A.F. Loureiro, A.O. Fernandes, and C.N. Coelho. System-level dynamic power management techniques for communication intensive devices. In *International Conference on Very Large Scale Integration*, pages 373–378, Oct. 2006.

G. Pei and C. Chien. Low power TDMA in large wireless sensor networks. volume 1, pages 347–351, 2001.

Y. Prakash and S.K.S. Gupta. Energy efficient source coding and modulation for wireless applications. *Wireless Communications and Networking*, 1:212–217, March 2003. ISSN 1525-3511.

D. E. Quevedo and D. Nešić. Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet loss. *Automatica*, 2012.

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, Dec. 2008.

D.E. Quevedo, A. Ahlén, and J. Ostergaard. Energy efficient state estimation with wireless sensors through the use of predictive power control and coding. *IEEE Transactions on Signal Processing*, 58 (9):4811–4823, Sept. 2010. ISSN 1053-587X.

M. Rabi and K. H. Johansson. Event-triggered strategies for industrial control over wireless networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, pages 1–7, 2008. ISBN 978-963-9799-36-3.

Y. Raimo and N. Vantaski. Control of systems with time-varying delays. In *3rd workshop on networked control system and fault tolerant control*, 2007.

D. M. Raimondo, D. Limón, M. Lazar, L. Magni, and E. Camacho. Min-max Model Predictive Control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, Jan. 2009.

J.B. Rawlings and D.Q. Mayne. *Model Predictive Control Theory and Design*. Nob Hill Pub., 2009. ISBN 9780975937709.

A. G. Ruzzelli, R. Jurdak, and G. M.P. O'Hare. The RFID wake-up impulse for multi-hop sensor networks. In *1st ACM Workshop on Convergence of RFID and Wireless Sensor Networks and their Applications (SenseID) at the Fifth ACM Conference on Embedded Networked Sensor Systems*, Nov. 2007.

P.S. Sausen, M.A. Spohn, F. Salvadori, M. de Campos, and A. Perkusich. Applying dynamic power management with mode switching in wireless sensor networks. pages 1713–1717, Nov. 2008.

L. Schenato. To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54:1093–1099, 2009. ISSN 0018-9286.

C. Schurgers, V. Raghunathan, and M. B. Srivastava. Power management for energy-aware communication systems. *ACM Trans. Embed. Comput. Syst.*, 2(3):431–447, 2003. ISSN 1539-9087.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 625–56, Oct. 1948.

A. Sinha and A. Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design & Test of Computers*, 18(2):62–74, 2001. ISSN 0740-7475.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, Sept. 2004. ISSN 0018-9286.

D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973. ISSN 0018-9448.

S.C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Transactions on Automatic Control*, 48(12):2163–2171, Dec. 2003. ISSN 0018-9286.

Sundström, O., Ambühl, D., and Guzzella, L. On implementation of dynamic programming for optimal control problems with final state constraints. *Oil Gas Sci. Technol. - Rev. IFP*, 65(1):91–102, 2010.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007. ISSN 0018-9286.

Z. Tang, I.A. Glover, A.N. Evans, and J. He. An energy-efficient adaptive DSC scheme for wireless sensor networks. *IEEE Transactions on Signal Processing*, 87(12):2896–2910, 2007. ISSN 0165-1684. Special Section: Information Processing and Data Management in Wireless Sensor Networks.

M. Trivellato and N. Benvenuto. Cross-layer design of networked control systems. In *IEEE International Conference on Communications*, pages 1–5, 14-18 2009.

D. Varagnolo, P. Chen, L. Schenato, and S.S. Sastry. Performance analysis of different routing protocols in wireless sensor networks for real-time estimation. In *47th IEEE Conference on Decision and Control*, pages 4904–4909, Dec. 2008.

M. Velasco, J. M. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *24th IEEE Real-Time Systems Symposium (Work-in-progress session)*, pages 67–70, 2003.

W. Wang, H. Wang, D. Peng, and H. Sharif. An energy efficient pre-schedule scheme for hybrid CSMA/TDMA MAC in wireless sensor networks. pages 1–5, Oct. 2006.

R. Wen, I. Mareels, and B. Krongold. Optimal power analysis for network lifetime balance in wireless sensor networks. In *International Symposium on Communications and Information Technologies*, pages 1161–1165, Oct. 2007.

D. Wu, J. Wu, and S. Chen. Robust control for discrete-time networked control systems. In *8th World Congress on Intelligent Control and Automation*, pages 3532–3537, July 2010.

Z. Xiong, A.D. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Transactions on Signal Processing*, 21(5):80–94, Sept. 2004. ISSN 1053-5888.

Y. Xu and J.P. Hespanha. Optimal communication logics in networked control systems. In *43rd IEEE Conference on Decision and Control*, volume 4, pages 3527–3532, 2004.

Y. Xu and J.P. Hespanha. Estimation under uncontrolled and controlled communications in networked control systems. pages 842–847, Dec. 2005.

R.W. Yeung. *Information Theory and Network Coding*. Information Technology: Transmission, Processing and Storage. Springer, 2008. ISBN 9780387792330.

J.K. Yook, D.M. Tilbury, and N.R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503–518, 2002. ISSN 1063-6536.

S. Yüksel. A tutorial on quantizer design for networked control systems: Stabilization and optimization. *Applied and Computational Mathematics*, 10, no.3:365–403, 2011.

W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21:84–99, 2001.

B. Zurita Ares, P. G. Park, C. Fischione, A. Speranzon, and K. H. Johansson. On power control for wireless sensor networks: System model, middleware component and experimental evaluation. In *IFAC European Control Conference*, 2007.